THESIS ON INFORMATICS AND SYSTEM ENGINEERING C73

Polynomial Methods for Nonlinear Control Systems

JURI BELIKOV

TUT PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Computer Control

Dissertation was accepted for the defence of the degree of Doctor of Philosophy in informatics and system engineering on April 11, 2012

Supervisor: PhD, Associate Professor Eduard Petlenkov, Department of Computer Control, Tallinn University of Technology

Advisor: DSc, Leading Researcher Ülle Kotta, Institute of Cybernetics, Tallinn University of Technology

Opponents: PhD, Professor Henk Nijmeijer, Department of Mechanical Engineering, Dynamics & Control Group, Eindhoven University of Technology, Eindhoven, The Netherlands

PhD, Professor Anna Maria Perdon, Universitá Politecnica delle Marche, Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione, Ancona, Italy

Defence of the thesis: June 11, 2012

Declaration: Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for any academic degree.

/Juri Belikov/

European Union
European Social Fund

Investing in your future

ISSN 1406-4731
ISBN 978-9949-23-292-5 (publ)
ISBN 978-9949-23-293-2 (PDF)

INFORMAATIKA JA SÜSTEEMITEHNIKA C73

Polünoommeetodid mittelineaarsetes juhtimissüsteemides

JURI BELIKOV

TTÜ KIRJASTUS

Contents

Chapter 1

Introduction

1.1 State of the Art

Nowadays, the control theory operates with many various approaches and methods to solve a full range of problems. They were created and developed in different ways. Many techniques have been obtained from the real needs and established for a specific tasks. Other methods are purely theoretical and meanwhile have not been implemented in real applications. In connection with the above, in this thesis we solve a number of theoretical problems and show the practical applicability of the obtained results. The developed application allows to verify created theory and find new directions. Thus this thesis may be considered as a symbioses of theory and practice. The thesis is devoted to the development of practice oriented theory about modeling and control of nonlinear systems.

Various methods and techniques have been created for the analysis and modeling of control systems. One of the most popular and widely used approaches is based on differential geometry, see, for example, [44] and [73]. However, there are also exist a lot of alternative methods. One of them is based on an algebraic point of view. The idea of the algebraic approach is based on the vector spaces of differential one-forms over suitable fields of nonlinear functions, see [23] for more details. Upon the latter a polynomial framework can be built. Together they are well suited for solving problems for both continuous- and discrete-time cases. Moreover, using the tools based on differential one-forms and the related methods based on the theory of the skew polynomial rings, one can work with algebraic equations rather than with differential counterparts, what inherently is more simple. Polynomial approach has been used so far to study problems like reduction of nonlinear i/o equations [54], linear i/o [51] and transfer equivalence [52], controllability [101] and used also in introducing the concept of transfer function into the nonlinear domain [36, 100]. Thus it has been already proved itself as a practical and reliable mathematical tool.

Within the polynomial framework a lot of different problems can be solved. Some of them are recalled in this thesis. Polynomial methods are used to provide a solution for two typical modeling and control system design problems. First, the realization of nonlinear i/o equations in the state-space form is presented. Second, the model matching problem is considered. Finally, the third problem, for which the polynomial formalism is not used in the explicit form, is a stability issue of a certain class of nonlinear systems linearized by output feedback.

One usually has to deal with an i/o data rather than with analytical description of the studied object when working with the real-life processes. To identify and then to analyze the system and its behavior one of the simplest and convenient way is to use the i/o models. This allows to describe adequately and represent in a compact form the object of practical interest [26]. However, despite the simplicity and success of this approach, state-space description usually becomes the basis for analysis and control of nonlinear systems. Indeed, if the model was not derived from the physical laws of the system, then it is most likely not realizable in the classical state-space form [31], which makes it highly undesirable for further analysis and control design. Thus, one of the goals of this thesis is to bridge the gap between two modeling approaches and present the algorithm allowing to construct a classical state-space equations from an arbitrary i/o model whenever possible. It should be mentioned that under classical we understand a minimal form for which the order of the i/o model coincides with the number of state equations.

The state-space realization of nonlinear i/o models has been the subject of many research works over the years. Some of the existing results may be found in [23, 24, 25, 66, 86, 91, 99] for continuous-time systems and in [60, 63, 64, 80, 81] for discrete-time systems, respectively. Note that a great number of the results have been obtained for single-input single-output (SISO) systems. However, multi-input multi-output (MIMO) case has not been left aside. There are many various approaches developed to solve this problem, among others based either on the sequence of the subspaces of differential one-forms [23], on the sequence of distributions of vector fields as in [91], on the iterative Lie brackets of the vector fields as in [25]. The comparison of different methods and the explicit relations between them have been reported in [56] and [57] for SISO and MIMO cases, respectively. One of the most common approaches is based on the algebraic formalism using the theory of differential one-forms [23]. However, it is not as transparent as polynomial approach in which the system is described by two polynomials from the skew polynomial ring, see, for example, [39] and [86]. Therefore, in this thesis we have restricted our attention to this approach and studied the realization problem for different types of systems.

Though, a lot of work has been done in this area, there are still some gaps related to this topic. Thus, in this thesis recently obtained results by the author are presented. In addition, the realization algorithm proposed in this thesis combines well with the existing results for the reduction problem. A small selection of articles on this subject can be represented by the following references [49, 53, 54] with respect to the types of the systems recalled in the thesis. Both the results presented in the references and this work rely on the same polynomial description of the system. Note that we have not considered the reduction problem in this thesis. Nevertheless, it is worth to be mentioned, because both methods allow to create a complete procedure for deriving state equations starting from the possibly reducible i/o equations. Namely, the realization procedure ends up with the controllable (accessible) realization iff the i/o equation is reduced to the simplest form, being transfer equivalent to the original equation.

Another common problem in the control theory that can be solved within the polynomial framework is a so-called model matching problem (MMP). It is usually used for control system design. The beauty of this problem lies in the fact that it accommodates various other problems such as i/o linearization, disturbance decoupling, model tracking, etc. The basic idea of MMP may be illustrated on the basis of linear systems, where one naturally requires the equality of the transfer functions of the reference model and that of the compensated system. In other words, the output of the controlled object has to coincide with the output of the etalon model after a certain amount of time. On the one hand, the MMP for nonlinear case has been mainly studied within the state-space approach, see [16, 17, 43, 72]. On the other hand, few results exist for systems described by the i/o equations [33, 47, 48]. The reason to state and solve the MMP for the i/o case is justified by the fact that nonlinear systems are not always realizable in the state-space form [38]. However, the MMP statements and solutions given in terms of i/o equations are not that simple and transparent as those relying on the state-space approach. Nevertheless, we can use a transfer function formalism, which can be naturally deduced from the polynomial formalism providing an alternative tool for modeling and analysis of nonlinear control systems [36]. Note that in [41] the transfer function formalism was applied for solving the MMP of nonlinear continuous-time systems. Here we describe a possible way to adapt the results of [41] for nonlinear discrete-time systems. In fact, this procedure is not straight-forward, because the polynomial ring is different as well as the basic operations used therein. The main difference is that the derivative and shift operators define the different multiplication rules. Two types of compensators are usually considered within the MMP, namely feedforward and feedback. Thus within the same problem two subtasks can be simultaneously solved. Note that in case of continuous-time systems it was shown that, unlike the feedback case, the feedforward solution of the MMP does not always exist. Therefore, this case has to be considered in more detail.

The last theoretical problem considered in this thesis is a feedback linearization problem. Indeed, it is well known and has a relatively long history. At different times this problem and its subproblems were successfully solved by various researchers. During the last three decades there has been made a significant research, both in continuous-time, see [18, 23, 44, 45, 68], and in discrete-time, see [3, 34, 71], cases, respectively. Over the years different aspects of the feedback linearization technique were considered and analyzed such as exact, input-output, input-to-state linearization, etc.

Note that in the classical statement of the problem the stability property of the closed-loop is not required. Usually, one is seeking for an appropriate feedback function of control which allows to transform equations describing the object to be controlled to a linear form. Thus, the stability is usually considered separately what, in principle, is logical. This is due to the fact that the issue of calculating the correct position of the poles of the closed-loop system is always solvable, whenever the system is controllable. Nevertheless, both problems have to be considered together before implementing the real control. In this thesis we analyze the stability problem for a certain class of nonlinear discrete-time systems, which can be linearized by an output feedback. On the one hand, such a restriction is not necessary and the presented technique may be extended to a more general class of systems. On the other hand, it will be shown that even in such relatively simple case, the solution of the stated problem becomes a very difficult task because of the high computational complexity. In fact, we consider this problem from a slightly different point of view. Since the stability of the closed-loop system can be achieved under certain conditions, we turned our attention to what happens to the control signal inside the closed-loop, and for which of the reference signals it remains bounded.

Creation and development of specific electronic devices such as computers, smartphones or tablets can be considered through the prism of history as the close cooperation of the theory and practice. A lot of theoretical algorithms were implemented in the form of various programs, thus becoming the engine for further development and improvement. As a result of the advanced computing abilities and ease of use, electronic devices infiltrated into all spheres of human activity. For example, they may be found in different places ranging from battleships to industrial robots, medical tools, and even children's toys. Moreover, complex systems and labour-consuming calculations have led to necessity of additional assistance, and therefore, development of specific computer software. Scientific disciplines also have not left aside. Huge class of practical problems can be solved by numerical methods. They play a much more prominent role in control systems theory and practice than symbolic computations. However, analytical results are very attractive prospect and thus interest in the use of symbolic computations does not decrease. Some techniques in control are especially well-suited for computer algebra applications, for example transformation of the system description into another one or system equations into various types of normal/canonical forms or finding the reduced order equivalent system representation. Note that defining normal forms and deriving algorithms to compute them is a classical topic in computer algebra.

This thesis is focused on the methods based on the theory of differential one-forms and skew polynomials. There exist several symbolic software packages which implement the methods of commutative polynomial theory for control. Two most complete applications are *Polynomial Control Systems*, written in Computer Algebra System (CAS) *Mathematica*, and *Polyx*, written for MATLAB; both deal with linear systems. Additionally, there exists a small Maple-based package *Polycon*, which utilizes commutative algebra to handle systems with rational nonlinearities, see [32]. As for the software related to non-commutative polynomials that is the study case of this thesis, the situation is different. In [21] the authors describe the Maple package *OreModules*, which offers symbolic tools to investigate the structural properties of multidimensional linear systems over Ore algebras. For Maple there is also available a general-purpose Ore algebra package called *OreTools*, described in [1], which does not include any built-in control tools. Comparing the two packages, *OreTools* seems to us more user friendly and its procedures provide a better basis for applications. Nevertheless, regarding the possible extension of *OreModules* and *OreTools* for nonlinear control systems, both have the small but crucial shortage: there is no possibility to define the Ore ring by system equations. In other words, one has to add a procedure for replacing some variables appearing in polynomial coefficients by expressions, defined by i/o equations. Otherwise the result of calculations may be erroneous. In fact, it is unclear if this replacement can be added upon the package by supplementary procedure, or, what is more likely, requires modifying the code of the original package. Finally, there is a small package *GTF_Tools*, see [75], built upon *OreTools*, implementing the construction of the transfer function of nonlinear system. We are not aware of any other software applying the theory of Ore polynomial rings to nonlinear control problems.

Therefore, the special NLControl package was created in the Institute of Cybernetics at Tallinn University of Technology [62]. Originally, the package was created by M. Tõnso, and during long period of time she was developing it alone, see [85]. NLControl is based on the algebraic methods of differential one-forms and skew polynomials, and is developed within symbolic software system *Mathematica*. It was created for solving different problems from the control theory. However, considering the subject of this work further we will focus on a subpackage inside NLControl, which implements the polynomial formalism. The first part of the software includes the functions that implement the basic operations with Ore polynomials, since there is neither built-in functions nor supplement package available for *Mathematica*, addressing these operations. The second part contains the programs for solving modelling problems by polynomial methods.

1.2 Outline and Contributions of the thesis

The thesis is organized such that each subsequent chapter is logically related to the previous one. Therefore, there are five connected parts excluding Chapter 2 which establishes the theoretical background for the problems discussed below. The problem statements and the main contributions of the thesis are presented in Chapters 3-5. Chapter 6 contains new contributions, but mostly focuses on implementation of the results from the previous chapters. Note that at the beginning of each chapter a certain amount of introductory material, necessary for a more accurate understanding of the obtained results, is presented.

Chapter 2

This chapter contains a brief summary of mathematical tools that directly or indirectly are used in the next parts of the thesis. The algebraic and polynomial frameworks as well as related notions are presented. Moreover, some basic definitions are recalled. The main methods, important for understanding the rest part of the thesis, are also considered.

Chapter 3

In this chapter the realization problem of nonlinear i/o equations in the classical state-space form is presented. We applied the polynomial formalism in which the system is described by two polynomials with elements from the non-commutative ring of skew polynomials. Such approach has allowed to simplify significantly the existing step-by-step algorithm based on certain sequences of one-forms. Polynomial formalism appeared to be a very flexible and convenient tool with respect to a given problem. It made possible to unify the solution of this problem for systems defined in different time domains. The presented new explicit formulas allow to compute the differentials of the state coordinates directly from the polynomial description of the nonlinear system. The proposed algorithm combines well with the existing results for the reduction problem, since both rely on system description in terms of two polynomials. Thus, it has led to the possibility of establishing a constructive algorithm which results in the accessible state-space realization whenever possible. Finally, note that the software programmes based on polynomial formulas are more transparent what in detail is considered in Chapter 6.

Chapter is based on the material presented in [11], [10], [8], [9], and [12]. In each of the listed articles roles of the authors were distributed in approximately equal parts. Ü. Kotta stated the problem to be solved and provided the necessary theoretical background. M. Tõnso has implemented the obtained results in *Mathematica* software NLControl package, that made possible to check and improve some moments. The role of the author was deriving main results and writing the corresponding articles, as well as illustrating the developed theory by means of different examples. The only exceptions are papers [9] and [10], in which in addition to the author and Ü. Kotta also P. Kotta has made a contribution. It should be mentioned that in [9], [10] the author proved the main propositions and found the corresponding realizability conditions. Additionally, he derived realizable subclass of an arbitrary order.

Chapter 4

The model matching problem of nonlinear SISO discrete-time systems is considered. We analyze both feedforward and feedback cases. For that purposes the transfer function approach, in which the system is described by the quotient of two polynomials from the non-commutative ring of skew polynomials, is used. It turns out that, in general, the feedforward solution does not always exist. However, the feedback solution always exists. Sometimes there is a need in finding a solution in a class of proper compensators. Therefore, the conditions for the existence of the proper compensator are provided.

Materials from the paper [8] were used for writing Chapter 4. The mentioned article was mostly written by the author of this thesis. But to be honest, he relied on the co-authors' previous results presented in [41] for the case of nonlinear continuous-time systems. So as a matter of fact [8] can be considered as an adaptation of the material from [41] for the discrete-time systems. However, it should be mentioned that in this case there are a number of difficulties due to the use of different time-domain and polynomial formalism, which includes the different multiplication rule. Thus, the role of co-authors was to check the obtained results and the legitimacy of such a transfer. In addition, the author specified a subclass of nonlinear control systems for which the feedforward model matching problem is always solvable.

Chapter 5

This chapter raises the question of stability of a certain class of closed-loop systems. Static and dynamic feedback linearization algorithms may be used to transform the nonlinear controlled system to a linear form. Such approach allows to solve at once two related problems, namely stabilization of the controlled system and stability of the closed loop. Clearly, after obtaining a linear closed-loop system, one can always use the algorithms like pole placement in order to guarantee the stability of the i/o relation. However, we state the related problem that is the stability of the control signal with respect to the reference and output signals. Moreover, we derive the relevant conditions which allow to guarantee that the whole control system behaves in the proper way.

The material presented in this chapter is based on the results published in [15]. Statement and solution of the problem discussed above are mostly belong to the author. The role of the co-author was reduced to verification and refinement of the obtained results, which, to a word, has appeared to be a very valuable.

Chapter 6

The objective of the chapter is devoted to the description of the *Mathematica* software package NLControl. At the beginning of the first section we acquaint the reader with the basic syntax and commands, as well as with the idea of creating special objects to perform various operations. After that the narration passes in the description of functions created on the basis of the theory presented in Chapters 3 and 4. It is followed by illustrative examples with detailed comments. The end of the chapter is devoted to web site, which provides a possibility to use the basic functionality of the NLControl package outside *Mathematica* environment.

Chapter is mainly based on the results presented in [14] and [87]. The contribution of the author can be divided into two parts. First, the author has created functions related to the model matching problem. However, the contribution of the author to development of the package is not limited to it, and there are other programs and functions beyond the research presented within the framework of this thesis. Second, in order to overcome a number of problems associated with the availability of the NLControl package by external users, the special web page has been created on the basis of *webMathematica* service, see [74]. Couple of years ago the management and maintenance of the site was given to the author of this thesis. Then, in cooperation with V. Kaparin, the old web page has been severely altered. V. Kaparin was responsible for the new design to make the site more simple and user-friendly. In addition, to implement the ideas of his colleague, the author rewrote all the functionality and the engine. Thus the structure of the site was simplified, unified, and the amount of code was significantly reduced. Now, the addition of new parts and elements does not take much time compared to the previous version.

1.3 Acknowledgements

Let me begin by expressing my sincerest gratitude to my parents who in spite of everything continued to believe in me and support throughout all these long years. Also I would like to emphasize that this thesis definitely would not exist without the support from my wife Ksenia. Her love is inspiring and encouraging me in the most difficult moments of my life.

I am deeply grateful to my supervisor Dr. Eduard Petlenkov for his guidance and help throughout my thesis work. He gave me complete freedom and could provide all possibilities for comfortable scientific research. Special thanks are directed to Dr. Ülle Kotta. The result of our collaboration is the most part of the material presented in this thesis.

I would like to thank all the co-authors of our joint papers for their valuable contributions. Also I am very grateful to my colleagues for their fruitful discussions and useful comments. Each of them has influenced on the results of this thesis in own way. Among others I would like to mention Dr. Sven Nõmm, who introduced me to the control theory and the enchanting world of a science. Also I am very grateful to Dr. Maris Tõnso for sharing her knowledge. She greatly helped me in complete understanding of the things related to polynomial methods and *Mathematica* software. Many thanks to Kristina Vassiljeva for her strong support and helpful advices. Last but not least, Vadim Kaparin whose help cannot be measured in the framework of existing theories.

This work was partially supported by the European Union through the European Regional Development Fund and the target project SF0140018s08 and the Estonian Science Foundation under grants no. 8738, 8787. Additionally, the research presented in this thesis was supported by the Nations Support Program for the ICT in Higher Education "Tiger University" and Estonian Doctoral School in Information and Communication Technology.

Chapter 2

Preliminaries

The control theory operates with many concepts borrowing definitions and notions from various scientific disciplines. Therefore, this chapter will serve as a brief introduction to only basic mathematical tools which will be important throughout the whole thesis.

2.1 Nonlinear control systems

In this section the author has compiled the basic facts of nonlinear discrete-time systems. Notions of the input-output and state-space forms of the system are presented.

2.1.1 Input-output systems

Consider a nonlinear single-input single-output discrete-time dynamical system, described by the input-output difference equation

$$y(t+n) = \phi(y(t), \ldots, y(t+n-1), u(t), \ldots, u(t+s)), \tag{2.1}$$

where $u : \mathbb{Z} \to \mathbb{R}$ is the input, $y : \mathbb{Z} \to \mathbb{R}$ is the output, $\phi : \mathbb{R}^n \times \mathbb{R}^{s+1} \to \mathbb{R}$ is a real analytic function. Moreover, we assume that n, s are non-negative integers such that $s < n$.

2.1.2 State-space systems

Consider a nonlinear single-input single-output discrete-time system, described by the state equations

$$\begin{aligned} x(t+1) &= f(x(t), u(t)) \\ y(t) &= h(x(t)) \end{aligned} \tag{2.2}$$

where $x : \mathbb{Z} \to \mathbb{R}^n$ is the n-dimensional state vector, $u : \mathbb{Z} \to \mathbb{R}$ is the input, $y : \mathbb{Z} \to \mathbb{R}$ is the output, $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}$ are the real analytic functions.

2.2 Algebraic framework

This section is devoted to a brief explanation of the algebraic framework based on the theory of differential one-forms.

2.2.1 Difference field

Consider the system described by equation (2.1). Let $\mathcal{K}$ denote the field of meromorphic functions in a finite number of independent system variables from the infinite set

$$\mathcal{C} = \{y(t), y(t+1), \ldots, y(t+n-1), u(t+k), k \geq 0\}.$$

The forward-shift operator $\sigma : \mathcal{K} \to \mathcal{K}$ is defined as follows

$$\begin{aligned}\sigma(F)(y(t), y(t+1), \ldots, y(t+n-1), u(t), u(t+1), \ldots, u(t+l)) :=\\ F(y(t+1), y(t+2), \ldots, \phi(\cdot), u(t+1), u(t+2), \ldots, u(t+l+1)),\end{aligned}$$

meaning that σ is applied in the element wise manner, i.e. to each argument of the function F. In other words, the arguments of the function are shifted according to the rules $\sigma y(t+i) = y(t+i+1)$, $\sigma u(t+j) = u(t+j+1)$. Moreover, it should be mentioned that application of the operator σ to $y(t+n-1)$ results in $y(t+n)$ which, according to (2.1), has to be replaced by $\phi(\cdot)$, whenever it occurs in some expression.

Assume that σ is an injective endomorphism on $\mathcal{K}$. It means that, according to [64], system (2.1) has to be submersive, that can be guaranteed by

$$\frac{\partial \phi}{\partial(y(t), u(t))} \not\equiv 0. \tag{2.3}$$

Under the latter assumption, the pair $(\mathcal{K}, \sigma)$ is a *difference field.*

The inverse σ operator is denoted by σ^{-1} and called the backward-shift operator. In general, the field $\mathcal{K}$ is not inversive meaning that $\sigma^{-1}\zeta$ may not have pre-image in $\mathcal{K}$ for any $\zeta \in \mathcal{K}$. Note that under assumption (2.3), there exists, up to an isomorphism, a unique difference overfield $\mathcal{K}^*$, called the *inversive closure* of $\mathcal{K}$ such that $\mathcal{K} \subset \mathcal{K}^*$ and the extension of σ to $\mathcal{K}^*$ is an automorphism, see [22]. A construction of $\mathcal{K}^*$ for practical computations is given in [3]; for the case ϕ in (2.1) being a rational function, a more detailed construction is given in [40].

According to this procedure one needs the rule to compute the k-step backward shifts σ^{-k}, $k \geq 1$ of the system variables. Note that the independent variables of the field $\mathcal{K}$ are given by the elements of the set $\mathcal{C}$, whereas the inversive closure $\mathcal{K}^*$ contains, in addition, the variables $\sigma^{-i}y(t)$ and $\sigma^{-i}u(t), i \geq 1$, where σ^{-i} means the i-time application of the backward-shift operator σ^{-1}. However, take into account that not all of those variables are independent. From (2.1), one can readily calculate the backward shifts of variable $y(t)$, if the condition $\partial\phi/\partial y(t) \not\equiv 0$ holds and $\sigma^{-k}u(t)$ are given. This can be done by solving equation (2.1) with respect to $y(t)$ and applying the backward shift to the result the required number of times. This shows that $\sigma^{-k}y(t)$ for $k \geq 1$ must not be considered as independent variables of the field extension $\mathcal{K}^*$ in the sense that they can be expressed as functions of the other variables such as $\sigma^{-k}u(t)$ and those from $\mathcal{C}$. Note that alternative possibility is to specify $\sigma^{-k}y(t)$ as the independent variables of the field extension $\mathcal{K}^*$. Then under the assumption that $\partial\phi/\partial u(t) \not\equiv 0$ equation (2.1) can be solved for $u(t)$ and shifted back to compute $\sigma^{-k}u(t)$ as dependent variables of $\mathcal{K}^*$. To conclude, we have two possibilities to solve equation (2.1), namely with respect to $y(t)$ or $u(t)$, meaning that either $\sigma^{-k}u(t)$ or $\sigma^{-k}y(t)$ for $k \geq 1$, respectively, have to be chosen as the independent variables for the construction of the inversive closure $\mathcal{K}^*$. Although the choice of variables is not unique, each possible choice brings up a field extension of $\mathcal{K}$ which is isomorphic to $\mathcal{K}^*$.

It should be mentioned that sometimes in the thesis we use the abridged notation $\sigma\varphi(t) = \varphi(t+1) =: \varphi^+$, $\sigma^{-1}\varphi(t) = \varphi(t-1) =: \varphi^-$ and $\sigma^k\varphi(t) = \varphi(t+k) =: \varphi^{[k]}$ for higher-order shifts to make the expressions visually more compact in case of discrete-time systems.

2.2.2 Differential forms

Define the difference vector space $\mathcal{E}$ spanned over the field $\mathcal{K}^*$ as $\mathcal{E} = \mathrm{span}_{\mathcal{K}^*}\mathrm{d}\mathcal{C}$, where either

$$\mathrm{d}\mathcal{C} = \{\mathrm{d}y(t), \mathrm{d}y(t+1), \ldots, \mathrm{d}y(t+n-1), \mathrm{d}y(t-k), k \geq 1, \mathrm{d}u(t+l), l \geq 0\}$$

or

$$\mathrm{d}\mathcal{C} = \{\mathrm{d}y(t), \mathrm{d}y(t+1), \ldots, \mathrm{d}y(t+n-1), \mathrm{d}u(t+l), l \geq 0, \mathrm{d}u(t-k), k \geq 1\}$$

with respect to the chosen independent variables. The elements of $\mathcal{E}$ are called *one-forms*. Note that any element in $\mathcal{E}$ is a vector of the form $\omega = \sum_i \alpha_i \mathrm{d}\zeta_i$ with $\zeta_i \in \mathcal{K}$, where only a finite number of $\alpha_i \neq 0 \in \mathcal{K}^*$. Then the operator $\sigma : \mathcal{K} \to \mathcal{K}$ induces the operator $\sigma : \mathcal{E} \to \mathcal{E}$ by $\sigma(\omega) := \sum_i \sigma\alpha_i \mathrm{d}(\sigma\zeta_i)$ and $\sigma^{-1} : \mathcal{E} \to \mathcal{E}$ by $\sigma^{-1}(\omega) := \sum_i \sigma^{-1}\alpha_i \mathrm{d}(\sigma^{-1}\zeta_i)$.

Define for system (2.1) the non-increasing sequence $\{\mathcal{H}_k\}_{k=1}^{\infty}$ of subspaces of $\mathcal{E}$ as follows [64]

$$
\begin{aligned}
\mathcal{H}_1 &= \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}y(0), \dots, \mathrm{d}y(n-1), \mathrm{d}u(0), \dots, \mathrm{d}u(s)\}, \\
\mathcal{H}_{k+1} &= \{\omega \in \mathcal{H}_k \mid \sigma(\omega) \in \mathcal{H}_k\}, \quad k \geq 1.
\end{aligned} \tag{2.4}
$$

There exists an integer k^* such that $\mathcal{H}_1 \supset \mathcal{H}_2 \supset \cdots \supset \mathcal{H}_{k^*} \supset \mathcal{H}_{k^*+1} = \mathcal{H}_{k^*+2} = \cdots =: \mathcal{H}_\infty$. Existence of k^* comes from the fact that each $\mathcal{H}_k$ is finite-dimensional $\mathcal{K}^*$-vector space, so that at each step either its dimension decreases or $\mathcal{H}_{k+1} = \mathcal{H}_k$.

We say that $\omega \in \mathcal{E}$ is an *exact* one-form, if there exists $\zeta \in \mathcal{K}^*$ such that $\mathrm{d}\zeta = \omega$. A one-form ω for which $\mathrm{d}\omega = 0$ is said to be *closed.* A subspace is said to be completely integrable or closed, if it has locally a basis which consists only of exact one-forms. Integrability of the subspace of one-forms may be checked by the Frobenius theorem below, where the symbol $\mathrm{d}\omega$ denotes the exterior derivative of one-form ω and $\wedge$ means the exterior or wedge product.

Theorem 2.1 ([20]) *Let $\mathcal{V} = \operatorname{span}_{\mathcal{K}^*}\{\omega_1, \dots, \omega_r\}$ be a subspace of $\mathcal{E}$. $\mathcal{V}$ is closed if and only if for all $k = 1, \dots, r$*

$$
\mathrm{d}\omega_k \wedge \omega_1 \wedge \cdots \wedge \omega_r = 0. \tag{2.5}
$$

2.2.3 Reducibility

Definition of the irreducible system is based on the notion of autonomous element.

Definition 2.1 *A function $\varphi \not\equiv \mathrm{const}$ with arguments in $\mathcal{K}^*$ is said to be an autonomous element for system (2.1) if there exist an integer $\nu \geq 1$ and a non-constant meromorphic function F such that $F(\varphi, \sigma\varphi, \dots, \sigma^\nu\varphi) = 0$.*

Note that the function φ depends on $y(t), u(t)$, their forward time shifts and represents the lack of controllability of the nonlinear system. It means that if φ exists and does not equal to constant, then system (2.1) is not accessible. The notion of autonomous element can be used to define (local) irreducibility of the system as follows.

Definition 2.2 *The system (2.1) is said to be generically irreducible if there does not exist any non-constant autonomous element for (2.1) in $\mathcal{K}^*$. Otherwise system (2.1) is called reducible.*

The definition of the autonomous element can be alternatively represented in terms of the subspace $\mathcal{H}_\infty$. A practical condition for evaluating reducibility of system (2.1) is formulated in the following theorem.

Theorem 2.2 ([64]) *The system* (2.1) *is irreducible iff* $\mathcal{H}_\infty = \{0\}$.

Henceforth we assume that the i/o difference equation (2.1) is irreducible in the sense that the system does not have autonomous element. In other words, this means that there does not exist the i/o equation of order $n' < n$, which is transfer equivalent to the original system (2.1). Irreducibility plays an important role in finding accessible state equations.

2.2.4 Observability

Given a system of the form (2.2), let us denote by $\mathcal{X}, \mathcal{Y}^k, \mathcal{Y}$ and $\mathcal{U}$ the following subspaces of differential one-forms

$$\begin{aligned}
\mathcal{X} &= \operatorname{span}_{\mathcal{K}}\{\mathrm{d}x(t)\}, \\
\mathcal{Y}^k &= \operatorname{span}_{\mathcal{K}}\{\mathrm{d}y(t+j),\ 0 \le j \le k\}, \\
\mathcal{Y} &= \operatorname{span}_{\mathcal{K}}\{\mathrm{d}y(t+j),\ j \ge 0\}, \\
\mathcal{U} &= \operatorname{span}_{\mathcal{K}}\{\mathrm{d}u(t+j),\ j \ge 0\}.
\end{aligned}$$

The chain of subspaces

$$0 \subset \mathcal{O}_0 \subset \mathcal{O}_1 \subset \mathcal{O}_2 \subset \cdots \subset \mathcal{O}_k \subset \cdots,$$

where $\mathcal{O}_k = \mathcal{X} \cap (\mathcal{Y}^k + \mathcal{U})$ is called the *observability filtration.* The limit of the observability filtration may be denoted by $\mathcal{O}_\infty = \mathcal{X} \cap (\mathcal{Y} + \mathcal{U})$.

Definition 2.3 *The subspace* $\mathcal{O}_\infty$ *is called the observable space of system* (2.2).

Proposition 2.1 ([50]) *System* (2.2) *is locally single-experiment observable if* $\mathcal{O}_\infty = \mathcal{X}$.

The dimension of the observable space can be computed, according to [82], by

$$\dim \mathcal{O}_\infty = \operatorname{rank}_{\mathcal{K}^*} \frac{\partial\left(h(x(t)), \sigma h(x(t)), \ldots, \sigma^{n-1}h(x(t))\right)}{\partial x(t)} = \bar{n}. \tag{2.6}$$

Remark 2.1 *Note that system* (2.2) *is said to be single-experiment observable if the observability matrix in* (2.6) *has generically full rank. In other words, the dimension of the observable space equals to the number of states, i.e.* $\bar{n} = n$.

2.3 Polynomial framework

In this section we recall the polynomial formalism which allows to represent the nonlinear i/o equation (2.1) via two polynomials and extend the notion of a transfer function to the case of nonlinear system.

2.3.1 Non-commutative polynomial ring

A *left* shift polynomial can be uniquely written in the form

$$a(z) = \alpha_0 z^n + \alpha_1 z^{n-1} + \cdots + \alpha_{n-1} z + \alpha_n \tag{2.7}$$

for $\alpha_i \in \mathcal{K}^*$, $i = 0, \ldots, n$, where z is a formal variable (polynomial indeterminate) and $a(z) \neq 0$ if and only if at least one of the functions α_i, for $i = 0, \ldots, n$, is non-zero. The highest power in the polynomial $a(z)$ is known as its degree and may be denoted by $\deg(a(z))$.

Definition 2.4 *The left skew polynomial ring, induced by $(\mathcal{K}, z)$, is the ring $\mathcal{K}^*[z;\sigma]$ of polynomials in the indeterminate z with usual addition and multiplication satisfying the relation*

$$z \cdot \alpha = \sigma(\alpha) z \tag{2.8}$$

for any $\alpha \in \mathcal{K}^*$.

The skew polynomial ring $\mathcal{K}^*[z;\sigma]$ is proved to satisfy the following left Ore condition.

Proposition 2.2 ([30]) *For all non-zero $a, b \in \mathcal{K}^*[z;\sigma]$ there exist non-zero $a_1, b_1 \in \mathcal{K}^*[z;\sigma]$ such that $a_1 b = b_1 a$.*

If the condition of the above proposition holds, then the skew polynomial ring is called the Ore ring.

Thus, the ring $\mathcal{K}^*[z;\sigma]$ can be embedded into the field of fractions, denoted as $\mathcal{K}^*(z;\sigma)$, see [76]. In $\mathcal{K}^*(z;\sigma)$ one can define the sum of two quotients as

$$b_1^{-1} a_1 + b_2^{-1} a_2 = (\beta_2 b_1)^{-1} (\beta_2 a_1 + \beta_1 a_2),$$

where $\beta_2 b_1 = \beta_1 b_2$ satisfy the Ore condition and the product as

$$b_1^{-1} a_1 b_2^{-1} a_2 = (\beta_2 b_1)^{-1} \alpha_1 a_2, \tag{2.9}$$

where $\beta_2 a_1 = \alpha_1 b_2$ again satisfy the Ore condition.

A ring is called an *integral domain*, if it does not contain any zero divisors. This means that for any two elements a and b of the ring, $ab = 0$ implies either $a = 0$ or $b = 0$.

Proposition 2.3 ([69])

(i) *The ring $\mathcal{K}^*[z;\sigma]$ is an integral domain.*

(ii) *If $a(z)$ and $b(z)$ are non-zero shift polynomials, then* $\deg(a(z) \cdot b(z)) = \deg a(z) + \deg b(z)$.

Note that since σ is an automorphism on $\mathcal{K}^*$, the *left division* operation is well-defined in $\mathcal{K}^*[z;\sigma]$, see [19]. The latter means that given two polynomials $p(z), q(z) \in \mathcal{K}^*[z;\sigma]$, $q(z) \neq 0$ with $\deg(p(z)) > \deg(q(z))$ there exist a unique *left quotient* polynomial $\gamma(z)$ and unique *left remainder* polynomial $\rho(z)$ such that $p(z) = q(z)\gamma(z) + \rho(z)$ and $\deg(\rho(z)) < \deg(q(z))$.

2.3.2 Polynomial system description

In order to describe the i/o equation (2.1) via two skew polynomials from the ring $\mathcal{K}^*[z;\sigma]$, we define

$$z^i \mathrm{d}y := \mathrm{d}y(t+i), \quad z^j \mathrm{d}u := \mathrm{d}u(t+j) \tag{2.10}$$

for $i = 0, \ldots, n-1$ and $j \geq 0$ in the vector space $\mathcal{E}$. An arbitrary one-form

$$\omega = \sum_{i=0}^{n-1} a_i \mathrm{d}y(t+i) + \sum_{j=0}^{l} b_j \mathrm{d}u(t+j),$$

where $a_i, b_j \in \mathcal{K}^*$, can be expressed in terms of two skew polynomials as

$$\omega = \left(\sum_{i=0}^{n-1} a_i z^i \right) \mathrm{d}y(t) + \left(\sum_{j=0}^{l} b_j z^j \right) \mathrm{d}u(t).$$

A skew polynomial may be understood as an operator from $\mathcal{E}$ to $\mathcal{E}$, satisfying the property

$$\left(\sum_{i=0}^{k} a_i z^i \right) (\alpha \mathrm{d}\zeta) := \sum_{i=0}^{k} a_i \left(z^i \cdot \alpha \right) \mathrm{d}\zeta$$

with $a_i, \alpha \in \mathcal{K}^*$ and $\mathrm{d}\zeta \in \{\mathrm{d}y(t), \mathrm{d}u(t)\}$. Moreover, it is easy to notice that $z(\omega) = \sigma(\omega)$ for $\omega \in \mathcal{E}$.

Now, by differentiating the i/o equation (2.1) and using (2.10), we get

$$p(z)\mathrm{d}y + q(z)\mathrm{d}u = 0 \tag{2.11}$$

with $p(z) = z^n - \sum_{i=0}^{n-1} p_i z^i$, $q(z) = -\sum_{j=0}^{s} q_j z^j$ and $p_i = \frac{\partial \phi}{\partial y(t+i)} \in \mathcal{K}^*$, $q_j = \frac{\partial \phi}{\partial u(t+j)} \in \mathcal{K}^*$.

2.3.3 Transfer functions

Now, we recall the definition of the transfer function from [37], see also [36].

Definition 2.5 *An element of the form* $F(z) := p^{-1}(z)q(z) \in \mathcal{K}^*(z;\sigma)$, *such that* $\mathrm{d}y = F(z)\mathrm{d}u$, *is said to be a transfer function of nonlinear system* (2.1).

Note that in the linear case each proper rational function may be interpreted as a transfer function, corresponding to the certain i/o equation of a control system. However, the nonlinear case is more complex. Though every system can be described by the rational function called the transfer function of the nonlinear system, the converse is not always true. It means that not every quotient of skew polynomials necessarily represents a control system, since the corresponding one-form may be non-integrable, see [37] for more details.

It follows from (2.11) that the transfer function of (2.1) can be represented as

$$F(z) = (z^n + \cdots + p_1 z + p_0)^{-1} (q_s z^s + \cdots + q_1 z + q_0) \,.$$

Definition 2.6 *The transfer function* $F(z)$ *is said to be proper if* $s = \deg q(z) \leq n = \deg p(z)$ *and strictly proper if* $s < n$*.*

Definition 2.7 *For a proper or strictly proper transfer function, the difference* $n - s$*, denoted as* $\operatorname{rel}\deg F(z)$*, is called the relative degree.*

Remark 2.2 *It should be mentioned that the theory presented above is recalled for the shift operator based discrete-time systems. It was motivated by the fact that the most of the results accommodate the discrete-time case. However, in the following sections part of the presented theory will also be extended for other types of systems such as continuous- and difference operator based discrete-time cases.*

Chapter 3

State-space realization of i/o equations

This chapter is devoted to the problem of finding classical state equations from a nonlinear input-output system. It is known form the literature that not every i/o model can be transformed into a state-space form. This is the main difference from the linear case in which the classical state-space realization for the proper systems always exists. Therefore, the problem under consideration can be stated in the following way. Suppose that an arbitrary i/o equation is given. Find, if possible, the state coordinates $x(t) \in \mathbb{R}^n$ such that in these coordinates the system takes the classical state-space form and the sequences, generated by i/o and state equations, coincide. The i/o equation is said to be realizable if it admits the classical state-space realization. Hereinafter talking about classical we understand minimal, i.e. accessible realization with respect to the theory presented in Section 2.2. More precisely the problem will be stated in the following sections.

Many control methods and techniques require the system to be in the state-space form. Therefore, in this chapter the basic idea of the realization procedure is explained as well as the corresponding algorithms and formulas for the certain types of systems are derived. Note that presented algorithms are constructive and, therefore, allow to derive the differentials of the state equations directly from the polynomial system description, assuming that the system is realizable and irreducible. First, the case of the single-input single-output equation, defined in terms of the pseudo-linear operator, is presented. After that we consider the realizability conditions for bilinear and quadratic discrete-time systems, and analyze the applicability of the tools developed in the framework of the linear parametric varying systems. Finally, the case of nonlinear multi-input multi-output continuous-time systems is addressed.

3.1 Realization of i/o equations via the tools of pseudo-linear algebra

In this section the realization problem is stated and solved in a unified manner. In particular, this means that both the i/o and state equations are described in terms of the pseudo-linear operator, and the formulas to find the state coordinates are also given in terms of these operators. For the special cases of continuous- and discrete-time systems, these operators take the form of differential, difference or shift operators.

3.1.1 Theoretical background

Below we shortly recall the algebraic formalism from [53], which allows to extend the theory presented in Section 2.2 for the discrete-time case, see also [19].

Let $\mathcal{K}$ be a field and $\sigma : \mathcal{K} \to \mathcal{K}$ an automorphism of $\mathcal{K}$.

Definition 3.1 *A map $\delta : \mathcal{K} \to \mathcal{K}$, which satisfies*

$$\begin{aligned} \delta(a+b) &= \delta(a) + \delta(b), \\ \delta(ab) &= \sigma(a)\delta(b) + \delta(a)b \end{aligned} \tag{3.1}$$

for $a, b \in \mathcal{K}$, is called a pseudo- or σ-derivation.

Definition 3.2 *A σ-differential field is a triple $(\mathcal{K}, \sigma, \delta)$, where $\mathcal{K}$ is a field, σ is an automorphism of $\mathcal{K}$ and δ is a σ-derivation.*

Hereinafter $(\mathcal{K}, \sigma, \delta)$ will be denoted by $\mathcal{K}$. Let V be a vector space over the field $\mathcal{K}$.

Definition 3.3 *An operator $\theta : V \to V$ is called pseudo-linear if*

$$\begin{aligned} \theta(v+w) &= \theta(v) + \theta(w), \\ \theta(aw) &= \sigma(a)\theta(w) + \delta(a)w \end{aligned} \tag{3.2}$$

for any $a \in \mathcal{K}$, $v, w \in V$.

Note that any field $\mathcal{K}$ is a vector space itself. Hence, (3.2) holds for any $a, v, w \in \mathcal{K}$. Any pseudo-derivation $\delta : \mathcal{K} \to \mathcal{K}$ is a pseudo-linear operator by letting $\theta = \delta$. Also for a shift operator, when $\delta = 0$, (3.2) is clearly satisfied by letting $\theta = \sigma$. Thus, pseudo-linear operators allow to handle differential, shift and difference structures from a unified standpoint. The basic types of operators that can be addressed within the pseudo-linear algebra are listed in Table 3.1.

Further in this section we use the abridged notation $\theta(y(t)) = y^{\langle 1 \rangle}$. It can be a derivation $y^{\langle 1 \rangle} = \dot{y}$ that corresponds to the continuous-time

Table 3.1: Basic types of operators

Operator	σ	δ	θ	$f^{\langle 1\rangle}(t)$
differential	$\mathrm{id}_{\mathcal{K}}$	$\frac{\mathrm{d}}{\mathrm{d}t}$	δ	$\frac{\mathrm{d}f(t)}{\mathrm{d}t}$
shift	σ	0	σ	$f(t+1)$
difference	σ	Δ	δ	$\frac{1}{\mu}(f(t+1)-f(t))$

case, a shift $y^{\langle 1\rangle} = \sigma(y)$, or a difference $y^{\langle 1\rangle} = \frac{1}{\mu}\big(\sigma(y) - y\big)$ with $\mu \in \mathbb{R}$ that correspond to two alternative discrete-time cases. Moreover, we use notation $\theta^k(y(t)) = y^{\langle k\rangle}$ for the k-fold application of the pseudo-linear operator.

Consider a nonlinear control system, described by the i/o equation

$$y^{\langle n\rangle} = \phi\left(y, \ldots, y^{\langle n-1\rangle}, u, \ldots, u^{\langle s\rangle}\right), \tag{3.3}$$

where $u, y \in \mathbb{R}$ are the input and the output of the system, respectively, ϕ is a real analytic function, and n, s are non-negative integers such that $s < n$. Assume that system (3.3) is generically submersive, i.e.

$$\operatorname{rank}\frac{\partial\sigma^n(y)}{\partial(y,u)} \not\equiv 0. \tag{3.4}$$

Note that assumption (3.4) is not restrictive since it is necessary condition for system accessibility. Besides, it reduces to the well-known condition (2.3) in case of nonlinear discrete-time systems when $y^{\langle 1\rangle} = \sigma(y)$ [35], and is trivially satisfied in case of continuous-time systems $y^{\langle 1\rangle} = \dot{y}$ when $\sigma(y) = y$.

Let $\mathcal{K}$ denote the field of meromorphic functions in the independent system variables $\mathcal{C} = \left\{y, y^{\langle 1\rangle}, \ldots, y^{\langle n-1\rangle}, u^{\langle k\rangle}, k \geq 0\right\}$ and let δ be a pseudo-derivation defined on $\mathcal{K}$. The field $\mathcal{K}$ may be endowed with a σ-differential structure $(\mathcal{K}, \sigma, \delta)$, determined by the system equations (3.3), see [53]. Define a pseudo-linear operator $\theta : \mathcal{K} \to \mathcal{K}$ separately for derivation, shift and difference operators:

- if $\sigma = \mathrm{id}_{\mathcal{K}}$ and $\delta = \mathrm{d}/\mathrm{d}t$, then a pseudo-linear operator $\theta = \delta$ and

$$\delta\varphi\left(y^{\langle j\rangle}, u^{\langle k\rangle}\right) = \frac{\partial\varphi}{\partial y^{\langle j\rangle}}\delta y^{\langle j\rangle} + \frac{\partial\varphi}{\partial u^{\langle k\rangle}}\delta u^{\langle k\rangle},$$

  where $\delta y^{\langle j\rangle} = y^{\langle j+1\rangle}$ for $j = 0, \ldots, n-2$, $\delta u^{\langle k\rangle} = u^{\langle k+1\rangle}$ for $k \geq 0$, but $\delta y^{\langle n-1\rangle}$ is defined by equation (3.3)

$$\delta y^{\langle n-1\rangle} = \phi\left(y, \ldots, y^{\langle n-1\rangle}, u, \ldots, u^{\langle s\rangle}\right);$$

- if $\delta = 0$, then a pseudo-linear operator $\theta = \sigma$ and

$$\sigma\varphi\left(y^{\langle j\rangle}, u^{\langle k\rangle}\right) = \varphi\left(\sigma y^{\langle j\rangle}, \sigma u^{\langle k\rangle}\right),$$

  where $\sigma y^{\langle j\rangle} = y^{\langle j+1\rangle}$ for $j = 0, \ldots, n-2$, $\sigma y^{\langle n-1\rangle} = \phi(\cdot)$ and $\sigma u^{\langle k\rangle} = u^{\langle k+1\rangle}$ for $k \geq 0$;

- if $\delta = \frac{1}{\mu}(\sigma - \mathrm{id}_{\mathcal{K}}) =: \Delta$ with $\mu \in \mathbb{R}$, then a pseudo-linear operator $\theta = \Delta$ and

$$\Delta\varphi\left(y^{\langle j\rangle}, u^{\langle k\rangle}\right) = \frac{1}{\mu}\left[\varphi\left(\sigma y^{\langle j\rangle}, \sigma u^{\langle k\rangle}\right) - \varphi\left(y^{\langle j\rangle}, u^{\langle k\rangle}\right)\right],$$

  where $\sigma = \mu\Delta + \mathrm{id}_{\mathcal{K}}$, $\Delta y^{\langle j\rangle} = y^{\langle j+1\rangle}$ for $j = 0, \ldots, n-2$, $\Delta y^{\langle n-1\rangle} = \phi(\cdot)$ and $\Delta u^{\langle k\rangle} = u^{\langle k+1\rangle}$ for $k \geq 0$.

Under assumption (3.4), there exists, up to an isomorphism, a unique difference overfield $\mathcal{K}^* \supseteq \mathcal{K}$, called the inversive closure of $\mathcal{K}$, with σ being an automorphism of $\mathcal{K}^*$, see [22]. An explicit construction of inversive closure is given in [6] and [3, 40] for the cases when θ is the difference or shift operator, respectively. Note that in the continuous-time case when $\sigma = \mathrm{id}_{\mathcal{K}}$, $\mathcal{K}^* = \mathcal{K}$.

Recall that in general, the new independent variables of the (isomorphic) field extension may be chosen in two different ways, either as $\sigma^{-k}(y)$, $k \geq 1$, or as $\sigma^{-k}(u)$, $k \geq 1$. Here the σ^{-k} means the k-time application of the backward-shift operator σ^{-1}. The other variables, that is, $\sigma^{-k}(u)$, or $\sigma^{-k}(y)$, respectively, may be calculated from the i/o equation (3.3), applying to it σ^{-1} the required number of times.

Over the field $\mathcal{K}^*$ one can define the vector space $\mathcal{E} := \mathrm{span}_{\mathcal{K}^*}\mathrm{d}\mathcal{C}$ of differential one-forms, where either

$$\mathrm{d}\mathcal{C} = \left\{\mathrm{d}y, \mathrm{d}y^{\langle 1\rangle}, \ldots, \mathrm{d}y^{\langle n-1\rangle}, \mathrm{d}y^{\langle -k\rangle}, k \geq 1, \mathrm{d}u^{\langle l\rangle}, l \geq 0\right\}$$

or

$$\mathrm{d}\mathcal{C} = \left\{\mathrm{d}y, \mathrm{d}y^{\langle 1\rangle}, \ldots, \mathrm{d}y^{\langle n-1\rangle}, \mathrm{d}u^{\langle l\rangle}, l \geq 0, \mathrm{d}u^{\langle -k\rangle}, k \geq 1\right\}$$

with respect to the chosen independent variables. The space $\mathcal{E}$ may be also endowed with the pseudo-linear operator $\theta : \mathcal{E} \to \mathcal{E}$ as follows

$$\theta(\alpha\mathrm{d}\zeta) = \sigma(\alpha)\mathrm{d}(\theta(\zeta)) + \delta(\alpha)\mathrm{d}\zeta.$$

Note that the operator θ commutes with the operator d, $\theta(\mathrm{d}\varphi) = \mathrm{d}(\theta(\varphi))$, and reduces to the well-known rules

$$\delta v = \sum_i \left[\gamma_i \mathrm{d}(\delta\zeta_i) + \delta(\gamma_i)\mathrm{d}\zeta_i\right]$$

and

$$\sigma v = \sum_i \sigma(\gamma_i)\mathrm{d}(\sigma\zeta_i)$$

for the special cases of continuous-time systems ($\sigma = \mathrm{id}_{\mathcal{K}}$, $\theta = \delta = \mathrm{d}/\mathrm{d}t$) and discrete-time systems ($\delta = 0, \theta = \sigma$), respectively.

Next, the polynomial formalism presented in Section 2.3 is extended to the case of systems defined in terms of pseudo-linear operator. A left polynomial can be uniquely written in the form $a(z) = \sum_{i=0}^{n} \alpha_i z^{n-i}$, $\alpha_i \in \mathcal{K}^*$, where z is a formal variable (polynomial indeterminate) and $a(z) \neq 0$ if and only if at least one of the functions α_i, $i = 0, \ldots, n$ is non-zero. Therefore, Definition 2.4 can be rewritten as follows.

Definition 3.4 *Automorphism σ and σ-derivation δ induce the left skew polynomial ring $\mathcal{K}^*[z;\sigma,\delta]$ of polynomials in z over $\mathcal{K}^*$ with usual addition, and multiplication satisfying the relation*

$$z \cdot \alpha = \sigma(\alpha)z + \delta(\alpha) \tag{3.5}$$

for any $\alpha \in \mathcal{K}^*$.

The ring $\mathcal{K}^*[z;\sigma,\delta]$ is an integral domain, see [69].

Remark 3.1 *For any differential field $\mathcal{K}$ with a derivation $\delta = \mathrm{d}/\mathrm{d}t$, $\mathcal{K}^*[z;\mathrm{id}_{\mathcal{K}},\delta] =: \mathcal{K}^*[z;\delta]$ is the ring of linear ordinary differential operators. If σ is the automorphism over $\mathcal{K}^*$ which takes t to $t+1$, then $\mathcal{K}^*[z;\sigma,0] =: \mathcal{K}^*[z;\sigma]$ is the ring of linear shift operators, while $\mathcal{K}^*[z;\sigma,\Delta]$, where $\Delta = \frac{1}{\mu}(\sigma - 1_{\mathcal{K}})$, $\mu \in \mathbb{R}$ is the ring of linear difference operators, see* [19].

Since σ is an automorphism on $\mathcal{K}^*$, the left division algorithm is applicable to the polynomials of $\mathcal{K}^*[z;\sigma,\delta]$ in the way described in Section 2.3. Thus, the nonlinear system (3.3) may be represented in terms of two skew polynomials in $\mathcal{K}^*[z;\sigma,\delta]$. For this differentiate (3.3) to obtain

$$\mathrm{d}y^{\langle n\rangle} - \sum_{i=0}^{n-1} \frac{\partial\phi}{\partial y^{\langle i\rangle}}\mathrm{d}y^{\langle i\rangle} - \sum_{j=0}^{s} \frac{\partial\phi}{\partial u^{\langle j\rangle}}\mathrm{d}u^{\langle j\rangle} = 0 \tag{3.6}$$

which may be rewritten as

$$p(z)\mathrm{d}y + q(z)\mathrm{d}u = 0, \tag{3.7}$$

where $p(z) = z^n - \sum_{i=0}^{n-1} p_i z^i$, $q(z) = -\sum_{j=0}^{s} q_j z^j$ and $p_i = \frac{\partial\phi}{\partial y^{\langle i\rangle}} \in \mathcal{K}^*$, $q_j = \frac{\partial\phi}{\partial u^{\langle j\rangle}} \in \mathcal{K}^*$, i.e. are polynomials over the σ-differential field $\mathcal{K}^*$.

3.1.2 Realization

The realization problem can be formally stated as follows. Given an i/o equation of the form (3.3), find, if possible, the state coordinates $x \in \mathbb{R}^n$ such that in these coordinates system takes the classical state-space form

$$\begin{aligned} x^{\langle 1\rangle} &= f(x,u) \\ y &= h(x) \end{aligned} \tag{3.8}$$

and sequences $\{u(t), y(t), t \geq 0\}$, generated by descriptions (3.3) and (3.8), coincide. The i/o equation (3.3) is said to be realizable if it admits the classical state-space realization of the form (3.8). Again, since we are looking for minimal, i.e. accessible and observable realization, irreducibility plays an important role. An nth-order realization of equation (3.3) is accessible if and only if system (3.3) is irreducible, see [53] for technical details. Besides, according to [82], system (3.8) is said to be single-experiment observable if the observability matrix has generically full rank, namely

$$\operatorname{rank}_{\mathcal{K}^*} \frac{\partial \left(h(x), \ldots, h^{\langle n-1\rangle}\left(x, u, \ldots, u^{\langle n-2\rangle}\right)\right)}{\partial x} = n.$$

Define the non-increasing sequence $\{\mathcal{H}_k\}_{k=1}^{\infty}$ of subspaces of $\mathcal{E}$ as follows

$$\begin{aligned} \mathcal{H}_1 &= \operatorname{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \ldots, \mathrm{d}y^{\langle n-1\rangle}, \mathrm{d}u, \ldots, \mathrm{d}u^{\langle s\rangle}\right\}, \\ \mathcal{H}_{k+1} &= \left\{\omega \in \mathcal{H}_k \mid \omega^{\langle 1\rangle} \in \mathcal{H}_k\right\}, \quad k \geq 1, \end{aligned} \tag{3.9}$$

playing the key role in the study of realization problem, see [55]. Now, we recall the necessary and sufficient realizability conditions.

Theorem 3.1 ([55]) *The nonlinear i/o equation* (3.3) *has an observable state-space realization if and only if the subspace $\mathcal{H}_{s+2}$, defined by* (3.9)*, is completely integrable.*

Though [55] provides necessary and sufficient realizability conditions for i/o equation (3.3), and the sufficiency part of the proof suggests that the integrable basis of $\mathcal{H}_{s+2}$ defines the differentials of the state coordinates $\mathrm{d}x_i$, $i = 1, \ldots, n$, it does not address the computation of the subspace $\mathcal{H}_{s+2}$. Whereas [55] and this thesis both use the formalism of differential forms, we build upon the latter the polynomial framework as in [53]. Our main result formulated below in Theorem 3.2 provides explicit polynomial formulas for computing the basis vectors of the subspace $\mathcal{H}_{s+2}$.

Recall that since σ is an automorphism of $\mathcal{K}^*$, the left division operation is well-defined in $\mathcal{K}^*[z;\sigma,\delta]$. Below we need certain sequences of left quotients, which are computed by starting with the skew polynomials $p_0 := p$

and $q_0 := q$ in (3.7) and then the element p_l (q_l) for $l = 1, \ldots, n$ is found as the left quotient of p_{l-1} (q_{l-1}) and the polynomial z:

$$\begin{aligned} p_{l-1} &= z \cdot p_l + r_l, \quad \deg r_l = 0, \\ q_{l-1} &= z \cdot q_l + \rho_l, \quad \deg \rho_l = 0. \end{aligned} \tag{3.10}$$

We introduce certain one-forms, in terms of which we will formulate the main result of this section in Theorem 3.2:

$$\omega_l = \begin{bmatrix} p_l & q_l \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}, \qquad l = 1, \ldots, n. \tag{3.11}$$

Theorem 3.2 *For the input-output model* (3.3), *the subspaces $\mathcal{H}_k$ may be calculated as*

$$\mathcal{H}_k = \mathrm{span}_{\mathcal{K}^*} \left\{ \omega_1, \ldots, \omega_n, \mathrm{d}u, \ldots, \mathrm{d}u^{\langle s-k+1 \rangle} \right\} \tag{3.12}$$

for $k = 1, \ldots, s+1$ and

$$\mathcal{H}_{s+2} = \mathrm{span}_{\mathcal{K}^*} \{ \omega_1, \ldots, \omega_n \}. \tag{3.13}$$

Proof: see Appendix.

Remark 3.2 *On the one hand, according to Theorem* 3.1, *the state coordinates can be obtained by integrating the exact basis elements of the subspace $\mathcal{H}_{s+2}$. In addition, the polynomial formulas, presented above, may be used to explicitly define the differentials of the state coordinates. Thus, we do not need to calculate all the previous*[1] *subspaces. On the other hand, in order to show that the subspace $\mathcal{H}_{s+2}$, constructed according to* (3.12) *or* (3.13), *coincides with the classical definition of subspaces* (3.9), *we have to use mathematical induction and show that the elements of the subspace $\mathcal{H}_k$ are in $\mathcal{H}_{k-1}$ and so on.*

Remark 3.3 *Note that though in case of the realizable i/o equation, $\mathcal{H}_{s+2}$, defined by* (3.13), *is completely integrable, the one-forms ω_l for $l = 1, \ldots, n$ are not necessarily always exact. In such a case, one has to find for $\mathcal{H}_{s+2}$ a new (locally) exact basis, using linear transformations over the field $\mathcal{K}^*$.*

[1] In fact, those subspaces may be useful for checking whether the system is realizable or not, since, for example, if already $\mathcal{H}_3$ is not integrable one may conclude that the i/o equations are not realizable. Another illustrative problem for which the previous subspaces may be important is the problem of lowering the input derivatives in the generalized state equations, see [61] for details.

In the algorithm below we summarize the realization procedure.

Algorithm:

Step 1. Given the i/o equation (3.3), find the polynomial description of the system by rewriting (3.6) in the form (3.7), using the rule (3.5).

Step 2. Given $p_0(z) := p(z)$ and $q_0(z) := q(z)$, obtained at **Step 1**, calculate, according to (3.10), two sequences $\{p_l(z)\}_{l=1}^n$, $\{q_l(z)\}_{l=1}^n$ of left quotients of polynomials $p(z)$ and $q(z)$, respectively.

Step 3. Construct the vector space $\mathcal{H}_{s+2} = \mathrm{span}_{\mathcal{K}^*}\{\omega_1, \ldots, \omega_n\}$, where the one-forms $\omega_l := p_l(z)\mathrm{d}y + q_l(z)\mathrm{d}u$, for $l = 1, \ldots, n$, and simplify the basis elements of $\mathcal{H}_{s+2}$ whenever possible.

Step 4. Check the integrability of the vector space $\mathcal{H}_{s+2}$. If $\mathcal{H}_{s+2}$ is integrable, go to **Step 5**. Otherwise, inform that the i/o equation is not realizable and go to **Step 7**.

Step 5. Check whether the basis one-forms of $\mathcal{H}_{s+2}$ are exact or not. If this is true, integrate the one-forms $\omega_1, \ldots, \omega_n$ to get $x_1, \ldots, x_n$. Otherwise, use before a linear transformation to find a new integrable basis.

Step 6. Compute the state equations, applying the pseudo-linear operator to $x_1, \ldots, x_n$.

Step 7. End of the algorithm.

Example 3.1 Consider the i/o equation

$$y^{\langle 2\rangle} + \alpha_1 y^{\langle 1\rangle} + \alpha_0 y\left(1 + \varepsilon_1 y^2\right) = \beta_0(1 + \varepsilon_2 y)u, \tag{3.14}$$

where $\alpha_0, \alpha_1, \beta_0, \varepsilon_1, \varepsilon_2 \in \mathbb{R}$. In [98] the system was studied separately for continuous- and discrete-time cases, the latter being based on the difference operator description. Here, however, we address the model within the framework of pseudo-linear algebra which accommodates both special cases in a single model.

Equation (3.14) can be described as in (3.7) by two polynomials $p(z) = z^2 + \alpha_1 z + \alpha_0 + 3\alpha_0\varepsilon_1 y^2 - \beta_0\varepsilon_2 u$ and $q(z) = -\beta_0(1 + \varepsilon_2 y)$. From (3.14), $n = 2$ and $s = 0$. Given $p_0(z) = p(z)$ and $q_0(z) = q(z)$, compute iteratively, according to (3.10), the polynomials $p_l(z)$ and $q_l(z)$ for $l = 1, 2$ dividing respectively $p_{l-1}(z)$ and $q_{l-1}(z)$ by z from the left:

$$\begin{aligned}
p_0(z) &= z^2 + \alpha_1 z + \alpha_0 + 3\alpha_0\varepsilon_1 y^2 - \beta_0\varepsilon_2 u, & q_0(z) &= -\beta_0(1 + \varepsilon_2 y),\\
p_1(z) &= z + \alpha_1, & q_1(z) &= 0,\\
p_2(z) &= 1, & q_2(z) &= 0.
\end{aligned}$$

Since $s = 0$, according to Remark 3.3 and using (3.11), the basis elements of the last subspace $\mathcal{H}_{s+2} = \mathcal{H}_2 = \mathrm{span}_{\mathcal{K}^*}\{\omega_1, \omega_2\}$ can be represented in the following form

$$\begin{aligned}
\omega_1 &= p_1(z)\mathrm{d}y + q_1(z)\mathrm{d}u = (z + \alpha_1)\mathrm{d}y,\\
\omega_2 &= p_2(z)\mathrm{d}y + q_2(z)\mathrm{d}u = \mathrm{d}y.
\end{aligned}$$

Finally, we get $\mathcal{H}_2 = \mathrm{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \mathrm{d}y^{\langle 1\rangle} + \alpha_1 \mathrm{d}y\right\}$. Simplifying the basis one-forms, the subspace may be rewritten as $\mathcal{H}_2 = \mathrm{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \mathrm{d}y^{\langle 1\rangle}\right\}$. The basis elements are exact, so one may choose $\mathrm{d}x_1 = \mathrm{d}y, \mathrm{d}x_2 = \mathrm{d}y^{\langle 1\rangle}$ and the state equations are

$$\begin{aligned}
x_1^{\langle 1\rangle} &= x_2\\
x_2^{\langle 1\rangle} &= -\alpha_1 x_2 - \alpha_0\left(1 + \varepsilon_1 x_1^2\right)x_1 + \beta_0(1 + \varepsilon_2 x_1)u\\
y &= x_1
\end{aligned} \tag{3.15}$$

For the special cases of continuous- and discrete-time models, (3.15) takes the forms

$$\begin{aligned}
\dot{x}_1 &= x_2\\
\dot{x}_2 &= f(x_1, x_2, u)\\
y &= x_1
\end{aligned} \qquad \text{and} \qquad \begin{aligned}
x_1^{\Delta} &= x_2\\
x_2^{\Delta} &= f(x_1, x_2, u)\\
y &= x_1
\end{aligned}$$

respectively, with $f(x_1, x_2, u) = -\alpha_1 x_2 - \alpha_0\left(1 + \varepsilon_1 x_1^2\right)x_1 + \beta_0(1 + \varepsilon_2 x_1)u$, like in [98].

It should be mentioned that since equation (3.14) depends only on u, but not on $u^{\langle k\rangle}$ for $k \geq 1$, by (3.9), $\mathcal{H}_2 = \mathrm{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \mathrm{d}y^{\langle 1\rangle}\right\}$, see [23] for details. In fact, we may skip the intermediate computations and directly write out the state-space realization of i/o equations (3.14); however, we decided to show them to illustrate the theory presented above.

Example 3.2 Consider the i/o equation

$$y^{\langle 2\rangle} = y^{\langle 1\rangle}u^{\langle 1\rangle} + uy$$

that may be described as in (3.7) by two polynomials $p(z) = z^2 - u^{\langle 1\rangle}z - u$ and $q(z) = -y^{\langle 1\rangle}z - y$. Note that $n = 2$ and $s = 1$. Given $p_0(z) := p(z)$ and $q_0(z) := q(z)$, compute, according to (3.10), two sequences of the left quotients as follows

$$\begin{aligned}
p_1(z) &= z - \sigma^{-1}\left(u^{\langle 1\rangle}\right), & q_1(z) &= -\sigma^{-1}\left(y^{\langle 1\rangle}\right),\\
p_2(z) &= 1, & q_2(z) &= 0.
\end{aligned}$$

By (3.11), one-forms of the subspace $\mathcal{H}_{s+2} = \mathcal{H}_3 = \mathrm{span}_{\mathcal{K}^*}\{\omega_1, \omega_2\}$ are

$$\omega_1 = p_1(z)\mathrm{d}y + q_1(z)\mathrm{d}u = \left(z - \sigma^{-1}\left(u^{\langle 1\rangle}\right)\right)\mathrm{d}y - \sigma^{-1}\left(y^{\langle 1\rangle}\right)\mathrm{d}u,$$
$$\omega_2 = p_2(z)\mathrm{d}y + q_2(z)\mathrm{d}u = \mathrm{d}y.$$

Since $\mathrm{d}y$ is the basis vector of the subspace $\mathcal{H}_3$, ω_1 may be simplified, resulting in $\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \mathrm{d}y^{\langle 1\rangle} - \sigma^{-1}(y^{\langle 1\rangle})\mathrm{d}u\right\}$. Note that integrability of $\mathcal{H}_3$ depends on $\sigma^{-1}\left(y^{\langle 1\rangle}\right)$. Next, we separately consider three typical cases. In the continuous-time case, when $\sigma = \sigma^{-1} = \mathrm{id}_{\mathcal{K}}$, the subspace

$$\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}y, \mathrm{d}\dot{y} - \dot{y}\mathrm{d}u\}$$

is, by Theorem 2.1, integrable. The choice $x_1 = y$, $x_2 = e^{-u}\dot{y}$ yields the classical state equations

$$\begin{aligned} \dot{x}_1 &= e^{u}x_2 \\ \dot{x}_2 &= e^{-u}ux_1 \\ y &= x_1 \end{aligned}$$

In the discrete-time case, when $\theta = \sigma$ and $\sigma^{-1}(\sigma(y)) = y$, the subspace

$$\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y, \mathrm{d}\sigma(y) - y\mathrm{d}u\}$$

is again, by Theorem 2.1, integrable, yielding the state coordinates $x_1 = y$, $x_2 = \sigma(y) - uy$ and the state equations

$$\begin{aligned} \sigma(x_1) &= ux_1 + x_2 \\ \sigma(x_2) &= ux_1 \\ y &= x_1 \end{aligned}$$

In the discrete-time case, when $\theta = \Delta$, the subspace

$$\begin{aligned} \mathcal{H}_3 = \mathrm{span}_{\mathcal{K}^*}&\left\{\mathrm{d}y, \mathrm{d}y^{\Delta} - \sigma^{-1}\left(y^{\Delta}\right)\mathrm{d}u\right\} = \\ &= \mathrm{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \frac{1}{\mu}\mathrm{d}\sigma(y) + \frac{1}{\mu}\left(\sigma^{-1}(y) - y\right)\mathrm{d}u\right\}. \end{aligned}$$

Since $\mathrm{d}\omega_2 \wedge \omega_1 = \frac{1}{\mu}\mathrm{d}[\sigma^{-1}(y)] \wedge \mathrm{d}u \wedge \mathrm{d}y \neq 0$, then, according to Theorem 2.1, $\mathcal{H}_3$ is not integrable. Recall that either $\sigma^{-1}(y)$ or $\sigma^{-1}(u)$ may be chosen as the independent variable of field extension $\mathcal{K}^*$. In the latter case $\sigma^{-1}(y) = \frac{\sigma(y) - 2y - yu + \sigma^{-1}(u)y}{(\mu^2+1)\sigma^{-1}(u) - u - 1}$, yielding again that $\mathrm{d}\omega_2 \wedge \omega_1 \neq 0$. Therefore, the system is not realizable.

Example 3.3 Consider the "ball and beam" system, with input being the angle and output being the ball position. The input-output equation of the system is

$$y^{\langle 2\rangle} = \frac{mR^2}{J + mR^2}\left(y\left(u^{\langle 1\rangle}\right)^2 - g\sin(u)\right), \tag{3.16}$$

where the constant parameters J, R, m represent, respectively, the inertia, radius and mass of the ball, and g is the gravitational constant. Usually, system (3.16) is considered separately for continuous- and discrete-time cases, see for example [42] and [77], respectively. Here, however, we consider the pseudo-linear operator based system description which accommodates both continuous and discrete-time models.

System (3.16) can be described as in (3.7) by two polynomials $p(z) = z^2 - \frac{mR^2}{J+mR^2}\left(u^{\langle 1\rangle}\right)^2$ and $q(z) = -\frac{2mR^2}{J+mR^2} yu^{\langle 1\rangle} z + \frac{gmR^2}{J+mR^2}\cos(u)$. Note that $n = 2$ and $s = 1$. Given $p_0(z) := p(z)$ and $q_0(z) := q(z)$, compute, according to (3.10), the left quotients of the polynomials $p_l(z)$ and $q_l(z)$ for $l = 1, 2$ as

$$\begin{aligned} p_1(z) &= z, & q_1(z) &= -\frac{2mR^2}{J+mR^2}\sigma^{-1}\left(yu^{\langle 1\rangle}\right), \\ p_2(z) &= 1, & q_2(z) &= 0. \end{aligned}$$

By (3.11), one-forms of the subspace $\mathcal{H}_{s+2} = \mathcal{H}_3 = \mathrm{span}_{\mathcal{K}^*}\{\omega_1, \omega_2\}$ are

$$\begin{aligned} \omega_1 &= p_1(z)\mathrm{d}y + q_1(z)\mathrm{d}u = \mathrm{d}y^{\langle 1\rangle} - \frac{2mR^2}{J+mR^2}\sigma^{-1}\left(yu^{\langle 1\rangle}\right)\mathrm{d}u, \\ \omega_2 &= p_2(z)\mathrm{d}y + q_2(z)\mathrm{d}u = \mathrm{d}y. \end{aligned}$$

By analogy with the previous example, consider separately three typical cases. In the continuous-time case, when $\sigma = \sigma^{-1} = \mathrm{id}_{\mathcal{K}}$, the subspace

$$\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}}\left\{\mathrm{d}y, \mathrm{d}\dot{y} - \tfrac{2mR^2}{J+mR^2} y\dot{u}\mathrm{d}u\right\}$$

is not integrable.

In the discrete-time case, when $\theta = \sigma$, the subspace

$$\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \mathrm{d}\sigma(y) - \tfrac{2mR^2}{J+mR^2}\sigma^{-1}(y)u\mathrm{d}u\right\}$$

is not integrable.

In the discrete-time case, when $\theta = \Delta$, the subspace

$$\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}^*}\left\{\mathrm{d}y, \mathrm{d}y^{\Delta} - \tfrac{2mR^2}{J+mR^2}\sigma^{-1}\left(yu^{\Delta}\right)\mathrm{d}u\right\}$$

is not integrable.

Thus, we may conclude that it is not possible to find the classical state-space realization of system (3.16) for the cases listed above.

3.2 Realization of i/o bilinear and quadratic models

In this section bilinear and quadratic equations are analyzed. Note that both are special cases of equation (2.1).

3.2.1 Introduction

A nonlinear control system, described by the i/o difference equation, is called bilinear if the equation contains the products of the input and output at the same or different time instances, but is otherwise linear in the output and input

$$y(t+n) = \sum_{i=1}^{n} a_i y(t+n-i) + \sum_{i=1}^{n} b_i u(t+n-i) + \\ + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} y(t+n-i)u(t+n-j). \quad (3.17)$$

It is a simple nonlinear extension of a linear system. Bilinear systems are common in engineering design and are also used as models of natural phenomena with variable growth rates, see [29, 70]. Furthermore, the bilinear structure is often assumed in system identification as a simple approximation of nonlinear dynamics, see [27, 84, 97]. Note that if the accuracy of the identified bilinear model is not enough or the corresponding order is too high, one may use quadratic models as a reasonable extension, see [83] for details.

$$\begin{aligned} y(t+n) = & \sum_{i=1}^{n} a_i y(t+n-i) + \sum_{i=1}^{n} b_i u(t+n-i) + \\ & + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} y(t+n-i)u(t+n-j) + \\ & + \sum_{i=1}^{n}\sum_{j=i}^{n} d_{ij} y(t+n-i)y(t+n-j) + \\ & + \sum_{i=1}^{n}\sum_{j=i}^{n} e_{ij} u(t+n-i)u(t+n-j). \end{aligned} \quad (3.18)$$

Unfortunately, the majority of nonlinear i/o equations, including those of bilinear equations, cannot be represented by state equations. Thus, the goal of this section is devoted to the study of the low-order discrete-time bilinear and quadratic equations with respect to realizability in the classical state-space form. More precisely, we demonstrate that the certain restrictions on the coefficients make the system to be realizable. In addition, a new realizable subclass and the corresponding state equations for bilinear system of the general order are presented. Both types of models are attractive because of their simplicity in form and identification, since both models are linear in parameters. In many cases a simple structure provides a reasonable approximation. Note that some realizable model structures allow simple translation from the i/o equation to state equations, whereas in the other cases the relationship is more complicated.

3.2.2 Main tools

Here, we recall the basic theoretical aspects which allow to define realizability conditions for bilinear and quadratic equations.

Theorem 3.3 ([64]) *The nonlinear system described by the irreducible input-output difference equation* (2.1) *has an observable and accessible state-space realization if and only if for* $1 \leq k \leq s+2$ *the subspaces* $\mathcal{H}_k$ *defined by* (2.4) *are completely integrable.*

Theorem 3.3 provides the solution of the minimal realization problem for an arbitrary nonlinear i/o difference equation of the form (2.1). In fact, Theorem 3.3 follows as a special case from Theorem 3.1. However, further in this section we analyze equations using polynomial framework. Therefore, in order to work with the polynomials (2.11) describing system (2.1), we recall the following definition.

Definition 3.5 ([63]) *The shift-and-cut operator* $\sigma_c^{-1} : \mathcal{K}^*[z;\sigma] \to \mathcal{K}^*[z;\sigma]$ *is defined as* $\sigma_c^{-1}(p(z)) = \sigma^{-1}(p(z) - p_0)$*.*

Note that Definition 3.5 presents a natural extension to the nonlinear case of the shift-and-cut operator introduced for linear time-invariant systems in [79]. Iterated k-fold application of σ_c^{-1} is denoted as σ_c^{-k}. Moreover, the shift-and-cut operator obeys the following elementary property which is used further in deriving realizability conditions. Let $r(z) \in \mathcal{K}^*[z;\sigma]$ and $r(z) = \sum_{i=0}^{\zeta} r_i z^i$, then

$$\sigma_c^{-l}(r(z)) = \sum_{i=l}^{\zeta} \left(\sigma^{-l} r_i\right) z^{i-l}.$$

Now, using definition of the shift-and-cut operator, the following theorem can be formulated, which allows to find subspaces $\mathcal{H}_k$ in case of polynomial system description.

Theorem 3.4 ([63]) *For the i/o model* (2.1)*, the subspaces* $\mathcal{H}_k$ *for* $k = 2, \ldots, s+2$ *can be calculated as* $\mathcal{H}_k = \mathrm{span}_{\mathcal{K}^*}\{\omega_l, \mathrm{d}u(t), \ldots, \mathrm{d}u(t+s-k+1)\}$, *where for* $l = 1, \ldots, k-2$

$$\omega_l = \sigma_c^{-l} \begin{bmatrix} p(z) & q(z) \end{bmatrix} \begin{bmatrix} \mathrm{d}y(t) \\ \mathrm{d}u(t) \end{bmatrix}. \tag{3.19}$$

Remark 3.4 *It is not hard to see that the shift-and-cut operator, recalled in Definition* 3.5*, is a special case of the left division operation of two polynomials from* $\mathcal{K}^*[z;\sigma]$ *presented in Section* 2.3 *and illustrated by relations* (3.10) *for a more general case. In other words, suppose that* $p_l(z) = \sum_{i=0}^{n} p_{i,l} z^{n-i}$ *for* $l = 1,\ldots,n$ *in* (3.10)*. Then, due to the commutation rule* (2.8)*, the relation* $p_{l-1} = z \cdot p_l + r_l$*,* $\deg r_l = 0$ *can be rewritten as* $p_l = \sigma^{-1}(p_{l-1} - r_l)$ *with* $r_l = p_{n,l-1}$*. Moreover, the application of the shift-and-cut operator is less time consuming than the division operation.*

3.2.3 Realizability conditions for bilinear models

Proposition 3.1 *The third-order bilinear system described by the i/o equation*

$$\begin{aligned} y^{+++} = a_1 y^{++} + a_2 y^{+} + a_3 y + b_1 u^{++} + b_2 u^{+} + b_3 u + \\ + c_{11} y^{++} u^{++} + c_{12} y^{++} u^{+} + c_{13} y^{++} u + c_{21} y^{+} u^{++} + \\ + c_{22} y^{+} u^{+} + c_{23} y^{+} u + c_{31} y u^{++} + c_{32} y u^{+} + c_{33} y u \end{aligned} \quad (3.20)$$

is realizable in the classical state-space form if and only if one of the following five conditions is satisfied:

(i) $a_3 = c_{21} = c_{31} = c_{32} = c_{33} = 0$;

(ii) $b_1 = c_{11} = c_{21} = c_{31} = c_{32} = 0$;

(iii) $b_2 = b_3 = c_{12} = c_{13} = c_{22} = c_{23} = c_{32} = c_{33} = 0$;

(iv) $b_3 = c_{13} = c_{23} = c_{31} = c_{33} = 0$;

(v) $c_{13} = c_{21} = c_{31} = c_{32} = 0$.

Proof: see Appendix.

Compared with the results obtained in [58], using the polynomial approach, we have found an additional restriction (i), meaning that the earlier conditions were actually only sufficient, whereas those presented in Proposition 3.1 are both necessary and sufficient. The novelty of the approach consists in consideration of two possible routes for construction of the inversive closure of the difference field $\mathcal{K}$. In the earlier paper only y^- was taken as an independent variable of $\mathcal{K}^*$, whereas in the proof presented in Appendix also the alternative case of u^-, being the independent variable of $\mathcal{K}^*$, is taken into account.

Proposition 3.2 *The fourth-order bilinear system described by the i/o equation* (3.17) *with* $n = 4$ *is realizable in the classical state-space form if and only if one of the following conditions is satisfied:*

(i) $a_3 = a_4 = c_{21} = c_{31} = c_{32} = c_{33} = c_{34} = c_{41} = c_{42} = c_{43} = c_{44} = 0;$

(ii) $a_4 = b_1 = c_{11} = c_{21} = c_{31} = c_{32} = c_{41} = c_{42} = c_{43} = c_{44} = 0;$

(iii) $a_4 = c_{14} = c_{21} = c_{31} = c_{32} = c_{41} = c_{42} = c_{43} = c_{44} = 0;$

(iv) $b_1 = b_2 = c_{11} = c_{12} = c_{21} = c_{22} = c_{31} = c_{32} = c_{41} = c_{42} = c_{43} = 0;$

(v) $b_1 = b_4 = c_{11} = c_{14} = c_{21} = c_{24} = c_{31} = c_{34} = c_{41} = c_{42} = c_{43} = 0;$

(vi) $b_1 = c_{11} = c_{14} = c_{21} = c_{31} = c_{32} = c_{41} = c_{42} = c_{43} = 0;$

(vii) $b_2 = b_3 = b_4 = c_{12} = c_{13} = c_{14} = c_{22} = c_{23} = c_{24} = c_{32} = c_{33} = c_{34} = c_{42} = c_{43} = c_{44} = 0;$

(viii) $b_3 = b_4 = c_{13} = c_{14} = c_{23} = c_{24} = c_{33} = c_{34} = c_{41} = c_{43} = c_{44} = 0;$

(ix) $c_{13} = c_{14} = c_{21} = c_{24} = c_{31} = c_{32} = c_{41} = c_{42} = c_{43} = 0.$

Proof: sketch of the proof is given in Appendix.

On the basis of Propositions 3.1 and 3.2 we introduce the new nth-order realizable subclass of the i/o bilinear models.

Proposition 3.3 *The class of bilinear models*

$$\begin{aligned} y(t+n) = \sum_{i=1}^{2} a_i y(t+n-i) + \sum_{i=1}^{n} b_i u(t+n-i) + \\ + \sum_{i=1}^{2} \sum_{j=i}^{n} c_{ij} y(t+n-i) u(t+n-j) \end{aligned} \tag{3.21}$$

is realizable in the classical state-space form

$$\begin{aligned} x_1(t+1) &= x_2(t) + (a_1 + c_{11}u(t))x_1(t) + b_1 u(t) \\ x_2(t+1) &= (a_2 + c_{22}u(t))x_1(t) + b_2 u(t) + \\ &\quad + c_{12}u(t)[x_2(t) + (a_1 + c_{11}u(t))x_1(t) + b_1 u(t)] \\ y(t) &= x_1(t) \end{aligned} \tag{3.22}$$

for $n = 2$ *and*

$$
\begin{aligned}
x_1(t+1) &= x_2(t) + (a_1 + c_{11}u(t))x_1(t) + b_1u(t) + \\
&\quad + \sum_{i=2}^{n-1}(b_i + c_{1i}x_1(t))x_{n-i+2}(t) \\
x_2(t+1) &= (a_2 + c_{22}u(t))x_1(t) + b_nx_3(t) + \\
&\quad + x_1(t)\sum_{i=3}^{n} c_{2i}x_{n-i+3}(t) + \\
&\quad + c_{1n}x_3(t)\Bigg[x_2(t) + (a_1 + c_{11}u(t))x_1(t) + b_1u(t) + \\
&\quad + \sum_{i=2}^{n-1}(b_i + c_{1i}x_1(t))x_{n-i+2}(t)\Bigg] \\
x_3(t+1) &= x_4(t) \\
&\vdots \\
x_{n-1}(t+1) &= x_n(t) \\
x_n(t+1) &= u(t) \\
y(t) &= x_1(t)
\end{aligned}
$$

for $n \geq 3$.

Proof: see Appendix.

Example 3.4 Consider the system of a jacketed Continuously Stirred Tank Reactor (CSTR) from [5], described by the third-order i/o bilinear equation

$$
\begin{aligned}
y^{+++} &= 1.3187y^{++} - 0.2214y^{+} - 0.1474y - 8.6337u^{++} + \\
&\quad + 2.9234u^{+} + 1.2493u - 0.0858y^{++}u^{++} + 0.0050y^{+}u^{++} + \\
&\quad + 0.0602y^{+}u^{+} + 0.0035yu^{++} - 0.0281yu + 0.0107yu, \qquad (3.23)
\end{aligned}
$$

where the output $y := y(t)$ and the input $u := u(t)$ of the model denote the temperature and the cooling water flow rate, respectively.

By inspection of equation (3.23) one can easily check that $c_{12} = c_{13} = c_{23} = 0$. However, Proposition 3.1 states that the third-order i/o bilinear model is realizable if and only if at least one of the conditions (i)-(v) is satisfied. Thus, we may conclude that model (3.23) is not realizable in the classical state-space form. Note that in [5] the approximate state equations with five states were found.

Example 3.5 The model of a grain drying process is described by the following third-order i/o bilinear equation, see [59]

$$\begin{aligned} y^{+++} &= 1.6389y^{++} - 0.4397y^{+} - 0.1803y + 0.0019u^{++} - 0.0041u^{+} + \\ &\quad + 0.0021u - 0.0082y^{++}u^{++} - 0.0042y^{+}u^{+} - 0.0074yu. \end{aligned} \tag{3.24}$$

Next, analyzing equation (3.24), we find that $c_{12} = c_{13} = c_{21} = c_{23} = c_{31} = c_{32} = 0$ and, according to Proposition 3.1, we conclude (case (v)) that the presented model is realizable in the following classical state-space form

$$\begin{aligned} x_1^+ &= x_2 + (0.0019 - 0.0082x_1)u \\ x_2^+ &= x_3 - (0.00099 + 0.0176x_1)u \\ x_3^+ &= -0.1803x_1 - 0.4397x_2 + 1.6389x_3 - (0.00035 + 0.0327x_1)u \\ y &= x_1 \end{aligned}$$

3.2.4 Realizability conditions for quadratic models

Proposition 3.4 *The second-order quadratic system described by the i/o equation* (3.18) *with* $n = 2$ *is realizable in the classical state-space form if one of the following conditions is satisfied:*

(i) $a_2 = c_{21} = c_{22} = d_{12} = d_{22} = 0$;

(ii) $b_2 = c_{12} = c_{22} = e_{12} = e_{22} = 0$;

(iii) $c_{21} = e_{12} = 0$.

Proof: see Appendix.

From the results of [60], applied to the second-order quadratic systems, one may obtain alternative sufficient conditions:

(I) $a_2 = c_{21} = c_{22} = d_{12} = d_{22} = 0$;

(II) $c_{21} = e_{12} = 0$.

Comparing the above results with those in Proposition 3.4, note that (I) and (II) coincide with conditions (i) and (iii), respectively. However, we suggest an additional possibility (ii).

Remark 3.5 *From the earlier results obtained in* [58] *for the second-order bilinear i/o equation, we can conclude that conditions* (*i*) *and* (*iii*) *extend condition* $c_{21} = 0$, *and also* (*ii*) *extends* $b_2 = c_{12} = c_{22} = 0$.

Proposition 3.5 *The third-order quadratic system described by the i/o equation* (3.18) *with* $n = 3$ *is realizable in the classical state-space form if one of the following conditions is satisfied:*

(i) $a_2 = a_3 = c_{21} = c_{22} = c_{23} = c_{31} = c_{32} = c_{33} = d_{12} = d_{13} = d_{22} = d_{23} = d_{33} = 0$;

(ii) $a_3 = c_{21} = c_{31} = c_{32} = c_{33} = d_{13} = d_{23} = d_{33} = e_{13} = 0$;

(iii) $b_1 = c_{11} = c_{21} = c_{31} = c_{32} = e_{11} = e_{12} = e_{13} = e_{23} = 0$;

(iv) $b_2 = b_3 = c_{12} = c_{13} = c_{22} = c_{23} = c_{32} = c_{33} = e_{12} = e_{13} = e_{22} = e_{23} = e_{33} = 0$;

(v) $b_3 = c_{13} = c_{23} = c_{31} = c_{33} = e_{12} = e_{13} = e_{23} = e_{33} = 0$;

(vi) $c_{13} = c_{21} = c_{31} = c_{32} = d_{13} = e_{12} = e_{13} = e_{23} = 0$.

Proof: see Appendix.

By analogy with the previous proposition, from the results of [60], applied to the third-order quadratic systems, one may obtain alternative sufficient conditions:

(I) $a_2 = a_3 = c_{21} = c_{22} = c_{23} = c_{31} = c_{32} = c_{33} = d_{12} = d_{13} = d_{22} = d_{23} = d_{33} = 0$;

(II) $a_3 = c_{21} = c_{31} = c_{32} = c_{33} = d_{13} = d_{23} = d_{33} = e_{13} = 0$;

(III) $b_1 = c_{11} = c_{21} = c_{31} = c_{32} = e_{11} = e_{12} = e_{13} = e_{23} = 0$;

(IV) $c_{13} = c_{21} = c_{31} = c_{32} = d_{13} = e_{12} = e_{13} = e_{23} = 0$.

Comparing the above results with those in Proposition 3.5, note that (I)-(IV) coincide with conditions (i)-(iii) and (vi). However, we suggest additional possibilities (iv) and (v).

Remark 3.6 *Note that all conditions* $(i)-(vi)$ *from Proposition* 3.5 *extend conditions for the third-order bilinear i/o equation presented in Proposition* 3.1. *However, in the bilinear case condition* (i) *becomes the special case of* (ii).

Remark 3.7 *Note that the complete list of the necessary and sufficient conditions would involve complex relations between different coefficients of* (3.18)*; for instance, calculating the elements of the subspace* $\mathcal{H}_3$ *in Proposition* 3.4*, we get the relations* $2a_2e_{22} - b_2c_{22} = 0$ *and* $4d_{22}e_{22} - c_{22}^2 = 0$. *From the identification point of view, these conditions seem artificial, since there is no reason to assume the relations between the coefficients to hold. For that reason we suggested the special cases of i/o equations given in Propositions* 3.4 *and* 3.5 *to be used for modeling purposes.*

Example 3.6 Consider the model of a hydraulically actuated electronic unit injection (HEUI) system from [65]

$$
\begin{aligned}
y^{++} = 0.67421y^{+} + 0.35097 \times 10^{-5}y^2 - 0.31181y + \\
+ 6.6372\left(u^{+}\right)^2 + 86.549u^{+} + 10.661u^2 - 0.53194 \times 10^{-2}y^{+}u + \\
+ 0.01297yu^{+} - 23.182u^{+}u + 0.10221 \times 10^5.
\end{aligned}
$$

Notice that though the presented model contains the constant term, the change of variables $y = \tilde{y} + 16030.4$ allows us to convert the system into the form (3.18)

$$
\begin{aligned}
\tilde{y}^{++} = 0.67421\tilde{y}^{+} - 0.31181\tilde{y} + 294.464u^{+} - 85.216u - \\
- 0.0053\tilde{y}^{+}u + 0.01297\tilde{y}u^{+} + 3.5097 \times 10^{-6}\tilde{y}u + \\
+ 6.6372\left(u^{+}\right)^2 - 23.182u^{+}u + 10.661u^2, \quad (3.25)
\end{aligned}
$$

where the output $\tilde{y}$ and the input u of the model denote the fuel rate and the injection pressure, respectively.

By simple inspection of equation (3.25) one can easily check that $b_2 = c_{11} = d_{11} = d_{12} = d_{21} = d_{22} = e_{21} = 0$. However, relying on the result presented in Proposition 3.4, we may conclude that the model (3.25) is not realizable in the classical state-space form.

3.3 Application of realization theory of linear parameter varying systems to bilinear models

In this section the basic theory of the linear parameter varying systems is recalled. After that, the applicability of the LPV tools to the realization problem of nonlinear input-output systems is analyzed.

3.3.1 Introduction

The standard approach in nonlinear control is to work with differentials of the system equations, i.e with the so-called 'tangent linearized system' equations rather than with nonlinear equations themselves [23]. This approach, though well-suited for checking generic necessary and sufficient solvability conditions of various control problems, has some drawbacks. Namely, to find the control law to be implemented, or equivalent reduced system equations, for example, at the last step one has to integrate the differential one-forms to get back to the equations level. In general, the integration of (integrable in principle) differential one-forms is known to be a difficult task.

In the linear parameter varying systems, relations between signals are considered to be linear, but the model parameters are assumed to be functions of time-varying signals, the so-called scheduling variable $\boldsymbol{p}$. It is suggested in numerous papers [67, 90, 95, 96] that the LPV approach may provide a complimentary, and sometimes easier way to tackle the nonlinear control problems, since by the result of this parameter variation, the class of LPV systems may describe both time-varying and nonlinear phenomena. For example, in [95, 96] a nonlinear system is embedded globally into a linear time-varying system (LTV). This is done by replacing certain appropriate independent functions of inputs, outputs and their derivatives by *free* time-varying parameters. Then the system properties like stability and controllability have been analyzed using the skew polynomial methods developed for LTV systems and it is claimed that the *structural* results are applicable also for the original nonlinear systems. By structural properties one means the properties that are valid for all parameter $\boldsymbol{p}$ values. As for accessibility, it is shown in [96] that the structural accessibility of the LTV model implies the accessibility of the original nonlinear system but not vice versa. As for stability, the original nonlinear model is stable if the LTV model is structurally stable. Again, the converse is not necessarily true, see [96] for details.

The main goal of this section is to study when the tools, worked out in [90] for realizability of LPV systems, are applicable for nonlinear control systems.

A basic issue in the conversion a nonlinear problem into an LPV problem is the choice of a scheduling signal $\boldsymbol{p}$. In principle, the proof of realization property of the LPV system relies on the assumption that $\boldsymbol{p}$ has to be considered as an independent signal of the control system. Of course, in converting a nonlinear system description into the LPV system description, this assumption is violated. Next, we study two simple examples which allow to explain in plain words the problem under consideration.

Consider two second-order bilinear i/o equations,

$$y^{++} = y^{+}u^{+} + y^{+}u \tag{3.26}$$

and

$$y^{++} = y^{+}u + yu^{+}. \tag{3.27}$$

According to the nonlinear realization theory presented in [58, 64], the model (3.26) is realizable, whereas model (3.27) is not. Next, we find an LPV state-space representation for model (3.26) following [90]. Suppose that $\boldsymbol{p} := y$ and $\sigma(p) = \boldsymbol{p}^{+} := y^{+}$. Then, taking $x_1 = y$ and $x_2 = y^{+} - \boldsymbol{p}u$, we obtain

$$\begin{aligned} x_1^+ &= x_2 + \boldsymbol{p}u \\ x_2^+ &= \boldsymbol{p}^+ u \\ y &= x_1 \end{aligned} \tag{3.28}$$

After that, substituting $\boldsymbol{p} = y = x_1$ and $\boldsymbol{p}^+ = y^+ = x_1^+ = x_2 + x_1 u$ into (3.28), the state equations are

$$\begin{aligned} x_1^+ &= x_2 + x_1 u \\ x_2^+ &= (x_2 + x_1 u)u \\ y &= x_1 \end{aligned}$$

Consider now an LPV state-space representation of the model (3.27). Suppose again that $\boldsymbol{p} := y$. Then, taking $x_1 = y$ and $x_2 = y^+ - \boldsymbol{p}^- u$, we obtain

$$\begin{aligned} x_1^+ &= x_2 + \boldsymbol{p}^- u \\ x_2^+ &= \boldsymbol{p}^+ u \\ y &= x_1 \end{aligned} \tag{3.29}$$

Finally, substituting $\boldsymbol{p} = y = x_1$ and $\boldsymbol{p}^- = x_1^-$ into (3.29), the state equations can be rewritten as

$$\begin{aligned} x_1^+ &= x_2 + x_1^- u \\ x_2^+ &= x_1^+ u \\ y &= x_1 \end{aligned}$$

which are not in the classical state-space form because of the term x_1^-.

In fact, a given nonlinear system description can be transformed into the LPV form in several ways. The freedom comes from the different possibilities to choose the scheduling variable $\boldsymbol{p}$. For example, in the model (3.26), the alternative choice for $\boldsymbol{p}$ would be $\boldsymbol{p} := u$. After completing the computations like above, we get $x_1 = y$, $x_2 = y^+ - (\boldsymbol{p}^- + \boldsymbol{p})y$, and the state-space representation

$$\begin{aligned} x_1^+ &= ux_1 + u^- x_1 + x_2 \\ x_2^+ &= 0 \\ y &= x_1 \end{aligned}$$

is not in the classical state-space form.

3.3.2 Relationship between realizability of the i/o bilinear equations and their LPV models

The results on the realization problem of LPV systems are recalled from [90]. The LPV system in the i/o form can be described by equation

$$y^{[n]} = \sum_{i=0}^{n-1} a_i(\boldsymbol{p}) y^{[i]} + \sum_{j=0}^{s} b_j(\boldsymbol{p}) u^{[j]}, \tag{3.30}$$

where a_i, b_j are functions on $\boldsymbol{p}$.

Proposition 3.6 ([13]) *The state coordinates for i/o equation* (3.30) *are defined as*

$$x_l = \sigma_c^{-l} \begin{bmatrix} \sum_{i=0}^{n-1} a_i(\boldsymbol{p}) z^i & \sum_{j=0}^{s} b_j(\boldsymbol{p}) z^j \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix}, \tag{3.31}$$

where $l = 1, \ldots, n$.

Note that in [90] the different notations are used to represent polynomials and cut-and-shift operator. Also note the similarity of the formulas (3.31) and (3.19). Instead of y and u the expression (3.19) includes the differentials $\mathrm{d}y(t)$ and $\mathrm{d}u(t)$. The formula (3.31) gives state coordinates, whereas (3.19) yields a set of one-forms, which should be integrated to obtain state coordinates.

It should be mentioned that, according to [90], an LPV i/o equation is always realizable in the state-space form. To keep the study simple, we restrict our attention to the case of the second-order discrete-time bilinear control systems. We demonstrate on the basis of the second-order bilinear model that if the bilinear i/o equation is realizable, according to nonlinear realization theory, then there exists a suitable parameterization in the LPV framework that lead to a state-space realization, though not all parameterizations lead to classical state equations. However, if the bilinear i/o equation is not realizable, no such parameterizations exist.

Proposition 3.7 ([58]) *The second-order bilinear system, described by the i/o equation*

$$\begin{aligned} y^{++} = {} & a_1 y^{+} + a_2 y + b_1 u^{+} + b_2 u + \\ & + c_{11} y^{+} u^{+} + c_{12} y^{+} u + c_{21} y u^{+} + c_{22} y u \end{aligned} \tag{3.32}$$

is realizable in the classical state-space form iff one of the following conditions is satisfied:

(i) $c_{21} = 0$;

(ii) $b_2 = c_{12} = c_{22} = 0$.

Since our aim is to study realizability of the second-order bilinear systems (3.32) using the LPV approach, we first have to transform equation (3.32) into the LPV form as described in [89]. Equation (3.32) includes four nonlinear terms: $c_{11}y^+u^+, c_{12}y^+u, c_{21}yu^+$ and $c_{22}yu$, each being a product of two variables. Thus, we have $2^4 = 16$ different possibilities to choose parameters. Below the state equations for all these parameterizations are presented. In order to illustrate the basic idea of the realization procedure, consider in detail one specific parametrization.

Let in the terms $c_{11}y^+u^+$ and $c_{12}y^+u$ the variable $y^+ := \boldsymbol{p}_1^+$, in $c_{21}yu^+$ let $y := \boldsymbol{p}_1$ and in $c_{22}yu$ let $u := \boldsymbol{p}_2$. Then the respective LPV system has the form

$$
\begin{aligned}
y^{++} = a_1y^+ + a_2y + b_1u^+ + b_2u+ \\
+ c_{11}\boldsymbol{p}_1^+u^+ + c_{12}\boldsymbol{p}_1^+u + c_{21}\boldsymbol{p}_1u^+ + c_{22}\boldsymbol{p}_2y.
\end{aligned}
$$

Rewriting the above equation in the polynomial form (2.11) as

$$
\left[z^2 - a_1z - (a_2 + c_{22}\boldsymbol{p}_2)\right] y - \left[\left(b_1 + c_{11}\boldsymbol{p}_1^+ + c_{21}\boldsymbol{p}_1\right) z + b_2 + c_{12}\boldsymbol{p}_1^+\right] u = 0
$$

allows to find the state coordinates by Proposition 3.6

$$
\begin{aligned}
x_1 &= y \\
x_2 &= y^+ - a_1y - \left(b_1 + c_{11}\boldsymbol{p}_1 + c_{21}\boldsymbol{p}_1^-\right) u
\end{aligned}
$$

and the state equations

$$
\begin{aligned}
x_1^+ &= b_1u + c_{11}\boldsymbol{p}_1u + c_{21}\boldsymbol{p}_1^-u + a_1x_1 + x_2 \\
x_2^+ &= b_2u + c_{12}\boldsymbol{p}_1^+u + a_2x_1 + c_{22}\boldsymbol{p}_2x_1 \\
y &= x_1
\end{aligned}
$$

After replacing parameters regarding that

$$
\begin{aligned}
\boldsymbol{p}_1^- &= y^- = x_1^-, \\
\boldsymbol{p}_1 &= y = x_1, \\
\boldsymbol{p}_1^+ &= y^+ = x_1^+ = b_1u + c_{11}x_1u + c_{21}x_1^-u + a_1x_1 + x_2, \\
\boldsymbol{p}_2 &= u,
\end{aligned}
$$

we obtain

$$
\begin{aligned}
x_1^+ &= b_1u + c_{11}x_1u + c_{21}x_1^-u + a_1x_1 + x_2 \\
x_2^+ &= b_2u + a_2x_1 + c_{22}ux_1 \\
&\quad + c_{12}\left(b_1u + c_{11}x_1u + c_{21}x_1^-u + a_1x_1 + x_2\right) u \\
y &= x_1
\end{aligned}
\tag{3.33}
$$

Clearly, the latter system is not in the classical state-space form due to x_1^-, which cannot be eliminated.

Using the different parameterizations of the i/o equation (3.32), we may alternatively reach to the following three representations:

$$\begin{aligned} x_1^+ &= b_1u + a_1x_1 + c_{11}ux_1 + x_2 \\ x_2^+ &= b_2u + a_2x_1 + c_{22}ux_1 + c_{21}u^+x_1 + \\ &\quad + c_{12}(b_1u + a_1x_1 + c_{11}ux_1 + x_2)u \\ y &= x_1 \end{aligned} \tag{3.34}$$

$$\begin{aligned} x_1^+ &= b_1u + a_1x_1 + c_{12}u^-x_1 + c_{11}ux_1 + x_2 \\ x_2^+ &= b_2u + a_2x_1 + c_{22}ux_1 + c_{21}u^+x_1 \\ y &= x_1 \end{aligned} \tag{3.35}$$

and

$$\begin{aligned} x_1^+ &= b_1u + c_{21}ux_1^- + a_1x_1 + c_{12}u^-x_1 + c_{11}ux_1 + x_2 \\ x_2^+ &= b_2u + a_2x_1 + c_{22}ux_1 \\ y &= x_1 \end{aligned} \tag{3.36}$$

Note that none of (3.33)-(3.36) is in the classical state-space form. Table 3.2 shows which state equations correspond to different choices of scheduling variables.

The conclusion is that for system (3.32), in general, does not exist parametrization yielding classical state equations.

Next, we separately study the realizable cases specified by Proposition 3.7.

Realizable case (i). Suppose that $c_{21} = 0$ in (3.32), then

$$y^{++} = a_1y^+ + a_2y + b_1u^+ + b_2u + c_{11}y^+u^+ + c_{12}y^+u + c_{22}yu. \tag{3.37}$$

Since equation (3.37) contains only three nonlinear terms, there are in total $2^3 = 8$ different choices for parameters. Application of the realization procedure for all parameterizations provides two different state equations:

$$\begin{aligned} x_1^+ &= b_1u + a_1x_1 + c_{11}ux_1 + x_2 \\ x_2^+ &= b_2u + a_2x_1 + c_{22}ux_1 + \\ &\quad + c_{12}(b_1u + a_1x_1 + c_{11}ux_1 + x_2)u \\ y &= x_1 \end{aligned} \tag{3.38}$$

and

$$\begin{aligned} x_1^+ &= b_1u + a_1x_1 + c_{12}u^-x_1 + c_{11}ux_1 + x_2 \\ x_2^+ &= b_2u + a_2x_1 + c_{22}ux_1 \\ y &= x_1 \end{aligned} \tag{3.39}$$

Table 3.2: Parameterizations and realizations for system (3.32)

parameterization of the nonlinear part	state equations
$c_{11}u^+\boldsymbol{p}^+ + c_{12}u\boldsymbol{p}^+ + c_{21}u^+\boldsymbol{p} + c_{22}u\boldsymbol{p}$	(3.33)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}u\boldsymbol{p}_1^+ + c_{21}u^+\boldsymbol{p}_1 + c_{22}y\boldsymbol{p}_2$	(3.33)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}u\boldsymbol{p}_1^+ + c_{21}y\boldsymbol{p}_2^+ + c_{22}u\boldsymbol{p}_1$	(3.34)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}u\boldsymbol{p}_1^+ + c_{21}y\boldsymbol{p}_2^+ + c_{22}y\boldsymbol{p}_2$	(3.34)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}y^+\boldsymbol{p}_2 + c_{21}u^+\boldsymbol{p}_1 + c_{22}u\boldsymbol{p}_1$	(3.36)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}y^+\boldsymbol{p}_2 + c_{21}u^+\boldsymbol{p}_1 + c_{22}y\boldsymbol{p}_2$	(3.36)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}y^+\boldsymbol{p}_2 + c_{21}y\boldsymbol{p}_2^+ + c_{22}u\boldsymbol{p}_1$	(3.35)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}y^+\boldsymbol{p}_2 + c_{21}y\boldsymbol{p}_2^+ + c_{22}y\boldsymbol{p}_2$	(3.35)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}u\boldsymbol{p}_1^+ + c_{21}u^+\boldsymbol{p}_1 + c_{22}u\boldsymbol{p}_1$	(3.33)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}u\boldsymbol{p}_1^+ + c_{21}u^+\boldsymbol{p}_1 + c_{22}y\boldsymbol{p}_2$	(3.33)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}u\boldsymbol{p}_1^+ + c_{21}y\boldsymbol{p}_2^+ + c_{22}u\boldsymbol{p}_1$	(3.34)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}u\boldsymbol{p}_1^+ + c_{21}y\boldsymbol{p}_2^+ + c_{22}y\boldsymbol{p}_2$	(3.34)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}y^+\boldsymbol{p}_2 + c_{21}u^+\boldsymbol{p}_1 + c_{22}u\boldsymbol{p}_1$	(3.36)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}y^+\boldsymbol{p}_2 + c_{21}u^+\boldsymbol{p}_1 + c_{22}y\boldsymbol{p}_2$	(3.36)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}y^+\boldsymbol{p}_2 + c_{21}y\boldsymbol{p}_2^+ + c_{22}u\boldsymbol{p}_1$	(3.35)
$c_{11}y^+\boldsymbol{p}^+ + c_{12}y^+\boldsymbol{p} + c_{21}y\boldsymbol{p}^+ + c_{22}y\boldsymbol{p}$	(3.35)

Table 3.3: Parameterizations and realizations for system (3.37)

parameterization of the nonlinear part	state equations
$c_{11}u^+\boldsymbol{p}^+ + c_{12}u\boldsymbol{p}^+ + c_{22}u\boldsymbol{p}$	(3.38)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}u\boldsymbol{p}_1^+ + c_{22}y\boldsymbol{p}_2$	(3.38)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}\boldsymbol{p}_2y^+ + c_{22}u\boldsymbol{p}_1$	(3.39)
$c_{11}u^+\boldsymbol{p}_1^+ + c_{12}\boldsymbol{p}_2y^+ + c_{22}y\boldsymbol{p}_2$	(3.39)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}u\boldsymbol{p}_1^+ + c_{22}u\boldsymbol{p}_1$	(3.38)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}u\boldsymbol{p}_1^+ + c_{22}y\boldsymbol{p}_2$	(3.38)
$c_{11}y^+\boldsymbol{p}_2^+ + c_{12}\boldsymbol{p}_2y^+ + c_{22}u\boldsymbol{p}_1$	(3.39)
$c_{11}y^+\boldsymbol{p}^+ + c_{12}\boldsymbol{p}y^+ + c_{22}y\boldsymbol{p}$	(3.39)

Note that equations (3.38) are in the classical state-space form, whereas equations (3.39) are not because of the term u^-. Table 3.3 shows which state equations correspond to every possible choice of parameters.

We have shown that for the nonlinear system (3.37) there exist parameterizations, which allow to transform this system into the classical state-space form using the LPV approach. Unfortunately, not all parameterizations yield the classical state equations and there is no rule to choose the right parametrization *a priori*.

Realizable case (ii). Suppose that $b_2 = c_{12} = c_{22} = 0$ in (3.32), then

$$y^{++} = a_1 y^+ + a_2 y + b_1 u^+ + c_{11} y^+ u^+ + c_{21} y u^+. \tag{3.40}$$

Equation (3.40) has two nonlinear terms, therefore there are $2^2 = 4$ possibilities for parameterization, which yields two different state equations:

$$\begin{aligned} x_1^+ &= b_1 u + a_1 x_1 + c_{11} u x_1 + x_2 \\ x_2^+ &= a_2 x_1 + c_{21} u^+ x_1 \\ y &= x_1 \end{aligned} \tag{3.41}$$

and

$$\begin{aligned} x_1^+ &= b_1 u + c_{21} u x_1^- + a_1 x_1 + c_{11} u x_1 + x_2 \\ x_2^+ &= a_2 x_1 \\ y &= x_1 \end{aligned} \tag{3.42}$$

Neither of these equations is in the classical state-space form. Table 3.4 shows the correspondence between possible parameterizations and state equations.

Table 3.4: Parameterizations and realizations for system (3.40)

parameterization of the nonlinear part	state equations
$c_{11} u^+ \boldsymbol{p}^+ + c_{21} u^+ \boldsymbol{p}$	(3.42)
$c_{11} u^+ \boldsymbol{p}_1^+ + c_{21} y \boldsymbol{p}_2^+$	(3.41)
$c_{11} y^+ \boldsymbol{p}_2^+ + c_{21} u^+ \boldsymbol{p}_1$	(3.42)
$c_{11} y^+ \boldsymbol{p}^+ + c_{21} y \boldsymbol{p}^+$	(3.41)

To conclude, the LPV approach is not a proper tool to find the state-space realization of the i/o equation (3.40), because none of the possible choices of parameters yields the classical state equations.

3.4 On realization of nonlinear MIMO continuous-time equations

In this section the polynomial method is applied to solve the realization problem of nonlinear multi-input multi-output continuous-time systems.

It should be mentioned that the proposed algorithm combines well with the existing results for the reduction problem [54]. Both the results of [54] and those of this section rely on system description in terms of two polynomial matrices. Thus, the basic different theoretical points between the mathematical tools used for the discrete- and continuous-time systems are explained.

3.4.1 Introduction

Consider a nonlinear MIMO continuous-time system, described by a set of higher order i/o differential equations, relating the inputs u_υ, $\upsilon = 1, \ldots, m$, the outputs y_ν, $\nu = 1, \ldots, p$ and a finite number of their time derivatives

$$
\begin{aligned}
y_i^{(n_i)} = \phi_i\Big(y_\nu, \dot{y}_\nu, \ldots, y_\nu^{(n_{i\nu})}, \nu = 1, \ldots, p,& \\
u_\upsilon, \dot{u}_\upsilon, \ldots, u_\upsilon^{(s_{i\upsilon})}, \upsilon = 1, \ldots, m\Big)& \qquad (3.43)
\end{aligned}
$$

for $i = 1, \ldots, p$. In (3.43) $u = [u_1, \ldots, u_m]^T \in \mathbb{R}^m$, $y = [y_1, \ldots, y_p]^T \in \mathbb{R}^p$ and ϕ_i are real analytic functions. Define $n := n_1 + \cdots + n_p$ and $s := \max\{s_{i\upsilon}, i = 1, \ldots, p, \upsilon = 1, \ldots, m\}$. Moreover, we assume that the following assumptions hold for system (3.43).

Assumption 3.1 *System* (3.43) *is strictly proper, i.e.* $s_{i\upsilon} < n_i$.

Assumption 3.2 *System* (3.43) *is in the canonical form*[2], *which means that* $n_{i\nu} < \min\{n_i, n_\nu\}$.

Note that if the system under consideration is not in the form (3.43), then it can be transformed into (3.43) using the approach proposed in [92], at least locally.

3.4.2 Main tools

Here, we adopt the algebraic and polynomial formalisms presented in Sections 2.2 and 2.3, respectively, for the case of nonlinear multi-input multi-output continuous-time equations.

By analogy with Section 2.2, let $\mathcal{K}$ denote the field of meromorphic functions in a finite number of the independent system variables from the set

$$
\mathcal{C} = \left\{y_i, \dot{y}_i, \ldots, y_i^{(n_i-1)}, i = 1, \ldots, p, u_\upsilon^{(l_\upsilon)}, \upsilon = 1, \ldots, m, l_\upsilon \geq 0\right\},
$$

[2]The form (3.43) is an extension of the Guidorzi canonical form, introduced in [7] for linear systems. The other forms, like Hermite or Popov forms, may also be used.

and let $\delta : \mathcal{K} \to \mathcal{K}$ be the time derivative operator $\frac{\mathrm{d}}{\mathrm{d}t}$. Then the pair $(\mathcal{K}, \delta)$ is a *differential field*, see [46] for details. Over the field $\mathcal{K}$ one can define a differential vector space, $\mathcal{E} := \mathrm{span}_{\mathcal{K}} \{\mathrm{d}\varphi \mid \varphi \in \mathcal{K}\}$ spanned by the differentials of the elements of $\mathcal{K}$. Consider a one-form $\omega \in \mathcal{E}$ such that $\omega = \sum_i \alpha_i \mathrm{d}\varphi_i$, $\alpha_i, \varphi_i \in \mathcal{K}$. Its derivative $\dot{\omega}$ is defined by $\dot{\omega} = \sum_i (\dot{\alpha}_i \mathrm{d}\varphi_i + \alpha_i \mathrm{d}\dot{\varphi}_i)$.

A sequence $\{\mathcal{H}_k\}_{k=1}^{\infty}$ of subspaces of $\mathcal{E}$ is defined by

$$\begin{aligned} \mathcal{H}_1 &= \mathrm{span}_{\mathcal{K}} \Big\{ \mathrm{d}y_i, \mathrm{d}\dot{y}_i, \ldots, \mathrm{d}y_i^{(n_i-1)}, i = 1, \ldots, p, \\ &\qquad \mathrm{d}u_\upsilon, \mathrm{d}\dot{u}_\upsilon, \ldots, \mathrm{d}u_\upsilon^{(s)}, \upsilon = 1, \ldots, m \Big\}, \\ \mathcal{H}_{k+1} &= \{\omega \in \mathcal{H}_k \mid \dot{\omega} \in \mathcal{H}_k\}, \quad k \geq 1. \end{aligned} \tag{3.44}$$

The necessary and sufficient conditions for the system of the form (3.43) to be realizable in the classical state-space form are presented in the following theorem.

Theorem 3.5 ([23]) *The nonlinear system, described by the set of irreducible i/o differential equations* (3.43)*, has an observable and accessible state-space realization iff for* $1 \leq k \leq s+2$ *the subspaces* $\mathcal{H}_k$ *defined by* (3.44) *are completely integrable. Moreover, the state coordinates can be obtained by integrating the basis vectors of* $\mathcal{H}_{s+2}$.

Next, the polynomial formalism for the case of MIMO continuous-time systems is presented. The differential field $(\mathcal{K}, \delta)$ induces a ring of the left differential polynomials, which is denoted by $\mathcal{K}[z; \delta]$ with respect to Remark 3.1. Note that for $\alpha \in \mathcal{K}$ the multiplication is defined by

$$z \cdot \alpha := \alpha \cdot z + \delta(\alpha), \tag{3.45}$$

which is different compared to the discrete-time case, but directly follows from (3.5) for $\sigma = \mathrm{id}_{\mathcal{K}}$, $\delta = \mathrm{d}/\mathrm{d}t$, and $\theta = \delta$.

Using notations and definitions presented above, the nonlinear system (3.43) can be represented in terms of two polynomial matrices as

$$P(z)\mathrm{d}y + Q(z)\mathrm{d}u = 0, \tag{3.46}$$

where $P(z)$ and $Q(z)$ are $p \times p$ and $p \times m$-dimensional matrices, respectively, whose elements $p_{i\nu}(z), q_{i\upsilon}(z) \in \mathcal{K}[z; \delta]$ and

$$p_{i\nu}(z) = \begin{cases} z^{n_i} - \sum_{\alpha=0}^{n_{i\nu}} p_{i\nu,\alpha} z^\alpha, & \text{if } i = \nu, \\ -\sum_{\alpha=0}^{n_{i\nu}} p_{i\nu,\alpha} z^\alpha, & \text{if } i \neq \nu, \end{cases}$$

$$q_{i\upsilon}(z) = -\sum_{\beta=0}^{s_{i\upsilon}} q_{i\upsilon,\beta} z^\beta,$$

where $p_{i\nu,\alpha} = \frac{\partial \phi_i}{\partial y_\nu^{(\alpha)}} \in \mathcal{K}$, $q_{i\upsilon,\beta} = \frac{\partial \phi_i}{\partial u_\upsilon^{(\beta)}} \in \mathcal{K}$. Further, the notations $p_{i\cdot}(z) := [p_{i1}(z), \ldots, p_{ip}(z)]$ and $q_{i\cdot}(z) := [q_{i1}(z), \ldots, q_{im}(z)]$ are used for row vectors of $P(z)$ and $Q(z)$, respectively.

3.4.3 Realization

Now, we introduce the certain one-forms in terms of which the main result of this section will be formulated. Let

$$\omega_{i,l} = \begin{bmatrix} p_{i\cdot,l}(z) & q_{i\cdot,l}(z) \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} \tag{3.47}$$

for $i = 1, \ldots, p$, $l = 1, \ldots, n_i$, where $p_{i\cdot,l}(z)$ and $q_{i\cdot,l}(z)$ are Ore polynomials, which can be recursively calculated from the equalities

$$\begin{aligned} p_{i\cdot,l-1}(z) &= z \cdot p_{i\cdot,l}(z) + \xi_{i\cdot,l}, \quad \deg \xi_{i\cdot,l} = 0, \\ q_{i\cdot,l-1}(z) &= z \cdot q_{i\cdot,l}(z) + \gamma_{i\cdot,l}, \quad \deg \gamma_{i\cdot,l} = 0 \end{aligned} \tag{3.48}$$

with initial polynomials $p_{i\cdot,0}(z) := p_{i\cdot}(z)$ and $q_{i\cdot,0}(z) := q_{i\cdot}(z)$.

Theorem 3.6 *For the input-output model* (3.43)*, the subspaces $\mathcal{H}_k$ may be calculated as*

$$\begin{aligned} \mathcal{H}_k = \mathrm{span}_{\mathcal{K}} \Big\{ & \omega_{i,l}, i = 1, \ldots, p, l = 1, \ldots, n_i, \\ & \mathrm{d}u_\upsilon, \ldots, \mathrm{d}u_\upsilon^{(s-k+1)}, \upsilon = 1, \ldots, m \Big\} \end{aligned} \tag{3.49}$$

for $k = 1, \ldots, s+1$ and

$$\mathcal{H}_{s+2} = \mathrm{span}_{\mathcal{K}} \{\omega_{i,l}, i = 1, \ldots, p, l = 1, \ldots, n_i\}. \tag{3.50}$$

Proof: see Appendix.

To illustrate the effectiveness of the described approach, consider the following examples.

Example 3.7 Consider the system

$$\begin{aligned} \ddot{y}_1 &= u_2 \dot{y}_1 + \dot{u}_1 y_2 \\ y_2^{(3)} &= -u_1 \dot{y}_1 + y_1 \dot{y}_2 - \ddot{u}_2 \end{aligned} \tag{3.51}$$

that can be described by two polynomial matrices in the following way

$$P(z) = \begin{pmatrix} z^2 - u_2 z & -\dot{u}_1 \\ u_1 z - \dot{y}_2 & z^3 - y_1 z \end{pmatrix}$$

and

$$Q(z) = \begin{pmatrix} -y_2 z & -\dot{y}_1 \\ \dot{y}_1 & z^2 \end{pmatrix}.$$

From (3.51) one can get that $n_1 = 2, n_2 = 3, n_{11} = 1, n_{12} = 0, n_{21} = 1, n_{22} = 1$ and $s_{11} = 1, s_{12} = 0, s_{21} = 0, s_{22} = 2$. Thus, for system (3.51), $n = n_1 + n_2 = 5$ and $s = \max\{s_{11}, s_{12}, s_{21}, s_{22}\} = 2$. Compute, according to (3.48), sequences of the left quotients of each element in matrices $P(z)$ and $Q(z)$ as

$$\begin{aligned}
\begin{bmatrix} p_{11,0}(z) & p_{12,0}(z) & q_{11,0}(z) & q_{12,0}(z) \end{bmatrix} &= \begin{bmatrix} z^2 - u_2 z & -\dot{u}_1 & -y_2 z & -\dot{y}_1 \end{bmatrix}, \\
\begin{bmatrix} p_{11,1}(z) & p_{12,1}(z) & q_{11,1}(z) & q_{12,1}(z) \end{bmatrix} &= \begin{bmatrix} z - u_2 & 0 & -y_2 & 0 \end{bmatrix}, \\
\begin{bmatrix} p_{11,2}(z) & p_{12,2}(z) & q_{11,2}(z) & q_{12,2}(z) \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
\begin{bmatrix} p_{21,0}(z) & p_{22,0}(z) & q_{21,0}(z) & q_{22,0}(z) \end{bmatrix} &= \begin{bmatrix} u_1 z - \dot{y}_2 & z^3 - y_1 z & \dot{y}_1 & z^2 \end{bmatrix}, \\
\begin{bmatrix} p_{21,1}(z) & p_{22,1}(z) & q_{21,1}(z) & q_{22,1}(z) \end{bmatrix} &= \begin{bmatrix} u_1 & z^2 - y_1 & 0 & z \end{bmatrix}, \\
\begin{bmatrix} p_{21,2}(z) & p_{22,2}(z) & q_{21,2}(z) & q_{22,2}(z) \end{bmatrix} &= \begin{bmatrix} 0 & z & 0 & 1 \end{bmatrix}, \\
\begin{bmatrix} p_{21,3}(z) & p_{22,3}(z) & q_{21,3}(z) & q_{22,3}(z) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

Further, recall that $\mathrm{d}y = [\mathrm{d}y_1, \mathrm{d}y_2]^T$, $\mathrm{d}u = [\mathrm{d}u_1, \mathrm{d}u_2]^T$. Since $s = 2$, using (3.47), the elements $\omega_{i,j}$, $i = 1, 2$, $j = 1, \ldots, n_i$ of the subspace of the one-forms $\mathcal{H}_{s+2} = \mathcal{H}_4$ can be represented in the following form

$$\begin{aligned}
\omega_{1,1} &= \begin{bmatrix} z - u_2 & 0 & -y_2 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}\dot{y}_1 - u_2 \mathrm{d}y_1 - y_2 \mathrm{d}u_1, \\
\omega_{1,2} &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}y_1, \\
\omega_{2,1} &= \begin{bmatrix} u_1 & z^2 - y_1 & 0 & z \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = u_1 \mathrm{d}y_1 + \mathrm{d}\ddot{y}_2 - y_1 \mathrm{d}y_2 + \mathrm{d}\dot{u}_2, \\
\omega_{2,2} &= \begin{bmatrix} 0 & z & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}\dot{y}_2 + \mathrm{d}u_2, \\
\omega_{2,3} &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}y_2.
\end{aligned}$$

Though the subspace $\mathcal{H}_4$ is completely integrable, $\omega_{1,1}$ and $\omega_{2,1}$ are not exact and, according to Remark 3.3, we have to replace them by integrable linear combinations of one-forms from $\mathcal{H}_4$ to obtain the differentials of the state coordinates

$$\begin{aligned}
\mathrm{d}x_1 &= \omega_{1,2} &&= \mathrm{d}y_1, \\
\mathrm{d}x_2 &= \omega_{2,3} &&= \mathrm{d}y_2, \\
\mathrm{d}x_3 &= \omega_{1,1} + u_2\omega_{1,2} - u_1\omega_{2,3} &&= \mathrm{d}(\dot{y}_1 - u_1 y_2), \\
\mathrm{d}x_4 &= \omega_{2,2} &&= \mathrm{d}(\dot{y}_2 + u_2), \\
\mathrm{d}x_5 &= \omega_{2,1} - u_1\omega_{1,2} + y_1\omega_{2,3} &&= \mathrm{d}(\ddot{y}_2 + \dot{u}_2).
\end{aligned}$$

In these coordinates the system has the classical state-space form

$$\begin{aligned}
\dot{x}_1 &= u_1 x_2 + x_3 \\
\dot{x}_2 &= x_4 - u_2 \\
\dot{x}_3 &= u_2 x_3 + u_1(u_2(x_2+1) - x_4) \\
\dot{x}_4 &= x_5 \\
\dot{x}_5 &= x_1(x_4 - u_2) - u_1(u_1 x_2 + x_3) \\
y_1 &= x_1 \\
y_2 &= x_2
\end{aligned}$$

Example 3.8 Consider a hopping robot, consisting of a body and a single leg, that can be described by the i/o equations as in [23]

$$\begin{aligned}
\ddot{y}_1 &= \frac{u_2}{m} + y_1 \dot{y}_3^2 \\
\dot{y}_2 &= -\frac{m}{J} y_1^2 \dot{y}_3 \\
\ddot{y}_3 &= -\frac{u_1 + 2m y_1 \dot{y}_1 \dot{y}_3}{m y_1^2}
\end{aligned} \tag{3.52}$$

where m is the mass of the leg, J the inertia momentum of the body, y_1 denote the length of the leg, y_2 the angular position of the body, and y_3 the angular position of the leg. Moreover, u_1 and u_2 control the orientation of the body with respect to the leg and the length of the leg, respectively.

Like in the previous example, (3.52) can be described by two polynomial matrices as follows

$$P(z) = \begin{pmatrix} z^2 - \dot{y}_3^2 & 0 & -2y_1\dot{y}_3 z \\ \frac{2my_1\dot{y}_3}{J} & z & \frac{my_1^2}{J} z \\ \frac{2\dot{y}_3}{y_1} z - \frac{2(u_1 + my_1\dot{y}_1\dot{y}_3)}{my_1^3} & 0 & z^2 + \frac{2\dot{y}_1}{y_1} z \end{pmatrix}$$

and

$$Q(z) = \begin{pmatrix} 0 & -\frac{1}{m} \\ 0 & 0 \\ \frac{1}{my_1^2} & 0 \end{pmatrix}.$$

From (3.52), $n = 5$ and $s = 0$. Compute, according to (3.48), sequences of the left quotients of each element in matrices $P(z)$ and $Q(z)$ as

$$\begin{aligned}
\begin{bmatrix} p_{1\cdot,1}(z) & q_{1\cdot,1}(z) \end{bmatrix} &= \begin{bmatrix} z & 0 & -2y_1\dot{y}_3 & 0 & 0 \end{bmatrix}, \\
\begin{bmatrix} p_{1\cdot,2}(z) & q_{1\cdot,2}(z) \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
\begin{bmatrix} p_{2\cdot,1}(z) & q_{2\cdot,1}(z) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & \frac{my_1^2}{J} & 0 & 0 \end{bmatrix}, \\
\begin{bmatrix} p_{3\cdot,1}(z) & q_{3\cdot,1}(z) \end{bmatrix} &= \begin{bmatrix} \frac{2\dot{y}_3}{y_1} & 0 & z + \frac{2\dot{y}_1}{y_1} & 0 & 0 \end{bmatrix}, \\
\begin{bmatrix} p_{3\cdot,2}(z) & q_{3\cdot,2}(z) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

Further, recall that $\mathrm{d}y = [\mathrm{d}y_1, \mathrm{d}y_2, \mathrm{d}y_3]^T$, $\mathrm{d}u = [\mathrm{d}u_1, \mathrm{d}u_2]^T$. By (3.47), we get the following basis one-forms of the last subspace $\mathcal{H}_2$

$$
\begin{aligned}
\omega_{1,1} &= \begin{bmatrix} z & 0 & -2y_1\dot{y}_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}\dot{y}_1 - 2y_1\dot{y}_3\mathrm{d}y_3, \\
\omega_{1,2} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}y_1, \\
\omega_{2,1} &= \begin{bmatrix} 0 & 1 & \frac{my_1^2}{J} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}y_2 + \frac{my_1^2}{J}\mathrm{d}y_3, \\
\omega_{3,1} &= \begin{bmatrix} \frac{2\dot{y}_3}{y_1} & 0 & z + \frac{2\dot{y}_1}{y_1} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \frac{2\dot{y}_3}{y_1}\mathrm{d}y_1 + \mathrm{d}\dot{y}_3 + \frac{2\dot{y}_1}{y_1}\mathrm{d}y_3, \\
\omega_{3,2} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} = \mathrm{d}y_3.
\end{aligned}
$$

Finally, we get $\mathcal{H}_2 = \operatorname{span}_{\mathcal{K}} \left\{ \mathrm{d}\dot{y}_1 - 2y_1\dot{y}_3\mathrm{d}y_3, \mathrm{d}y_1, \mathrm{d}y_2 + \frac{my_1^2}{J}\mathrm{d}y_3, \frac{2\dot{y}_3}{y_1}\mathrm{d}y_1 + \mathrm{d}\dot{y}_3 + \frac{2\dot{y}_1}{y_1}\mathrm{d}\right.$ Simplifying the basis one-forms, the subspace can be rewritten as $\mathcal{H}_2 = \operatorname{span}_{\mathcal{K}}\{\mathrm{d}y_1, \mathrm{d}\dot{y}_1, \mathrm{d}y_2, \mathrm{d}y_3, \mathrm{d}\dot{y}_3\}$, which is closed. Therefore, the state equations are

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{u_2}{m} + x_1x_5^2 \\
\dot{x}_3 &= -\frac{m}{J}x_1^2x_5 \\
\dot{x}_4 &= x_5 \\
\dot{x}_5 &= -\frac{u_1 + 2mx_1x_2x_5}{mx_1^2} \\
y_1 &= x_1 \\
y_2 &= x_3 \\
y_3 &= x_4
\end{aligned}
$$

It should be mentioned that since equations (3.52) do not include derivatives of the control variables u_1, u_2, we need to integrate the elements of the subspace $\mathcal{H}_2$, which according to (3.44) is always in the obtained form, see [23] for details. In fact, we can skip intermediate computations and directly write out the state space realization of i/o equations (3.52); however, we decided to show them to demonstrate the applicability of the polynomial method.

Chapter 4

Model matching problem: transfer function approach

In this chapter the model matching problem of nonlinear single-input single-output discrete-time systems is considered. Both feedforward and feedback solutions are given. The problem is studied within the transfer function approach presented in Section 2.3. It was mentioned that in this case the system is described by the quotient of two polynomials from the Ore polynomial ring.

4.1 Feedforward compensator

Consider a nonlinear system F and a model G described by their transfer functions

$$F(z) = p_F^{-1}(z)q_F(z) \tag{4.1}$$

and

$$G(z) = p_G^{-1}(z)q_G(z), \tag{4.2}$$

respectively. Find a (proper) feedforward compensator R described by its transfer function

$$R(z) = p_R^{-1}(z)q_R(z)$$

such that the transfer function of the compensated system coincides with that of the model G, i.e.

$$G(z) = F(z)R(z),$$

or equivalently

$$R(z) = F^{-1}(z)G(z), \tag{4.3}$$

as depicted in Figure 4.1.

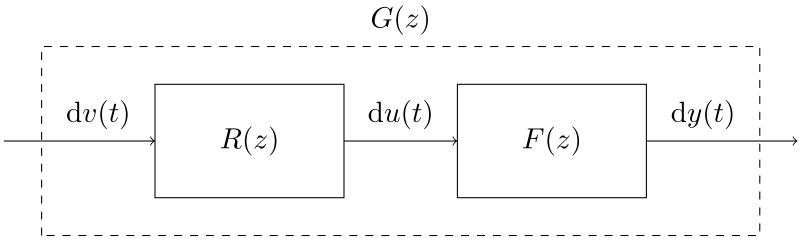

Figure 4.1: Compensated system

Proposition 4.1 *Given $F(z) \neq 0$ and $G(z)$, the feedforward model matching problem is solvable if the one-form $p_R(z)\mathrm{d}u(t) - q_R(z)\mathrm{d}v(t)$ is integrable.*

Proof: see Appendix.

Proposition 4.1 gives a weak result, because it does not define the class of nonlinear systems for which the feedforward compensator exists. In Proposition 4.2 below we specify one such subclass.

Proposition 4.2 *The one-form $p_R(z)\mathrm{d}u(t) - q_R(z)\mathrm{d}v(t)$ is always integrable if the system F and the model G are given by*

$$\begin{aligned} y(t+n_F) = f_1(y(t), y(t+1), \ldots, y(t+n_F-1)) + \\ + f_2(u(t), u(t+1), \ldots, u(t+s_F)) \end{aligned} \tag{4.4}$$

and

$$\begin{aligned} y(t+n_G) = g_1(y(t), y(t+1), \ldots, y(t+n_G-1)) + \\ + g_2(v(t), v(t+1), \ldots, v(t+s_G)), \end{aligned} \tag{4.5}$$

respectively, such that

$$\begin{aligned} p_F(z) = \gamma_F(z)\rho(z), \\ p_G(z) = \gamma_G(z)\rho(z), \end{aligned} \tag{4.6}$$

where $\gamma_F(z)$ and $\gamma_G(z)$ are polynomials with real coefficients, and $\rho(z) = \sum_{i=0}^{m} \rho_i z^{m-i}$ with $\rho_i \in \mathcal{K}^*$.

Proof: see Appendix.

In most cases one is interested in finding a solution in a class of proper compensators. Therefore, to guarantee the existence of the solution one has to introduce the restriction on the relative degree of the model G.

Proposition 4.3 *The transfer function of compensator* (4.3) *is proper (causal) if and only if*

$$\operatorname{rel\,deg} G(z) \geq \operatorname{rel\,deg} F(z). \tag{4.7}$$

Proof: see Appendix.
Example 4.1 Consider the system

$$y(t+2) = y(t) + u(t)u(t+1)$$

and compute its transfer function

$$F(z) = \left(z^2 - 1\right)^{-1} (u(t)z + u(t+1))$$

which is strictly proper. Suppose that the reference model is

$$G(z) = z^{-2}.$$

By (4.3) and (2.9), we can find the transfer function of the compensator

$$R(z) = (u(t)z + u(t+1))^{-1} \left(z^2 - 1\right) z^{-2} =$$
$$= \left(u(t+2)z^3 + u(t+3)z^2\right)^{-1} \left(z^2 - 1\right),$$

where the Ore condition $\beta\left(z^2 - 1\right) = \alpha z^2$ is satisfied for $\alpha = z^2 - 1$ and $\beta = z^2$. Note that $R(z)$ results in the integrable one-form, yielding the compensator given by the equation

$$u(t+2)u(t+3) = v(t+2) - v(t).$$

Moreover, this compensator has a classical state-space realization of the form

$$\begin{aligned} u(t) &= \eta_1(t) \\ \eta_1(t+1) &= \eta_2(t) + \frac{v(t)}{\eta_1(t)} \\ \eta_2(t+1) &= \frac{\eta_3(t)}{v(t) + \eta_1(t)\eta_2(t)} \\ \eta_3(t+1) &= -v(t)\left(\eta_2(t) + \frac{v(t)}{\eta_1(t)}\right) \end{aligned}$$

Example 4.2 Consider the system

$$y(t+2) = u(t)y(t) + u(t+1) \tag{4.8}$$

with the transfer function

$$F(z) = \left(z^2 - u(t)\right)^{-1} (z + y(t)).$$

Suppose that the reference model is

$$G(z) = z^{-2}.$$

By (4.3) and (2.9), one can find

$$R(z) = (z + y(t))^{-1} \left(z^2 - u(t)\right) z^{-2} = \\ = \left(z^3 + y(t+2)z^2\right)^{-1} \left(z^2 - u(t+2)\right), \quad (4.9)$$

where the Ore condition $\beta_2 \left(z^2 - u(t)\right) = \alpha_1 z^2$ is satisfied for $\alpha_1 = z^2 - u(t+2)$ and $\beta_2 = z^2$. Thus, the transfer function (4.9) results in the one-form

$$\mathrm{d}u(t+3) + y(t+2)\mathrm{d}u(t+2) = \mathrm{d}v(t+2) - u(t+2)\mathrm{d}v(t),$$

which after replacing $y(t+2)$ by the right-hand side of equation (4.8) yields

$$\mathrm{d}u(t+3) + (u(t+1) + u(t)y(t))\mathrm{d}u(t+2) = \\ = \mathrm{d}v(t+2) - u(t+2)\mathrm{d}v(t). \quad (4.10)$$

According to condition (2.5), the one-form (4.10) is non-integrable. The latter means that $R(z)$ in (4.9) does not correspond to any compensator R.

Thus, unlike the linear time-invariant case, a class of nonlinear systems for which the solution in terms of a feedforward compensator exists is, due to the integrability condition, quite restricted. Therefore, it is natural to look for a solution in a (more general) class of feedback compensators.

4.2 Feedback compensator

Consider a nonlinear system F and a model G described by their transfer functions

$$F(z) = p_F^{-1}(z) q_F(z) \quad (4.11)$$

and

$$G(z) = p_G^{-1}(z) q_G(z), \quad (4.12)$$

respectively, find a (proper) feedback compensator R

$$\mathrm{d}u(t) = R_v(z)\mathrm{d}v(t) + R_y(z)\mathrm{d}y(t), \quad (4.13)$$

described by the transfer functions from $\mathrm{d}v(t)$ to $\mathrm{d}u(t)$ and $\mathrm{d}y(t)$ to $\mathrm{d}u(t)$, i.e. by

$$R_v(z) = p_R^{-1}(z) q_{R_v}(z), \quad (4.14)$$

$$R_y(z) = p_R^{-1}(z) q_{R_y}(z), \quad (4.15)$$

respectively, such that the transfer function of the compensated system coincides with that of the model G

$$G(z) = (1 - F(z)R_y(z))^{-1} F(z) R_v(z) \quad (4.16)$$

as depicted in Figure 4.2.

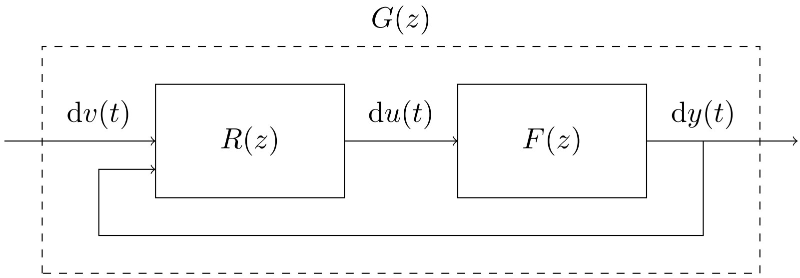

Figure 4.2: Compensated system

Assumption 4.1 $\deg p_G(z) \geq \deg p_F(z)$.

Theorem 4.1 *Given $F(z) \neq 0$ and $G(z)$ satisfying Assumption* 4.1, *the model matching problem by feedback* (4.13) *is always solvable.*

Proof: see Appendix.

Remark 4.1 *Assumption* 4.1 *in the proof of Theorem* 4.1 *is clearly necessary to get a reasonable solution by the left division algorithm of $p_G(z)$ and $p_F(z)$. However, this assumption is not restrictive, since instead of model* (4.12) *with $\deg p_G(z) < \deg p_F(z)$ one can always, without loss of generality, use the transfer function $G'(z) = \left[z^k p_G(z)\right]^{-1} z^k q_G(z)$ being transfer equivalent to $G(z)$, such that $\deg\left(z^k p_G(z)\right) \geq \deg p_F(z)$. Roughly speaking, modulo transfer equivalence there always exists a feedback compensator which solves the model matching problem for given $F(z)$ and $G(z)$.*

If one is looking for a solution within a class of proper compensators, then the situation is similar to that of the case of a feedforward solution.

Proposition 4.4 *$R(z)$ is proper (causal) if and only if*

$$\operatorname{rel}\deg G(z) \geq \operatorname{rel}\deg F(z). \tag{4.17}$$

Proof: see Appendix.

Remark 4.2 *In* [48] *the solution of the MMP via dynamic output feedback was proposed. Comparing to our results the authors of* [48] *consider only the case of proper compensators. Moreover, solution is based on application of the implicit function theorem. As a result, it is constructive up to application of this theorem. Therefore, we may conclude that the approach based on the transfer function formalism is more compact and transparent.*

Example 4.3 Consider the system and the model from Example 4.2, where the feedforward solution did not exist. Note that

$$\begin{aligned} p_F(z) &= z^2 - u(t), & p_G(z) &= z^2, \\ q_F(z) &= z + y(t), & q_G(z) &= 1. \end{aligned}$$

Using the left division algorithm, we get $\gamma(z) = 1$ and $q_{R_y}(z) = -u(t)$ such that $p_G(z) = \gamma(z)p_F(z) - q_{R_y}(z)$. The compensator $p_R(z)\mathrm{d}u(t) = q_{R_v}(z)\mathrm{d}v(t) + q_{R_y(z)}\mathrm{d}y(t)$ is determined by the polynomials

$$\begin{aligned} q_{R_v}(z) &= q_G(z) = 1, \\ q_{R_y}(z) &= -u(t), \\ p_R(z) &= \gamma(z)q_F(z) = z + y(t). \end{aligned}$$

Thus, the one-form, corresponding to the compensator R, is

$$\mathrm{d}u(t+1) + y(t)\mathrm{d}u(t) = \mathrm{d}v(t) - u(t)\mathrm{d}y(t). \tag{4.18}$$

Integrating (4.18) yields $u(t+1) = v(t) - u(t)y(t)$. Finally, the compensator has the following state-space realization

$$\begin{aligned} u(t) &= \eta(t) \\ \eta(t+1) &= v(t) - y(t)\eta(t) \end{aligned}$$

Example 4.4 Consider the system and the model from Example 4.1. In the same manner as in the previous example, we get $\gamma(z) = 1$, $q_{R_y}(z) = -1$, $q_{R_v}(z) = q_G(z) = 1$, $p_R(z) = \gamma(z)q_F(z) = u(t)z + u(t+1)$ and

$$(u(t)z + u(t+1))\mathrm{d}u(t) = \mathrm{d}v(t) - \mathrm{d}y(t),$$

yielding the equation of the compensator $u(t)u(t+1) = v(t) - y(t)$. Finally, note that the compensator has the following state-space realization

$$\begin{aligned} u(t) &= \eta(t) \\ \eta(t+1) &= \frac{v(t) - y(t)}{\eta(t)} \end{aligned}$$

Chapter 5

Region of admissible reference signal values

In this chapter the input-output linearization of a single-input single-output nonlinear discrete-time system by output feedback is recalled from [78]. We investigate the problem of stability of a control signal. The main variable which plays a key role in the behavior of the control function is a reference signal. Clearly, for the different reference signals the control signal behaves in different ways. Thus, it may happen that for some of them it becomes unbounded. Therefore, the main problem under consideration is determination of conditions under which the control signal remains stable. First, we formulate a condition ensuring a bounded-input bounded-output behavior of the controlled object by using a static output feedback. Next, we derive conditions for the algorithm based on a dynamic output feedback.

5.1 Notations and definitions

This section will serve as a brief introduction containing only basic notations which will be used throughout this chapter.

5.1.1 Boundedness

Definition 5.1 *A function $f(x) : \mathbb{R} \to \mathbb{R}$ is called bounded if there exists a real number $M < \infty$ such that $|f(x)| \leq M$ for all $x \in \mathbb{R}$.*

Theorem 5.1 ([93], Theorem 17.23) *A continuous real-valued function on a compact topological space is bounded.*

5.1.2 Functional and numerical series

Definition 5.2 *The series*

$$\sum_{i=1}^{\infty} f_i(x) = f_1(x) + f_2(x) + \cdots + f_l(x) + \cdots , \tag{5.1}$$

the terms of which are functions $f_i(x) : \mathbb{R} \to \mathbb{R}$, is called a functional series.

The sequence of partial sums can be defined as $S_l(x) = \sum_{i=1}^{l} f_i(x)$. Fixing $x = x_0$ in (5.1), one gets the corresponding numerical infinite series

$$\sum_{i=1}^{\infty} f_i(x_0) = f_1(x_0) + f_2(x_0) + \cdots + f_l(x_0) + \cdots , \tag{5.2}$$

which can:

(i) converge to a real number A, i.e. $\lim_{l\to\infty} S_l(x_0) = A$;

(ii) diverge to $\pm\infty$, i.e. $\lim_{l\to\infty} S_l(x_0) = \pm\infty$;

(iii) neither converge nor diverge to $\pm\infty$, i.e. oscillate or diverge by oscillation.

Remark 5.1 *The last item* (*iii*) *states that functional series* (5.1) *diverges by oscillation at the point $x = x_0$, however, the numerical series* (5.2) *is both upper and lower bounded.*

Definition 5.3 *The set of values of the independent variable x for which the series* (5.1) *converges constitutes what is called the region of convergence of that series, denoted by $\mathcal{R}$.*

5.1.3 Stability

In order to get reasonable results, it is necessary to put the following restriction on the reference signal $v(t)$.

Assumption 5.1 *The function $v(t)$ is supposed to be bounded.*

In order to behave properly the i/o system must usually have the following property: bounded inputs must produce bounded outputs, i.e. the system is nonexplosive. This property forms the basis of the following definition of stability used in this chapter, see for instance [28].

Definition 5.4 *A system* (2.1) *is said to be bounded-input bounded-output stable if any admissible bounded input signal $u(t)$ results in a bounded output $y(t)$.*

5.2 Input-output linearization by output feedback

Below we briefly recall the basic facts about output feedback linearization from [78], which allows to describe the class of the systems that can be analyzed.

System is said to be partially linearizable by input-output injections if there is a coordinates transformation $\xi = \Phi(x(t))$, an integer $\bar{n}$, computed according to (2.6), and $\bar{n}$ functions $\phi_1(y(t), u(t)), \ldots, \phi_{\bar{n}}(y(t), u(t))$, such that the system in the new coordinates reads locally as

$$\begin{aligned}
\xi_1(t+1) &= \xi_2(t) + \phi_1(y(t), u(t)) \\
&\vdots \\
\xi_{\bar{n}-1}(t+1) &= \xi_{\bar{n}}(t) + \phi_{\bar{n}-1}(y(t), u(t)) \\
\xi_{\bar{n}}(t+1) &= \phi_{\bar{n}}(y(t), u(t)) \\
\xi_{\bar{n}+1}(t+1) &= \bar{f}_{\bar{n}+1}(\xi(t), u(t)) \\
&\vdots \\
\xi_n(t+1) &= \bar{f}_n(\xi(t), u(t)) \\
y(t) &= \xi_1(t)
\end{aligned}$$

If $\bar{n}$ happens to be n, i.e. equals to the number of the state equations, then the system is said to be fully linearizable by input-output injections.

Define $\mathcal{V}^0 = 0$, $\mathcal{V}^l = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t), \ldots, \mathrm{d}y(t+l-1), \mathrm{d}u(t), \ldots, \mathrm{d}u(t+l-1)\}$, for $l \geq 1$.

Lemma 5.1 ([78]) $\mathrm{d}y(t+\bar{n}) \in \mathcal{V}$ *is linearizable by* $\bar{n}$ *input-output injections* $\phi_1, \ldots, \phi_{\bar{n}}$*, when*

$$\begin{aligned}
\mathrm{d}y(t+\bar{n}) = \mathrm{d}\sigma^{\bar{n}-1}\phi_1(y(t), u(t)) + \\
+ \mathrm{d}\sigma^{\bar{n}-2}\phi_2(y(t), u(t)) + \cdots + \mathrm{d}\phi_{\bar{n}}(y(t), u(t)).
\end{aligned}$$

5.2.1 Static output feedback

System (2.2) is said to be i/o linearizable by static output feedback if there is a regular static output feedback

$$u(t) = H(y(t), v(t)) \tag{5.3}$$

such that the closed-loop system

$$\begin{aligned}
x(t+1) &= f(x(t), H(h(x(t)), v(t))) \\
y(t) &= h(x(t))
\end{aligned}$$

is diffeomorphic to

$$\begin{aligned}\xi^1(t+1) &= A\xi^1(t) + bv(t)\\ \xi^2(t+1) &= \bar{f}^2(\xi(t), v(t))\\ y(t) &= c\xi^1(t)\end{aligned}$$

where $\xi^1 \in \mathbb{R}^{\bar{n}}$, $\xi^2 \in \mathbb{R}^{n-\bar{n}}$, (c, A) is an observable pair.

Theorem 5.2 ([78]) *Let r be the finite relative degree ($r < \infty$) of system (2.2). The system is input-output linearizable by static output feedback if and only if*

(i) $\mathrm{d}y(t+\bar{n})$ *is linearizable by* $\bar{n}$ *output injections* $\phi_1(y(t), u(t)), \ldots, \phi_{\bar{n}}(y(t), u(t))$;

(ii) $\dim_{\mathbb{R}}(\mathrm{span}_{\mathbb{R}}\{\mathrm{d}y(t), \mathrm{d}\phi_1(y(t), u(t)), \ldots, \mathrm{d}\phi_{\bar{n}}(y(t), u(t))\}) = 2.$

5.2.2 Dynamic output feedback

System (2.2) is said to be i/o linearizable by dynamic output feedback if there exists a dynamic output feedback

$$\begin{aligned}u(t) &= H(y(t), \eta(t), v(t))\\ \eta(t+1) &= F(y(t), \eta(t), v(t))\end{aligned} \tag{5.4}$$

such that the closed-loop system

$$\begin{aligned}x(t+1) &= f(x(t), H(h(x(t)), \eta(t), v(t)))\\ \eta(t+1) &= F(y(t), \eta(t), v(t))\\ y(t) &= h(x(t))\end{aligned}$$

is diffeomorphic to

$$\begin{aligned}\xi^1(t+1) &= A\xi^1(t) + bv(t)\\ \xi^2(t+1) &= \bar{f}^2(\xi(t), \eta(t), v(t))\\ y(t) &= c\xi^1(t)\end{aligned}$$

where $\eta(t) \in \mathbb{R}^q$, $\xi^1 \in \mathbb{R}^{\bar{n}}$, $\xi^2 \in \mathbb{R}^{n+q-\bar{n}}$, (c, A) is an observable pair.

Theorem 5.3 ([78]) *Let r be the finite relative degree ($r < \infty$) of system (2.2). The system is input-output linearizable by dynamic output feedback if and only if*

$$\begin{aligned}\mathrm{d}y(t+\bar{n}) = \lambda_1 \mathrm{d}y(t+\bar{n}-1) + \cdots + \lambda_{r-1}\mathrm{d}y(t+\bar{n}-r+1)+\\ + \mathrm{d}\phi_{\bar{n}}(\cdot, y(t), u(t)) \circ \sigma\phi_{\bar{n}-1}(\cdot, y(t), u(t)), \circ \cdots\\ \cdots \circ \sigma\phi_{r+1}(\cdot, y(t), u(t)) \circ \phi_r(y(t), u(t)).\end{aligned}$$

The theory presented in this section allows to determine whether the controlled system is linearizable by static or dynamic output feedback or not. The algorithm for constructing an appropriate controller can be found in [78].

5.3 Region of admissible values

Consider a linear closed-loop system consisting of nonlinear controlled system defined by equation (2.1) or (2.2) and a regulator based on the input-output feedback linearization algorithm. Then, the structure of the corresponding control system is represented schematically in Figure 5.1.

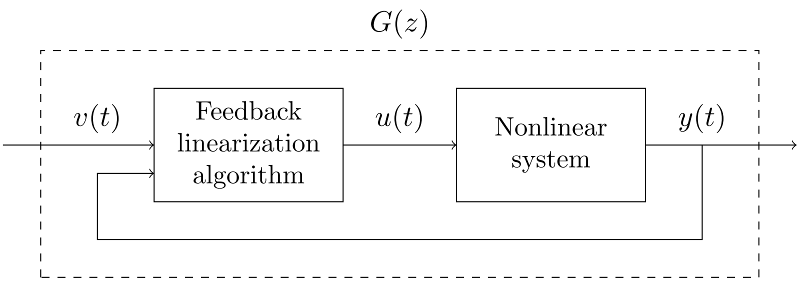

Figure 5.1: Control system

Let the reference model, defining the relation between the output of the system and the reference signal of the control system presented in Figure 5.1, be described by the following equation

$$y(t+n) + a_{n-1}y(t+n-1) + \cdots + a_0 y(t) = \\ = b_{n-r}v(t+n-r) + \cdots + b_0 v(t), \quad (5.5)$$

where $a_i, b_j \in \mathbb{R}$ for $i = 0, \ldots, n-1$, $j = 0, \ldots, n-r$ are parameters of the reference model, or by the transfer function

$$G(z) = \frac{b_{n-r}z^{n-r} + b_{n-r-1}z^{n-r-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1}z^{n-1} + a_{n-2}z^{n-2} + \cdots + a_1 z + a_0},$$

where $G(z)$ is a linear discrete-time reference model defining dynamics of the closed-loop control system, with characteristic polynomial being Schur stable, i.e. all its roots are placed inside the unit circle.

It should be mentioned that the main idea of using a control strategy based on the output feedback linearization algorithm consists in modifying the system structure by suitable feedbacks, replacing nonlinear relations between $y(t)$ and $u(t)$ with linear ones, so that the closed-loop control system can be described by the linear transfer function $G(z)$ or equivalently by (5.5). It means that after transformation one can work with the control system as with linear one. The fundamental requirement is that this approach would lead to a bounded-input bounded-output stable closed-loop system. Thus, choosing a linear reference model (5.5) with all its poles strictly inside the unit circle and constructing an appropriate regulator, one can guarantee that the regulator generates a control signal resulting in the bounded output. However, nothing definite can be said about $u(t)$. In other words, observing only the output, we cannot say for sure whether $u(t)$ is bounded or not. It is not difficult to model a situation when the use of any specific function $v(t)$ or even $v(t) = \text{const}$ yields an unbounded control signal, whereas, the output of the system remains bounded.

Example 5.1 Consider the system $y(t+1) = u(t)y(t)^2$, which can be linearized by the feedback $u(t) = \frac{v(t)}{y(t)^2}$. The application of $u(t)$ to the system results in the stable closed-loop system described by the relation $y(t+1) = v(t)$, since the only pole is in the origin. Let us define the reference signal as $v(t) = e^{-t^2}$. The latter means that

$$u(t) = \frac{v(t)}{y(t)^2} = \frac{v(t)}{v(t-1)^2} = \frac{e^{-t^2}}{e^{-2(t-1)^2}} = e^{t^2-4t+2},$$

which exponentially grows as t tends to infinity.

Definition 5.5 *The set of values of the reference signal $v(t)$ for which the control signal $u(t)$, produced by regulator* (5.3) *or* (5.4)*, remains bounded is called the region of admissible values and denoted by Ω.*

Our next task is to specify Ω. Due to the fact that there are two different types of regulators, the solution of the above problem is investigated first in the simpler static output regulator case and then in the dynamic case.

5.3.1 The case of static output feedback

Consider the control system presented in Figure 5.1, in which the regulator is defined by equation (5.3). The behavior of the control signal $u(t)$ can be determined by the following proposition.

Proposition 5.1 *If the function $H(y(t), v(t))$ in* (5.3) *is continuous on the whole real line $\mathbb{R}$, then the control signal $u(t)$ is always bounded.*

Proof: see Appendix.

Corollary 5.1 *The region of admissible values is a whole real line, i.e. $\Omega = \mathbb{R}$.*

Though Proposition 5.1 states a nice result, it should be mentioned that the condition on continuity of the function $H(y(k), v(k))$ is too restrictive. As a result, such regulator, which is a special case of (5.3) under this assumption, can be applied to a very small class of nonlinear models.

Example 5.2 Consider the following neutron kinetics system, from [4], described via the state-space equations as

$$\begin{aligned} x_1(t+1) &= x_2(t) + b_1 u(t) x_1(t) \\ x_2(t+1) &= a_2 x_1(t) + b_2 u(t) + a_1 b_1 u(t) x_1(t) \\ y(t) &= x_1(t) \end{aligned} \tag{5.6}$$

where $x_1(t)$ denote the population of neutrons, $x_2(t)$ denote the average population of precursor groups, $u(t)$ is the reactivity and is a control variable. The relative variables and the parameters are: $a_1 = 1.24$, $a_2 = -0.24$, $b_1 = 0.9$, $b_2 = 0.891$.

Now, one can check that both conditions of Theorem 5.2 are fulfilled. Thus, controlled system defined by (5.6) can be i/o linearized. In [78] the output feedback $u(t) = v(t)/y(t)$ was proposed. However, easy calculations show that the closed-loop system has the following pair of poles $z_1 = 0.24$ and $z_2 = 1$. From the control point of view, this choice is undesirable, since the second pole causes the non-asymptotic behavior of the control system. Therefore, we propose the alternative output feedback $u(t) = (v(t) - K)/y(t)$ with $K \in (0, 760/891)$ which solves the problem with stability of the output signal. Obviously, under Assumption 5.1, conditions (5.5) and $v(t) \not\equiv 0$ the function describing the control signal will always be bounded resulting to the stable behavior of the control system. Moreover, the region of admissible values in this case is $\Omega = \mathbb{R} \setminus \{v(t) \equiv 0\}$.

Note that the controlled system is fully linearized, i.e. $\bar{n} = n = 2$, and the closed-loop system reads as

$$\begin{aligned} \xi(t+1) &= A\xi(t) + bv(t) \\ y(t) &= c\xi(t) \end{aligned}$$

with

$$A = \begin{pmatrix} 0 & 1 \\ -0.24 - 0.891K & 1.24 - 0.9K \end{pmatrix},$$

$$b = \begin{pmatrix} 0.9 \\ 2.007 - 0.81K \end{pmatrix}, \quad c = \begin{pmatrix} 1 & 0 \end{pmatrix},$$

$\xi_1(t) = x_1(t), \xi_2(t) = x_2(t)$, $\xi(t) \in \mathbb{R}^2$ in the state-space representation, or as

$$\begin{aligned} y(t+2) + (0.9K - 1.24)y(t+1) + (0.891K + 0.24)y(t) &= \\ &= 0.9v(t+1) + 0.891v(t) \end{aligned}$$

in the input-output form, respectively.

5.3.2 The case of dynamic output feedback

In practice, the conditions of Theorem 5.2 can seldom be fulfilled, and regulator (5.4) can be applied to a very restricted class of nonlinear models. Therefore, we specify Ω in the context of dynamic output feedback. Thus, the main purpose of the further part is establishing the conditions allowing to confirm that not only the output of the system, but also the control signal remain bounded.

Next, we present the algorithm, with explanatory comments, for determining the region of admissible values, which helps us to formulate the criterion for the control signal to be both upper and lower bounded during the control period of time.

Algorithm:

Step 1. Apply the backward-shift operator to the states $\eta(t+1)$ of regulator (5.4) the sufficient number of times and substitute them into the function of control signal as

$$\begin{aligned} u(t) = \tilde{H}(&y(t), y(t-1), \ldots, y(t-\bar{n}+1), \\ &v(t), v(t-1), \ldots, v(t-\bar{n}+1), \\ &u(t-1), u(t-2), \ldots, u(t-\bar{n}+1)). \end{aligned} \tag{5.7}$$

Step 2. Replace all time instances of the output in (5.7) by the corresponding time instances of the reference signal under condition (5.5) yielding that $u(t) = \tilde{\tilde{H}}(\cdot)$.

Step 3. Assuming that the initial conditions $u(0) := u_0, u(-1) := u_1, \ldots, u(-\bar{n}+1) := u_{\bar{n}-1}$, solve the difference equation

$$u(t) = \tilde{\tilde{H}}(\cdot). \tag{5.8}$$

Now, two cases are possible:

(i) if it is not possible to solve (5.8), then stop[1];

(ii) if the solution of (5.8) exists

$$u(t) = \sum_{i=1}^{t} \varphi_i(\cdot), \tag{5.9}$$

then go to **Step 4**.

[1] To be more precise, it is a very common situation, when the nonlinear difference equation is not solvable analytically.

Since we are looking for the control signal to be bounded at infinity, i.e. when $t \to \infty$, for the sake of convenience, the upper bound of (5.9) can be replaced by infinity. The latter means that (5.9) can be rewritten in the form of the infinite functional series

$$u(t) = \sum_{i=1}^{\infty} \varphi_i(\cdot). \tag{5.10}$$

Step 4. Since we are looking for $u(t)$ to be both upper and lower bounded, under Assumption 5.1, it is enough to consider the case when the reference signal is a constant function, i.e. $v(t) = \text{const} =: v$. Assuming that the initial conditions are $v(0) := v_0, v(-1) := v_1, \ldots, v(-2\bar{n} + 1) := v_{2\bar{n}-1}$, equation (5.10) can be rewritten as follows

$$u(t) = \sum_{i=1}^{\infty} \tilde{\varphi}_i(v). \tag{5.11}$$

Step 5. Determine the region of convergence $\mathcal{R}$ of (5.11) in the sense of Definition 5.3.

Notice that $\mathcal{R}$ gives us only the set of points where (5.11) converges.

Step 6. Check all the boundary points where (5.11) diverges with respect to Remark 5.1 and denote the set of all points where the corresponding numerical series appears to be bounded by $\mathcal{D}$.

Finally, the region of admissible values can be calculated by the formula

$$\Omega = \mathcal{R} \cup \mathcal{D}. \tag{5.12}$$

Step 7. End of the algorithm.

Now, using Definition 5.5 and the algorithm presented above, we can formulate the following theorem.

Theorem 5.4 *The controlled system* (2.1) *is bounded-input bounded-output stable if and only if* $v(t) \in \Omega$.

Proof: see Appendix.
Example 5.3 Consider the i/o equation

$$y(t+2) = y(t+1) + a_1 u(t+1) + a_2 u(t) + y(t+1)u(t+1) + y(t)u(t),$$

where $a_1, a_2 \in \mathbb{R}$.

First, note that the relative degree of this system is equal to one. After that, one can check that condition (ii) of Theorem 5.2 is not fulfilled;

however, the condition of Theorem 5.3 is satisfied. Thus, using the approach proposed in [78], dynamic regulator (5.4) can be represented by the following equations

$$u(t) = \frac{\eta_1(t) - y(t)}{a_1 + y(t)}, \tag{5.13}$$

$$\eta_1(t+1) = v(t) - a_2 u(t) - u(t)y(t) \tag{5.14}$$

and the closed-loop system reads as

$$y(t+2) = v(t). \tag{5.15}$$

Now, in order to calculate the region of admissible values, we use the algorithm introduced above. First, apply the backward-shift operator to equation (5.14) once and substitute the obtained expression into the function of control signal (5.13) as follows

$$u(t) = \frac{v(t-1) - (a_2 + y(t-1))u(t-1) - y(t)}{a_1 + y(t)}. \tag{5.16}$$

Next, using relation (5.15), we can replace all time instances of the output by the corresponding time instances of the reference signal and rewrite equation (5.16) in the following form

$$u(t) = \frac{v(t-1) - (a_2 + v(t-3))u(t-1) - v(t-2)}{a_1 + v(t-2)}. \tag{5.17}$$

After that, assuming that $u(-1) = 0$, the solution of (5.17) can be represented in the closed form by

$$u(t) = \left[\prod_{i=1}^{t-1} \frac{-a_2 - v(i-2)}{a_1 + v(i-1)}\right] \cdot \sum_{j=0}^{t-1} \frac{v(j) - v(j-1)}{(a_1 + v(j-1)) \prod_{i=1}^{j} \frac{-a_2 - v(i-2)}{a_1 + v(i-1)}}. \tag{5.18}$$

Let the reference signal be defined as a step function

$$v(t) = \begin{cases} v, & t > 0, \\ 0, & t \leq 0. \end{cases} \tag{5.19}$$

Now, according to Step 4 of the algorithm, after the simplification of (5.18) with respect to conditions (5.19), we obtain

$$u(t) = \sum_{i=0}^{\infty} (-1)^{i+1} \frac{a_2 v (v + a_2)^i}{a_1 (v + a_1)^{i+1}} = -\frac{a_2 v}{a_1 (v + a_1)} \sum_{i=0}^{\infty} \left(-\frac{v + a_2}{v + a_1}\right)^i \tag{5.20}$$

for $t > 2$.

Notice that the first three partial sums for time instances $t = 0, 1, 2$ in (5.18) cannot be calculated from (5.20) and their values are $S_0(v) = 0, S_1(v) = 0$ and $S_2(v) = v/a_1$, respectively.

Obviously, (5.20) is a special case of geometric series $\sum_{i=0}^{\infty} cq^i$ with $c = -\frac{a_2 v}{a_1(v+a_1)}$ and $q = -\frac{v+a_2}{v+a_1}$. It means that (5.20) converges if and only if $|q| < 1$. Solving this inequality, we obtain the following three cases

(i) if $a_1 > a_2$, then $v > -\frac{a_1+a_2}{2}$;

(ii) if $a_1 < a_2$, then $v < -\frac{a_1+a_2}{2}$;

(iii) if $a_1 = a_2$, then $u(t) = \sum_{i=0}^{\infty} c(-1)^i$ does not converge in the sense of Remark 5.1; however, it is both upper and lower bounded, hence $v(t) \in \mathbb{R}$.

Now, according to Step 6, we have to check the boundary points for the first two cases (i) and (ii). After substituting $d_0 = -\frac{a_1+a_2}{2}$ into (5.20), we get $u(t) = -\frac{a_2 v}{a_1(v+a_1)}$ concluding that for d_0 the corresponding numerical series converges.

Finally, summarizing the obtained information, we may conclude that

- if $a_1 > a_2$, then $\mathcal{R}_1 = \left(-\frac{a_1+a_2}{2}, \infty\right)$ which together with $\mathcal{D} = \{d_0\}$ by (5.12) give us the following region of admissible values $\Omega_1 = \left[-\frac{a_1+a_2}{2}, \infty\right)$;
- if $a_1 < a_2$, then $\mathcal{R}_2 = \left(-\infty, -\frac{a_1+a_2}{2}\right)$ which together with $\mathcal{D} = \{d_0\}$ by (5.12) give us the following region of admissible values $\Omega_2 = \left(-\infty, -\frac{a_1+a_2}{2}\right]$;
- if $a_1 = a_2$, then $\Omega_3 = \mathbb{R}$.

It means that if we choose an arbitrary function $v(t)$ whose values are inside the region Ω_i, then the control signal $u(t)$ for the corresponding case, produced by the regulator (5.4), will remain bounded during the control period of time and the object will be bounded-input bounded-output stable.

Example 5.4 The second-order discrete-time model is described by the i/o equation [94]

$$y(t+2) = 1.2y(t+1) - 0.8y(t) + u(t+1) + \\ + 0.6u(t) + 0.2y(t+1)u(t+1).$$

One can check that the static compensator is not applicable. However, the following dynamic regulator (5.4) can be used

$$u(t) = \frac{\eta_1(t) - 1.2y(t)}{1 + 0.2y(t)},$$
$$\eta_1(t+1) = v(t) - 0.6u(t) + 0.8y(t).$$

Note that the calculation procedure of Ω is similar to the one illustrated in the previous example and is therefore omitted. One can check that $\Omega = (-\infty, -8] \bigcup [-2, +\infty)$.

Remark 5.2 *If the system is not fully linearizable by output feedback, then we have to make an additional assumption about stable zero dynamics. In fact, this assumption is not restrictive, but necessary in order to guarantee stability of non-linearized states.*

Chapter 6

Symbolic polynomial tools as subpackage of NLControl

In this chapter the theory of Ore polynomials, presented in Chapter 2, is encapsulated in the form of *Mathematica* functions through the NLControl package. We have developed a set of functions for solving the modelling problems of nonlinear control systems, based on the theory of Ore polynomial rings. The implemented functions can be divided into several groups according to their functionality and tasks to be solved. The first part of the software includes the assistant functions that do not solve any control problems directly. For example there is a number of functions that implement the basic operations with Ore polynomials, since there is neither built-in functions nor supplement package available for *Mathematica*, addressing these operations. These basic functions include addition and multiplication, the left (right) quotient and reminder, the greatest common left (right) divisor and the least common left (right) multiple. The second part contains the programs for solving different problems arising in the control theory by means of polynomial formalism. These are, for example reduction, realization and model matching. Note that the problems listed above are only those studied in this thesis; however, the functionality of NLControl is not limited only by corresponding functions, and the package contains more different options and possibilities.

At the beginning of the chapter a brief overview of the package is presented. After that the basic operations that can be performed with Ore polynomials are explained. Next, the transformations between different system descriptions are shown, followed by the model matching problem. In the next section a number of illustrative examples is presented. Finally, small notes about the realization of the package by means of *webMathematica* service conclude the chapter.

6.1 Overview of the NLControl package

This section provides necessary information on the NLControl package and presents a number of important functions.

Mathematica uses certain data structures or so-called control objects to represent the control system. These objects contain all necessary information about the system. The most typical examples within the framework of the package are `StateSpace`, `IO` and `TransferFunction`. In order to provide the possibility to use the functions from the package, one should load it, what can be done by the following command

```
In[1]:= <<NLControl`Master`
```

To perform computations with the system described by the state equations, it should be entered in the form determined by *Mathematica* and NLControl package as shown below, which can be obtained using the following function

`StateSpace[f, Xt, Ut, t, h, Yt, Type]`,

where *f* is a list of the state functions, *Xt*, *Ut* and *Yt* define lists of the state, input and output variables, respectively; *t* is a time argument and *h* defines the output function. The argument *Type* may have one of the following values: *TimeDerivative* stands for continuous-time case and *Shift* for discrete-time case.

To enter the i/o system, one has to use the syntax

`IO[eqs, Ut, Yt, t, Type]`,

where the meanings of the arguments *Ut*, *Yt*, *t* and *Type* are the same as in the case of `StateSpace`, and *eqs* defines the i/o equation.

The third object is

`TransferFunction[z, F, Ut, Yt, t, Type]`,

which represents a system given by its transfer matrix *F* with *z* being polynomial indeterminate. Remaining arguments are the same as for i/o equations.

Note that the keywords `StateSpace`, `IO` and `TransferFunction` act also as functions transforming system between different representations. For example, if one desires to get the transfer function of the i/o equation, then the following shortened code may be called: `TransferFunction[z, IO]` with `IO` being previously defined as a *Mathematica* object. Thus, one may see, and it will be shown further more precisely, that there exist several alternative ways to solve the same problem.

Remark 6.1 *The tools of NLControl are not designed for approximate calculations. Therefore, all real (floating-point) numbers are transformed into rational numbers by* `StateSpace` *and* `IO`*.*

Sometimes the form of the objects, determined by *Mathematica* and NLControl package, differs from the traditional and familiar form. Thus, to make the output pleasant to read the function `BookForm` was introduced. One of the optional arguments of this function is *`TimeArgument`*. Its value can be *`True`*, *`False`* or *`Subscripted`*. If the value is

- *`True`*, then the time argument *`t`* will be printed for each variable in brackets `[]`;
- *`False`*, then the time argument will be left out to make the output result visually more compact;
- *`Subscripted`* means that *`t`* will be printed as a subscript.

The default value of *`TimeArgument`* is *`True`*.

6.2 Ore polynomials: standard operations

Recall that polynomials, describing nonlinear system, belong to the Ore polynomial ring. Moreover, the polynomial coefficients belong to the differential/difference field depending on the type of the system under consideration. Thus, NLControl uses a special object `OreRing` to store and handle the corresponding information. It encapsulates all necessary data about relations between variables. In order to create the object `OreRing`, the following function has to be used

`DefineOreRing[`*`z`*`, `*`sys`*`],`

where *`z`* is a polynomial variable and *`sys`* is a control system. Note that it is also possible to work with Ore rings not associated with any control system. In this case, the object `OreRing` can be created by

`DefineOreRing[`*`z`*`, `*`t`*`, `*`Shift`*`],`

if the polynomial coefficients are from a difference field, and by

`DefineOreRing[`*`z`*`, `*`t`*`, `*`TimeDerivative`*`],`

if the polynomial coefficients are from a differential field.

Additionally, we define the objects representing the Ore polynomial and the fraction of Ore polynomials. The reason for creating a special object can be seen from the fact that by default *Mathematica* changes the order of the factors related by the standard multiplication operator "$\star$". For example, the expression `y[t]*z` would be converted to `zy[t]`. However, according to (2.8) or (3.45), such a permutation is wrong for Ore polynomials. Therefore, the Ore polynomial of the form (2.7) can be represented as

`OreP[`a_0`, `a_1`, ..., `a_n`],`

where $a_0, \ldots, a_n$ are polynomial coefficients. The fraction of two polynomials $p^{-1}(z)q(z)$ has to be entered as

`OreR[`p`, `q`].`

Another useful function

`OreSimplify[`p`, `R`]`

simplifies the polynomial p, assuming it belongs to the Ore ring R. The argument R has to be given as the object `OreRing`. If there are relations defined between polynomial coefficients, this yields that certain expressions in the field $\mathcal{K}$ are equal to zero. The function `OreSimplify` applies these relations to polynomial coefficients and then simplifies the result. Note that these relations are not applied automatically, since the polynomial object `OreP` has no information about them.

Addition of polynomials may be performed by *Mathematica* standard summation operator "+", but addition of polynomial fractions requires a special function

`OrePlus[OreR[`p_1`, `q_1`], OreR[`p_2`, `q_2`], `R`],`

where p_1, q_1, p_2, q_2 are polynomials from the Ore ring R. The function

`OreMultiply[`p_1`, `q_1`, `R`]`

computes a product of polynomials p_1, q_1 from the Ore ring R and is based on commutation rule (2.8) or (3.45), depending on the time domain. The product of left fractions may be found by the same function.

Suppose that p and q are polynomials from the Ore ring R. The following functions may be applied to them:

- `LeftQuotientRemander[`p`, `q`, `R`]` returns a list $\{\gamma, \rho\}$, where γ is the left quotient and ρ is the left remainder;
- `LeftQuotient[`p`, `q`, `R`]` finds the left quotient;
- `LeftRemainder[`p`, `q`, `R`]` finds the left remainder;
- `LeftGCD[`p`, `q`, `R`]` finds the greatest common left divisor;
- `LeftLCM[`p`, `q`, `R`]` finds the least common left multiple.

Corresponding *right* functions are also available. For that purposes one needs to replace `Left` by `Right` in the name of the function. Finally, there are three functions for matrices with entries belonging to the Ore ring R:

- `OreDot[`A_1`, ..., `A_n`, `R`]` finds the product of polynomial matrices $A_1, \ldots, A_n$ analogously with the standard matrix multiplication function `Dot`;
- `LowerLeftTriangularMatrix[`A`, `R`]` transforms the polynomial matrix A into the lower left triangular form;
- `OreInverse[`A`, `R`]` computes the inverse of the polynomial matrix A.

6.3 Polynomial system description

Sometimes it is necessary to have possibility for working with polynomials describing the system. Moreover, some of the programs implemented in the NLControl package requires that the system was put into the polynomial form. It can be performed by the following function

`FromIOToOreP[`*sys*`]`,

where the argument *sys* represents a system defined by the i/o equation. In general, this function computes and returns the $p \times p$ and $p \times m$-dimensional matrices $P(z)$ and $Q(z)$, respectively, with entries from the Ore polynomial ring. Note that in case of SISO systems the function returns matrices consisting of only single element each. By default all the functions, based on polynomial formalism, use `FromIOToOreP` to transform the original system, represented by the i/o equations into the polynomial form.

6.4 Reduction and Realization problems

Let us start from the system described by the i/o equation or the set of i/o equations. In Chapter 3 the basic idea of the realization procedure was explained as well as the corresponding algorithms and formulas for the certain types of systems were derived. Note that presented algorithms are constructive, and therefore, allow to derive the classical state-space form directly from the polynomial system description. It means that they can be formalized and easily implemented within *Mathematica* or any other software. However, we would like to emphasize that the obtained polynomial formulas and algorithms are designed for the realizable and irreducible systems. It was mentioned in Section 2.2 that an nth-order realization of the i/o equation under consideration is accessible if and only if the system is irreducible. Thus, the following function has been implemented in NLControl

`Irreducibility[`*sys*`],`

where the argument *sys* represents a system. This function allows to check whether the original system is reducible or not. Obviously, if the returned answer is *False*, then one may be sure that the state equations will be accessible whenever they exist. However, if the answer is *True*, then one may use another function

`Reduction[`*sys*`],`

where the meaning of the argument *sys* is the same as in case of the function `Irreducibility`. `Reduction` returns a new lower order i/o system, which is transfer equivalent to the original one. It means that the transfer functions of these systems are equivalent. In fact, one is not obliged to use the function `Irreducibility` and in principle may directly use the function `Reduction`, which in case of irreducible equations returns the original equations.

Once we have obtained new equations or made sure that the original system is irreducible, we may proceed further. It is known that the classical state-space realization does not necessarily exist for every nonlinear i/o model. Therefore, to check whether the system is realizable or not, one may use the following function

`Realizability[`*sys*`],`

where *sys* again represents a system. If the returned answer is *False*, then the minimal state-space representation does not exist. However, if the answer is *True*, then one may use the following function

`Realization[`*sys*`,` $x_{\#}$`[`*t*`]&],`

where the second argument $x_{\#}$`[`*t*`]&` stands for the so-called pure function, which determines the state variables to be denoted as x_1`[`*t*`]`, x_2`[`*t*`]`, ..., x_n`[`*t*`]`, where n is an order of the system. The alternative possibility is to replace the pure function by the predefined list of state variables `{`x_1`[`*t*`], ...,` x_n`[`*t*`]}`, but in this case one has to be careful and choose a list of the correct length. `Realization` returns a set of state equations. By analogy with `Reduction` in principle one may immediately use the function `Realization`, which in case of non-realizable equations returns the empty set.

The general scheme of deriving the state equations from the i/o equations by means of `Realization` function is shown in Figure 6.1.

Note that the block Assumptions in Figure 6.1 requires a number of different system specific assumptions, defined in the previous chapters, to be hold. It means that as soon as the user defined the i/o equation, programm starts to check whether the entered data is correct or not.

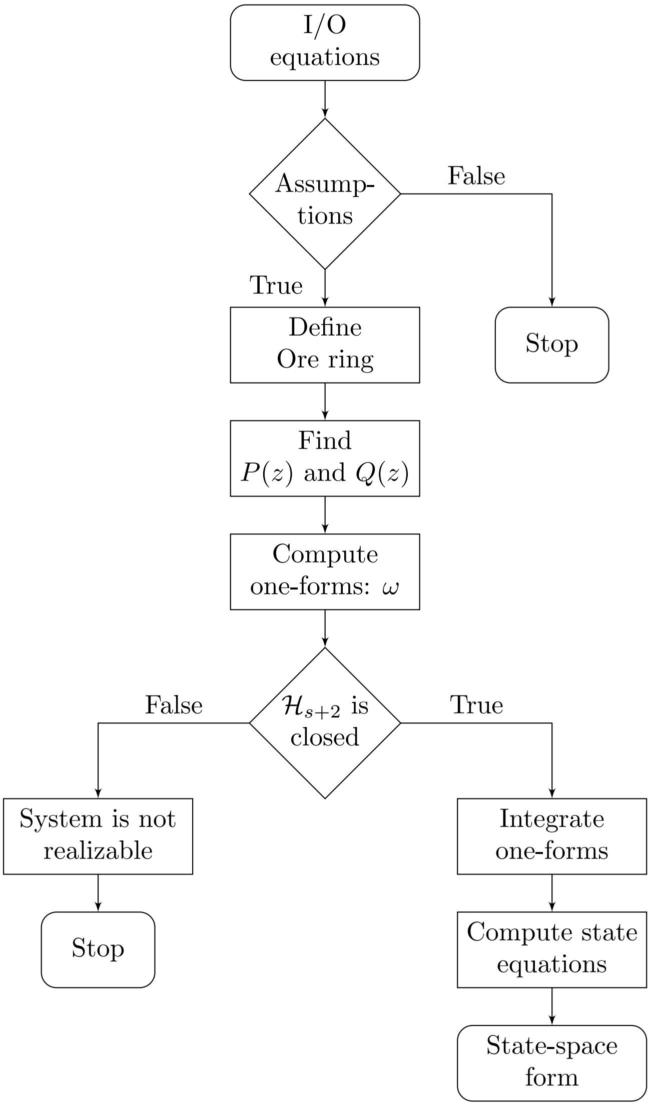

Figure 6.1: The block-diagram of the function `Realization`

6.5 Model matching problem

At the beginning of the chapter the basic idea of representing a system in the form of the `TransferFunction` object was explained. It can be used in several ways for solving different problems. Here, we will show how the transfer functions can be applied to the model matching problem. It is known that there are two typical feedforward and feedback compensators, considered for open- and closed-loop systems, respectively. For the both cases the general statement of the problem can be formulated as follows. Given a model to be controlled and a reference model described by their transfer functions. Find a compensator described by its transfer function such that the transfer function of the compensated system coincides with that of the reference model.

In case of the feedforward compensator the following function can be used

`FeedforwardCompensator[`*`eqs1`*`, `*`eqs2`*`],`

where *eqs1* and *eqs2* are the i/o equations of the system and reference model, respectively. However, this function was designed in such a way that alternative input arguments can be used, namely

`FeedforwardCompensator[`*`F`*`, `*`G`*`],`

where *F* and *G* stand for the transfer functions of the system and reference model, respectively. In the first case the function produces the output in the form of `IO` object, i.e. the function returns the i/o equation of the compensator. Note that, according to the theory, the feedforward solution does not always exist. It means that the one-form representing the compensator may not be integrable. Therefore, it may happen that the function returns the empty set. As for the second case the output is always in the form of `TransferFunction` object, whenever the problem is solvable.

In case of the feedback compensator the similar function is available. Its name is `FeedbackCompensator`. The meaning of the arguments is the same as for the feedforward case. It should be mentioned that, according to the theory, the solution and as a result the corresponding compensator are always exist, meaning that the output of the function cannot be the empty set.

Remark 6.2 *Some of the functions, presented above as well as available in NLControl, are designed for MIMO systems. For the more detailed information we refer the reader to* [62]*,* [88] *and the web page* [74] *discussed below.*

It should be mentioned that if the called function is not able to produce the output, or the calculation process was interrupted, then *Mathematica* generates an error message. Thus, in order to keep specifications according to *Mathematica* requirements, we created a number of warning and error messages, which can be easily distinguished by a particular color, name, or semantic load.

6.6 Examples

In the following example the functions discussed above will be illustrated with a brief explanation of the basic theoretical and technical moments.
Example 6.1 Consider the discrete-time model of the controlled van der Pol oscillator derived in [2]

$$y(t+2) = \theta_1 y(t+1) - \theta_2 y(t) + \theta_3 y(t)^2 y(t+1) + \theta_4 y(t)^3 + \theta_5 u(t), \quad (6.1)$$

where $\theta_i \in \mathbb{R}$ for $i = 1, \ldots, 5$. In order to use the functions described above, we first have to load the NLControl package.

```
In[1]:= <<NLControl`Master`
```

Next, we enter the system as follows.

```
In[2]:= eqs = {y[t+2] == θ1y[t+1] - θ2y[t] + θ3y[t]^2y[t+1] +
        θ4y[t]^3 + θ5u[t]};
        Ut = {u[t]};
        Yt = {y[t]};
        sysIO = IO[eqs, Ut, Yt, t, Shift];
        BookForm[sysIO]

Out[6]= y[t+2] = θ5u[t] - θ2y[t] + θ4y[t]^3 + (θ1 + θ3y[t]^2)y[t+1]
```

Let us start the analysis by examining whether model (6.1) is reducible or not. It can be performed by running the following command

```
In[7]:= Reduction[sysIO]
        Reduction::irred: The system already has irreducible form.
Out[7]= IO[{y[t+2] == θ5u[t] - θ2y[t] + θ4y[t]^3 + θ1y[t+1] +
          θ3y[t]^2y[t+1]}, {u[t]}, {y[t]}, t, Shift]
```

From the output of the function and generated warning message we can see that the system has irreducible form. This means that there is no any other model of the lower order. Since the system is irreducible, we may continue our analysis and proceed to the next function, which allows to find the state equations, whenever possible.

```
In[8]:= {cls,repl} = Realization[sysIO, {x1[t], x2[t]}];
        BookForm[cls]
```

Out[9]= $x_1[t+1] = x_2[t]$
$x_2[t+1] = \theta_5 u[t] - \theta_2 x_1[t] + \theta_4 x_1[t]^3 + \left(\theta_1 + \theta_3 x_1[t]^2\right) x_2[t]$
$y[t] = x_1[t]$

Thus, the system is realizable and it was possible to construct the classical state equations. Now, we proceed to the functions which allow to calculate the feedforward and feedback compensators. Suppose that the reference model is

```
In[10]:= rmIO = IO[{y[t+2] == v[t]}, {v[t]}, {y[t]}, t, Shift];
         BookForm[rmIO]
```

Out[11]= $y[t+2] = v[t]$

The following commands have to be used for the case of feedforward compensator

```
In[12]:= FeedforwardCompensator[sysIO, rmIO]
```

FeedforwardCompensator::unable: The function is unable to construct feedforward compensator.

IntegrateOneForms::nonint: The set of differential one-forms is not completely integrable.

Out[12]= { }

and for the case of feedback compensator

```
In[13]:= fbc = FeedbackCompensator[sysIO, rmIO];
         BookForm[fbc]
```

Out[14]= $\theta_5 u[t] - v[t] - \theta_2 y[t] + \theta_4 y[t]^3 + \left(\theta_1 + \theta_3 y[t]^2\right) y[t+1] = 0$

It has been already mentioned that the class of systems for which the feedforward compensator can be found is quite restricted. That is the case for system (6.1). It means that there is no any compensator which solves the feedforward model matching problem. On the other hand, the feedback MMP is always solvable and the corresponding i/o equation was found. However, the output is given by the implicit function. Thus, before implementing the control, one has to solve the corresponding equation with respect to the control signal, what can be done by the functions available in the standard *Mathematica* version.

Next, we present several separate examples, which illustrate the essence of each function. Suppose that for each of the following examples the NLControl package is already loaded.

Example 6.2 Recall the system from Example 4.1

$$y(t+2) = y(t) + u(t)u(t+1), \tag{6.2}$$

which can be entered as follows

```
In[2]:= sysIO = IO[{y[t+2] == y[t] + u[t]u[t+1]}, {u[t]}, {y[t]},
          t, Shift];
        BookForm[sysIO]

Out[3]= y[t+2] = u[t]u[t+1] + y[t]
```

After that, we construct the Ore ring associated with system (6.2).

```
In[4]:= K = DefineOreRing[z, sysIO];
```

Now, we find the polynomial description of the system.

```
In[5]:= {psys, qsys} = Flatten[FromIOToOreP[sysIO]];
        BookForm[psys, K]
        BookForm[qsys, K]
```

Out[6]= $z^2 - 1$

Out[7]= $-u[t]z - u[t+1]$

Next, compute the transfer function, according to Definition 2.5.

```
In[8]:= TFsys = OreR[psys, -qsys];
        BookForm[TFsys, K]
```

Out[9]= $\dfrac{u[t]z + u[t+1]}{z^2 - 1}$

Note that the same visual (output) result can be obtained by using the function `TransferFunction` with the `sysIO` as input argument. However, the generated output objects will be different. Thus, the use of the function `TransferFunction` is more preferable, because it generates the correct object. But in spite of this for our illustrative purposes we use the way presented above.

Suppose that the reference model is given by the following i/o equation

$$y(t+2) = v(t),$$

which has to be entered as follows

```
In[10]:= rmIO = IO[{y[t+2] == v[t]}, {v[t]}, {y[t]}, t, Shift];
         BookForm[sysIO]

Out[11]= y[t+2] = v[t]
```

In order to calculate the transfer function of the reference model, by analogy with system (6.2) we can perform the same steps

In[12]:=
```
{prm, qrm} = Flatten[FromIOToOreP[rmIO]];
TFrm = OreR[prm, -qrm];
BookForm[TFrm, K]
```

Out[14]= $\frac{1}{z^2}$

By (4.3) multiplying transfer functions `TFsys` and `TFrm`, we can find the transfer function representing the compensator

In[15]:=
```
TFcomp = OreMultiply[Power[TFsys, -1], TFrm, K];
BookForm[TFcomp, K]
```

Out[16]= $\frac{z^2 - 1}{u[t+2]\,z^3 + u[t+3]\,z^2}$

Next, we check whether the corresponding one-form is integrable or not. For that purposes we need to preform an intermediate step

In[17]:=
```
dcomp = FromOrePToSpanK[{{TFcomp[[1]]}}, {{-TFcomp[[2]]}},
  {u[t]}, {y[t]}, t, Shift];
BookForm[dcomp]
```

Out[18]= `SpanK[u[t+3] du[t+2] + u[t+2] du[t+3] + dv[t] - dv[t+2]]`

Note that the keyword `SpanK` represents the subspace spanned over the set of one-forms. Now, we can use the function which allows to integrate the one-form

In[19]:=
```
IntegrateOneForms[dcomp]
```

Out[19]= `{u[t+2] u[t+3] + v[t] - v[t+2]}`

The latter after equating to zero becomes an ordinary i/o equation. That is the reason why the obtained result is always in the implicit form. However, in the previous example it was already mentioned that it is not a crucial problem and the standard *Mathematica* functions may be used for obtaining the explicit solution, if the latter is necessary. Note that the MMP for the feedback case may be analyzed in the same way and the similar functions can be used.

The next example demonstrates the step-by-step realization technique, described in Chapter 3.

Example 6.3 Recall the model presented in Example 5.4

$$y(t+2) = 1.2y(t+1) - 0.8y(t) + u(t+1) + \\ + 0.6u(t) + 0.2y(t+1)u(t+1). \quad (6.3)$$

By direct observation of the system (6.3) we may see that it is a special case of the second-order bilinear models. This means that it is possible to apply the theoretical results obtained in [58] under which the system is realizable.

Create the object `IO` for this system

```
In[2]:= ioeq = IO[{y[t+2] == 1.2y[t+1] - 0.8y[t] + u[t+1] +
          0.6u[t] + 0.2y[t+1]u[t+1]}, {u[t]}, {y[t]}, t, Shift];
```

and associate the Ore ring

```
In[3]:= K = DefineOreRing[z, ioeq];
```

Next, equation (6.3) can be described by two polynomials

```
In[4]:= {p, q} = Flatten[FromIOToOreP[ioeq]];
        BookForm[p, K]
        BookForm[q, K]
```

Out[5]= $z^2 + \left(-\frac{6}{5} - \frac{1}{5}u[t+1]\right)z + \frac{4}{5}$

Out[6]= $\left(-1 - \frac{1}{5}y[t+1]\right)z - \frac{3}{5}$

After that, we compute, according to (3.10) for the shift operator based discrete-time case, two sequences of the left quotients as follows

```
In[7]:= BookForm[p1 = LeftQuotient[p, OreP[1, 0], K], K]
        BookForm[p2 = LeftQuotient[p1, OreP[1, 0], K], K]
```

Out[7]= $z + \frac{1}{5}(-6 - u[t])$

Out[8]= 1

```
In[9]:= BookForm[q1 = LeftQuotient[q, OreP[1, 0], K], K]
        BookForm[q2 = LeftQuotient[q1, OreP[1, 0], K], K]
```

Out[9]= $\left(-1 - \frac{y[t]}{5}\right)$

Out[10]= 0

which represent the one-forms $\omega_1 = p_1 \mathrm{d}y + q_1 \mathrm{d}u, \omega_2 = p_2 \mathrm{d}y + q_2 \mathrm{d}u$. Indeed, the same operation can be performed by means of the following function

```
In[11]:= omega =
          SimplifyBasis[StateDifferentialsLeftQuotient[ioeq]];
          BookForm[omega]
```

```
Out[12]= SpanK[ⅆy[t], 5ⅆy[t+1] + (-5 - y[t])ⅆu[t]]
```

Note that the function `SimplifyBasis` was used. This is due to the fact that, according to Remark 3.3, we can preform the linear transformations with basis one-forms in $\mathcal{H}_k$. After integration of the one-forms

```
In[13]:= states = IntegrateOneForms[omega]
```

$$\text{Out[13]= } \{y[t], -\frac{1}{5}u[t](5+y[t])+y[t+1]\}$$

the state equations can be found as

```
In[14]:= {cls, repl} = Realization[ioeq, {x1[t], x2[t]}, states];
         BookForm[cls]
```

$$\text{Out[15]= } \begin{aligned} x_1[t+1] &= u[t](1+0.2x_1[t])+x_2[t] \\ x_2[t+1] &= u[t](1.8+0.24x_1[t])-0.8x_1[t]+1.2x_2[t] \\ y[t] &= x_1[t] \end{aligned}$$

To conclude, one may easily check that system (6.3) is a special case of the system presented in Proposition 3.3 for the case $n = 2$, where the corresponding state equations were derived.

6.7 webMathematica application

Mathematica is a commercial software program conceived by Stephen Wolfram and developed by the Wolfram Research company. It allows to create your own package library. Since *Mathematica* does not contain any built-in functions or packages related to the nonlinear control theory, Institute of Cybernetics created so-called NLControl package addressing all the common problems such as modeling, analysis, synthesis, etc. However, NLControl is not a standalone application and can be used only within the *Mathematica* environment. As a result all the implemented functions cannot be used outside the *Mathematica* making it necessary to be installed on the local computer. In order to overcome this limitation, the Wolfram Research company proposes to use the webMathematica service. It allows a web browser to act as a front end to a remote *Mathematica* server. It means that any application written in *Mathematica* can be remotely accessed via a browser on any platform. Therefore, we have developed an application that partially contains the productivity of NLControl. The latter means that now any researcher, student or just interested in the control theory person can use capabilities of this package through the world-wide-web without necessity to install *Mathematica* software. It should be mentioned that this way of using does not give full access to *Mathematica*, but only to the part of NLControl package.

Note that the Nonlinear Control web site has several restrictions in comparison with the full version of package available for the standard *Mathematica*. First, only the most important functions are accessible through the web site. Second, the application automatically interrupts all the computations lasting longer than 30 seconds. We had to implement this rule, because at the moment there is only one *Mathematica* kernel running on the server, which is shared among all users. It is clear that the interruption time can be set up with respect to the load on the server. Third, equations entered by the user into the fields on the web site have to be in the plain text form. The latter means that, for example, instead of subscripted variables $x_1, x_2, \ldots$ one has to use $x1, x2, \ldots$, and there are similar inconvenient problems with some other symbols naturally used in *Mathematica* environment.

The Nonlinear Control web site may be divided into two basic parts. The first is a web graphical user interface, which provides access to basic functions, help files and additional information. Its structure is simple and consists of menu with several tabs and pages with access points to the corresponding functions. The implemented functions from NLControl are grouped according to the time domain, e.g. continuous- and discrete-time, in order to make their use more convenient for users. The functionality of the visible part of the site was implemented using HTML and JavaScript language. The other (invisible or computational) part consists of files incorporating HTML and Java language. The overall computational (request-response) process can be described as follows.

- User enters the data to be calculated into the corresponding text fields and selects additional possibilities (output format, the presence of the time argument, etc).

- The browser sends a request, which includes symbolic expressions and additional information, to the server side.

- The webMathematica kernel manager acquires the *Mathematica* kernel. The entered data is sent to this kernel.

- The *Mathematica* kernel is initialized with input (request) parameters. It carries out all the necessary calculations and returns the result to the server.

- The webMathematica server sends results back to the browser.

- The obtained result is shown on a separate web page.

Note that requests are sent to the server with webMathematica web pages that are based on two standard Java technologies, which are Java Servlet and JavaServer Pages (JSP). Servlets are special Java programs that run in a Java-enabled web server, which is typically called a "servlet container". JavaServer Pages use a special library of tags that works with *Mathematica*. This library of tags is called the MSP Taglib, for the more precise information we refer the reader to [88] and the references therein. The developed web page is available at [74].

Conclusions

This thesis is devoted to study several important problems in design of nonlinear control systems. In the introductory part of this work we showed the basic idea of constructing the polynomial formalism upon the linear algebraic framework. The latter is based on the theory of differential one-forms. After that the polynomial methods were recalled to solve the modelling and control system design problems. In this chapter we summarize main results presented in the thesis.

First, the computation of the state coordinates for nonlinear i/o equations was addressed for realizable systems. Our goal was to obtain explicit polynomial formulas that are easy to implement. The solution has been obtained by the tools from the theory of skew polynomial rings for the cases of SISO equations, defined in terms of pseudo-linear operator, and MIMO continuous-time systems. The latter requires system to be represented via polynomial description. The explicit formulas to compute the basis one-forms of certain vector space directly from the polynomial system description were presented. If this vector space is not integrable, the i/o equation is not realizable in the state-space form. However, when the vector space is integrable, integration of its exact basis elements results in the desired state coordinates. The approach presented in this thesis, in comparison with the algebraic method based on the solution of the set of nonlinear equations, has a number of advantages. First, it is straightforward, meaning that there is no need to compute step-by-step all the $\mathcal{H}_k$ subspaces, for $k = 1, \ldots, s+2$, in order to find $\mathcal{H}_{s+2}$ as was proposed in [23, 55]. In other words, using the polynomial representation of the system, one can immediately find the subspace $\mathcal{H}_{s+2}$ with one-forms defining the state coordinates. Second, the results of this thesis combine well with those presented in [53, 54] allowing to work out the complete procedure for deriving the minimal state equations starting from the possibly reducible i/o equation. Finally, we have implemented the results of this work in *Mathematica* package NLControl. In this regard, we may conclude that the program code of the introduced algorithms is shorter and more compact compared with the previous methods. Moreover, this approach can be implemented within the most of symbolic programming languages.

Second, the realization problem of the discrete-time i/o bilinear and quadratic control systems in the classical state-space form was addressed. In [58] it was shown that, in general, the bilinear i/o difference equation is not realizable in the state-space form. However, certain structural restrictions in terms of system parameters can guarantee the realizability property. The complete list of necessary and sufficient conditions for the realizability of the third- and fourth-order systems has been given. Moreover, one additional realizable subclass of an arbitrary order was presented. Besides, using the approach presented in this thesis, we have derived a number of sufficient realizability conditions for the second- and third-order quadratic equations.

Third, we have demonstrated that the LPV approach is not applicable to the realization problem in the classical state-space form, unless the nonlinear system is not identified (approximated) directly as an LPV model. The reason comes from the fact that LPV i/o equations are always transformable into the state-space form, whereas this is not always true for nonlinear i/o equations considered in the thesis. The results were illustrated on the basis of the second-order discrete-time bilinear equations. One may conclude that when the i/o bilinear equation is not realizable in the state-space form, according to the theory presented above, no parametrization exists that allows to develop the state equations using the LPV approach. Namely, for all possible parameterizations yielding the i/o LPV model, the corresponding state equations, when the parameters are replaced by the respective variables used in parameterizations are not anymore in the classical state-space form. Moreover, for the realizable i/o equations results depend on the "good/bad" choice of the parameterizations. It means that some parameterizations yield a classical state equations, whereas the others do not. The drawback is that one has not formal rules to distinguish between "good" and "bad" choices *before* making computations. Therefore, at least for the nonlinear realization problem, the LPV approach does not provide the proper tools.

The fourth problem considered in the thesis is the model matching problem. The transfer function formalism was employed to solve the MMP of nonlinear SISO discrete-time systems. The problem was studied within the transfer function approach in which the system is described by the quotient of two polynomials from the skew polynomial ring. Both feedforward and feedback solutions were given in which the input-output map is transfer equivalent to a prespecified model. It was shown that the feedforward solution requires a restrictive integrability condition and therefore does not always exist. Thus, the respective subclass of nonlinear control systems was specified. In contrast, the feedback compensator exists whenever the system is nontrivial. Moreover, the properness of the compensator is justified by inequality of the relative degrees of the system and that of the model.

The last theoretical problem presented in the thesis is the stability of nonlinear discrete-time systems linearized by output feedback. In fact, we determined the criteria ensuring the bounded-input bounded-output stability of the whole control system. This may be divided into two parts. The second part can be solved introducing the assumption on the linear reference model to be Schur stable. However, the first part is more complex. To solve it, the notion of the region of admissible values for the reference signal as well as the algorithm for its determination were explained. The static and dynamic cases were presented separately.

In addition to the theoretical problems listed above, the NLControl package based on *Mathematica* software and its *webMathematica* application are described. The package was created to simplify a large number of symbolic computations arising in the case of nonlinear control systems. The thesis describes one of the subpackages of NLControl. It provides the possibility to solve several modeling and design problems for nonlinear control systems and mathematically relies on the polynomial formalism. All NLControl functions described above are designed for several types of systems. It should be mentioned that the polynomial approach allows to address a multitude of other problems, not discussed in this thesis. In other words, the functionality of NLControl is not restricted only by the examined problems and there are a lof of tasks outside of the thesis that can be solved by means of the tools provided by the package.

To conclude, the author would like to recall the following quotation:

> *Many can imagine or picture the green sun but to make a world inside which the green sun will be credible will probably require hard labour and thought.*
>
> J.R.R. Tolkien.

Appendix

Proofs

Proof of Theorem 3.2

The proof is based on the principle of mathematical induction. First, we show that formula (3.12) holds for $k = 1$. To show this, prove that $\mathcal{H}_1$ in (3.9) may be represented as (3.12) for $k = 1$. In order to simplify the proof note that the recursive formulas (3.10) may be rewritten for $l = 1, \ldots, n$ explicitly as

$$\begin{aligned} p(z) &= z^l \cdot p_l(z) + R_l(z), \quad \deg R_l(z) < l, \\ q(z) &= z^l \cdot q_l(z) + P_l(z), \quad \deg P_l(z) < l \end{aligned} \tag{A.1}$$

with $R_l(z) = \sum_{i=1}^{l} z^{i-1} r_i$ and $P_l(z) = \sum_{j=1}^{l} z^{j-1} \rho_j$.

Suppose $l = n$. According to (3.11), $\omega_n = p_n(z)\mathrm{d}y + q_n(z)\mathrm{d}u$. Due to the structure of the i/o equation, $\deg(p(z)) = n$, and $p(z)$ is monic. Then, it follows from (A.1) that $p_n(z)$ is a left quotient of $p(z)$ and z^n, i.e. $p_n(z) = 1$. Notice that $s < n$ meaning that the quotient of $q(z)$ and z^n is equal to zero. Consequently, $\omega_n = \mathrm{d}y$. Next, take $l = n-1$ and compute ω_{n-1}. Now, it follows from (A.1) that $p_{n-1}(z)$ is a polynomial of the first degree. Thus, $\omega_{n-1} = \mathrm{d}y^{\langle 1 \rangle} + \alpha\omega_n + \beta\mathrm{d}u$ with $\alpha, \beta \in \mathcal{K}^*$, where ω_n and $\mathrm{d}u$ are independent elements in $\mathcal{H}_1$, so ω_{n-1} may be replaced by the simpler one-form $\mathrm{d}y^{\langle 1 \rangle}$. Continuing in the similar manner, it is possible to show that the remaining basis one-forms ω_l, for $l = n-2, \ldots, 1$, in $\mathcal{H}_1$ may be replaced by $\mathrm{d}y^{\langle 2 \rangle}, \ldots, \mathrm{d}y^{\langle n-1 \rangle}$, respectively. As a result, the statement is true for $k = 1$.

Assume now that formula (3.12) holds for r and prove it to be valid for $r+1$. We have to prove that

$$\mathcal{H}_{r+1} = \operatorname{span}_{\mathcal{K}^*} \left\{ \omega_1, \ldots, \omega_n, \mathrm{d}u, \ldots, \mathrm{d}u^{\langle s-r \rangle} \right\}, \tag{A.2}$$

calculated according to formula (3.12), satisfies condition (3.9).

First, note that the one-forms $\omega_1, \ldots, \omega_n, \mathrm{d}u, \ldots, \mathrm{d}u^{\langle s-r \rangle} \in \mathcal{H}_r$, since (3.12) holds for r. Second, we have to prove that the derivatives of the basis one-forms in (A.2) belong to $\mathcal{H}_r$. By (3.11), we have for $l = 1, \ldots, n$

$$\omega_l^{\langle 1 \rangle} = \begin{bmatrix} z \cdot p_l(z) & z \cdot q_l(z) \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}.$$

Using relations (3.10), we get

$$\omega_l^{\langle 1 \rangle} = \begin{bmatrix} p_{l-1}(z) - r_l & q_{l-1}(z) - \rho_l \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}$$

or after reordering the terms

$$\omega_l^{\langle 1 \rangle} = \begin{bmatrix} p_{l-1}(z) & q_{l-1}(z) \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} - \begin{bmatrix} r_l & \rho_l \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}. \tag{A.3}$$

Thus, the one-form $\omega_l^{\langle 1 \rangle}$ is represented as a sum of two terms. For the first term we consider two separate cases. In case $l = 1$, the first term yields $p_0(z)\mathrm{d}y + q_0(z)\mathrm{d}u = p(z)\mathrm{d}y + q(z)\mathrm{d}u = 0$ due to polynomial system description (3.7). In case $l = 2, \ldots, n$, the first term of (A.3) is equal to ω_{l-1} by (3.11) and, therefore, in $\mathcal{H}_r$. The second term of (A.3) is a linear combination of $\mathrm{d}y, \mathrm{d}u \in \mathcal{H}_r$, since the elements of r_l and ρ_l are functions from $\mathcal{K}^*$. Consequently, $\omega_l^{\langle 1 \rangle} \in \mathcal{H}_r$ for $l = 1, \ldots, n$. Finally, we observe that the derivatives of the rest of the basis one-forms in (A.2) are $\mathrm{d}u^{\langle 1 \rangle}, \ldots, \mathrm{d}u^{\langle s-r+1 \rangle}$, which are also in $\mathcal{H}_r$. It should be mentioned that the subspace $\mathcal{H}_{s+2}$ does not contain the elements $\mathrm{d}u^{\langle j \rangle}$, $j = 1, \ldots, s$. Thus, we have shown that $\mathcal{H}_k$, computed according to (3.12) for $k = 1, \ldots, s+1$ and (3.13) for $k = s + 2$, agrees with definition (3.9). □

Proof of Proposition 3.1

System (3.20) can be described in the form (2.11), where

$$\begin{aligned} p(z) = z^3 &- \left(a_1 + c_{11}u^{++} + c_{12}u^{+} + c_{13}u\right) z^2 - \\ &- \left(a_2 + c_{21}u^{++} + c_{22}u^{+} + c_{23}u\right) z - \left(a_3 + c_{31}u^{++} + c_{32}u^{+} + c_{33}u\right) \end{aligned}$$

and

$$\begin{aligned} q(z) = &- \left(b_1 + c_{11}y^{++} + c_{21}y^{+} + c_{31}y\right) z^2 - \\ &- \left(b_2 + c_{12}y^{++} + c_{22}y^{+} + c_{32}y\right) z - \left(b_3 + c_{13}y^{++} + c_{23}y^{+} + c_{33}y\right). \end{aligned}$$

Next, two different cases will be considered separately.

Case 1: $\sigma^{-k}y$, $k \geq 1$ *are the independent variables in the construction of the inversive closure.*

According to Theorem 3.3, we have to check the integrability of all subspaces $\mathcal{H}_k$ for $k = 1, \ldots, 4$. By (2.4), whatever the i/o equation (2.1) for $n = 3$,

$$\mathcal{H}_1 = \operatorname{span}_{\mathcal{K}^*} \left\{ \mathrm{d}y, \mathrm{d}y^+, \mathrm{d}y^{++}, \mathrm{d}u, \mathrm{d}u^+, \mathrm{d}u^{++} \right\}$$

and

$$\mathcal{H}_2 = \operatorname{span}_{\mathcal{K}^*} \left\{ \mathrm{d}y, \mathrm{d}y^+, \mathrm{d}y^{++}, \mathrm{d}u, \mathrm{d}u^+ \right\},$$

which are always integrable. Thus, compute $\mathcal{H}_3 = \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}y, \mathrm{d}y^+, \mathrm{d}u, \omega_1\}$, where ω_1 is given by (3.19) as follows

$$\begin{aligned} \omega_1 = \sigma_c^{-1} \left(\sum_{i=1}^{3} p_i z^{i-1} \mathrm{d}y + \sum_{i=1}^{2} q_i z^{i-1} \mathrm{d}u \right) = \sigma^{-1}(p_3) z^2 \mathrm{d}y + \\ + \sigma^{-1}(p_2) z \mathrm{d}y + \sigma^{-1}(p_1) z^0 \mathrm{d}y + \sigma^{-1}(q_2) z \mathrm{d}u + \sigma^{-1}(q_1) z^0 \mathrm{d}u. \quad \text{(A.4)} \end{aligned}$$

Since $\mathrm{d}y, \mathrm{d}y^+$ and $\mathrm{d}u$ are the basis vectors of the subspace $\mathcal{H}_3$, the second, third and fifth terms can be eliminated from the right-hand side of (A.4). Thus, after simplification we obtain $\mathcal{H}_3 = \operatorname{span}_{\mathcal{K}^*} \{\mathrm{d}y, \mathrm{d}y^+, \mathrm{d}u, \tilde{\omega}_1\}$ with

$$\tilde{\omega}_1 = \mathrm{d}y^{++} - (b_1 + c_{11} y^+ + c_{21} y + c_{31} y^-) \mathrm{d}u^+.$$

Now, it follows from Theorem 2.1 that $\mathcal{H}_3$ is completely integrable iff $\mathrm{d}\tilde{\omega}_1 \wedge \tilde{\omega}_1 \wedge \mathrm{d}y \wedge \mathrm{d}y^+ \wedge \mathrm{d}u = 0$ or alternatively, iff

$$c_{31} \mathrm{d}y^- \wedge \mathrm{d}y \wedge \mathrm{d}y^+ \wedge \mathrm{d}y^{++} \wedge \mathrm{d}u \wedge \mathrm{d}u^+ = 0. \quad \text{(A.5)}$$

From the previous equation we get the condition for $\mathcal{H}_3$ to be integrable, i.e.

$$c_{31} = 0. \quad \text{(A.6)}$$

Next, we examine the integrability of $\mathcal{H}_4 = \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}y, \omega_1, \omega_2\}$ under the condition (A.6), where ω_1 and ω_2 are given by (3.19) as follows

$$\begin{aligned} \omega_1 = \sigma^{-1}(p_3) z^2 \mathrm{d}y + \sigma^{-1}(p_2) z \mathrm{d}y + \sigma^{-1}(p_1) z^0 \mathrm{d}y + \\ + \sigma^{-1}(q_2) z \mathrm{d}u + \sigma^{-1}(q_1) z^0 \mathrm{d}u \quad \text{(A.7)} \end{aligned}$$

and

$$\omega_2 = \sigma^{-2}(p_3) z \mathrm{d}y + \sigma^{-2}(p_2) z^0 \mathrm{d}y + \sigma^{-2}(q_2) z^0 \mathrm{d}u. \quad \text{(A.8)}$$

Start with ω_1. Since $\mathrm{d}y$ is the element of the subspace $\mathcal{H}_4$, the term $\sigma^{-1}(p_1) z^0 \mathrm{d}y$ can be eliminated from the expression of ω_1. Thus, we obtain

$$\begin{aligned} \tilde{\omega}_1 = \mathrm{d}y^{++} - \left(a_1 + c_{11} u^+ + c_{12} u + c_{13} u^- \right) \mathrm{d}y^+ - \\ - \left(b_1 + c_{11} y^+ + c_{21} y \right) \mathrm{d}u^+ - \left(b_2 + c_{12} y^+ + c_{22} y + c_{32} y^- \right) \mathrm{d}u. \quad \text{(A.9)} \end{aligned}$$

Notice that we have chosen y^- as the independent variable of the inversive closure; therefore, we have to get rid of u^- in (A.9). The latter can be done by applying the backward-shift operator σ^{-1} to equation (3.20) once and expressing u^- from it via y^- and the other variables

$$\begin{aligned} u^- = -\left(b_3 + c_{13}y^+ + c_{23}y + c_{33}y^-\right)^{-1} \times \\ \times \left(a_1y^+ + a_2y + a_3y^- + b_1u^+ + b_2u + c_{11}u^+y^+ + \right. \\ \left. + c_{12}uy^+ + c_{21}u^+y + c_{22}uy + c_{32}uy^- - y^{++}\right). \end{aligned} \quad \text{(A.10)}$$

Finally, substituting (A.10) into (A.9), we get

$$\begin{aligned} \tilde{\tilde{\omega}}_1 = \mathrm{d}y^{++} - \left(b_2 + c_{12}y^+ + c_{22}y + c_{32}y^-\right)\mathrm{d}u - \\ - \left(b_1 + c_{11}y^+ + c_{21}y\right)\mathrm{d}u^+ - \\ - \left(a_1 + c_{11}u^+ + c_{12}u - \frac{c_{13}}{b_3 + c_{13}y^+ + c_{23}y + c_{33}y^-} \times \right. \\ \times \left(a_1y^+ + a_2y + a_3y^- + b_1u^+ + b_2u + c_{11}u^+y^+ + \right. \\ \left.\left. + c_{12}uy^+c_{21}u^+y + c_{22}uy + c_{32}uy^- - y^{++}\right)\right)\mathrm{d}y^+. \end{aligned}$$

Next, we continue with ω_2 given by (A.8) that after simplification yields

$$\tilde{\omega}_2 = \mathrm{d}y^+ - \left(b_1 + c_{11}y^+ + c_{21}y\right)\mathrm{d}u.$$

In order to guarantee the integrability of $\mathcal{H}_4 = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y, \tilde{\tilde{\omega}}_1, \tilde{\omega}_2\}$, the following conditions have to be satisfied

$$\mathrm{d}\tilde{\tilde{\omega}}_1 \wedge \tilde{\tilde{\omega}}_1 \wedge \tilde{\omega}_2 \wedge \mathrm{d}y = 0 \quad \text{(A.11)}$$

and

$$\mathrm{d}\tilde{\omega}_2 \wedge \tilde{\tilde{\omega}}_1 \wedge \tilde{\omega}_2 \wedge \mathrm{d}y = 0. \quad \text{(A.12)}$$

From (A.12) one gets

$$\begin{aligned} c_{21}\left(b_1 + c_{11}y^+ + c_{21}y\right)\mathrm{d}y^- \wedge \mathrm{d}y \wedge \mathrm{d}y^+ \wedge \mathrm{d}u \wedge \mathrm{d}u^+ + \\ + c_{21}\mathrm{d}y^- \wedge \mathrm{d}y \wedge \mathrm{d}y^+ \wedge \mathrm{d}y^{++} \wedge \mathrm{d}u = 0, \end{aligned} \quad \text{(A.13)}$$

yielding the condition

$$c_{21} = 0. \quad \text{(A.14)}$$

Taking into account (A.14), condition (A.11) takes the form

$$
\begin{aligned}
&\Big(c_{32} - \big(b_3 + c_{13}y^+ + c_{23}y + c_{33}y^-\big)^{-2} (b_1 + c_{11}y)c_{13}\big((a_3 + c_{32}u)\times \\
&\times \big(b_3 + c_{13}y^+ + c_{23}y + c_{33}y^-\big) - c_{33}\big(a_1y^+ + a_2y + a_3y^- + \\
&+ b_1u^+ + b_2u + c_{11}y^+u^+ + c_{12}y^+u + c_{22}yu + \\
&+ c_{32}y^-u - y^{++}\big)\big)\Big)\mathrm{d}y^- \wedge \mathrm{d}y \wedge \mathrm{d}y^+ \wedge \mathrm{d}y^{++} \wedge \mathrm{d}u + \\
&+ \Big(c_{32}\left(b_1 + c_{11}y^+\right) - \big(b_3 + c_{13}y^+ + c_{23}y + c_{33}y^-\big)^{-2} \times \\
&\times c_{13}(b_1 + c_{11}y)\left(b_1 + c_{11}y^+\right)\big((a_3 + c_{32}u)\big(b_3 + c_{13}y^+ + c_{23}y + c_{33}y^-\big) - \\
&- c_{33}\big(a_1y^+ + a_2y + a_3y^- + b_1u^+ + b_2u + c_{11}y^+u^+ + c_{12}y^+u + \\
&+ c_{22}yu + c_{32}y^-u - y^{++}\big)\big)\Big)\mathrm{d}y^- \wedge \mathrm{d}y \wedge \mathrm{d}y^+ \wedge \mathrm{d}u \wedge \mathrm{d}u^+ = 0
\end{aligned}
\tag{A.15}
$$

that can be guaranteed in three different ways, either by

$$a_3 = c_{32} = c_{33} = 0, \tag{A.16}$$

or by

$$b_1 = c_{11} = c_{32} = 0, \tag{A.17}$$

or by

$$c_{13} = c_{32} = 0. \tag{A.18}$$

Now, summarizing the above cases, we have:

- (A.6), (A.14) and (A.16);
- (A.6), (A.14) and (A.17);
- (A.6), (A.14) and (A.18)

that will yield the conditions (i), (ii) and (v) of Proposition 3.1, respectively. These cases are presented schematically in Figure A.1. Any of these three conditions will yield a realizable system.

Figure A.1 illustrates the basic steps of the proof of the first case. One should *read* it from top to bottom. Above the dotted line we consider the $\mathcal{H}_3$ subspace and the condition for its integrability obtained from equation (A.5). Besides that, under the dotted line the subspace $\mathcal{H}_4$ is considered. In order to find its integrability conditions and to simplify the subsequent calculations, we change the order of the examined equations (A.11) and (A.12), since equation (A.15) is more complex than (A.13) which in turn already implies the condition (A.14). To conclude, passing successively along all arrows from top to bottom and gathering together all obtained conditions, one will get the complete necessary and sufficient conditions for (3.20) to be realizable, for Case 1.

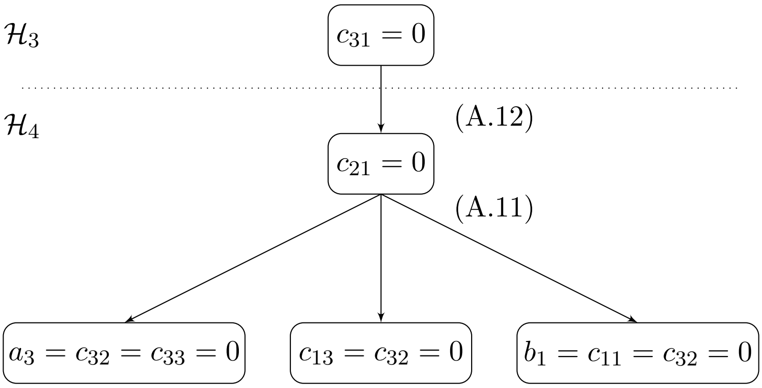

Figure A.1: Realizability conditions for Case 1

Case 2: $\sigma^{-k}u$, $k \geq 1$ *are the independent variables in the construction of the inversive closure.*

In the same manner as in Case 1, we have to check the integrability of $\mathcal{H}_k$ for $k = 1, \ldots, 4$. Again, $\mathcal{H}_1$ and $\mathcal{H}_2$ are integrable by (2.4). Therefore, compute $\mathcal{H}_3 = \operatorname{span}_{\mathcal{K}^*} \{\mathrm{d}y, \mathrm{d}y^+, \mathrm{d}u, \omega_1\}$, where ω_1 can be calculated as in (A.4), and replaced by the simplified one-form

$$\tilde{\omega}_1 = \mathrm{d}y^{++} - \left(b_1 + c_{11}y^+ + c_{21}y + c_{31}y^-\right) \mathrm{d}u^+. \tag{A.19}$$

However, note that unlike in Case 1 we have chosen u^- as the independent variable, and therefore, we have to replace y^- in (A.19) in terms of independent variables. This can be done by applying backward-shift operator to equation (3.20) once and solving it with respect to y^- via u^- and other independent variables as follows

$$\begin{aligned} y^- = &- \left(a_3 + c_{31}u^+ + c_{32}u + c_{33}u^-\right)^{-1} \left(a_1y^+ + a_2y + \right. \\ &+ b_1u^+ + b_2u + b_3u^- + c_{11}y^+u^+c_{12}y^+u + c_{13}y^+u^- + \\ &\left. + c_{21}yu^+ + c_{22}yu + c_{23}yu^- - y^{++}\right). \end{aligned} \tag{A.20}$$

Substituting (A.20) into (A.19), we get

$$\begin{aligned} \tilde{\tilde{\omega}}_1 = \mathrm{d}y^{++} - \Bigg(& b_1 + c_{11}y^+ + c_{21}y - \frac{c_{31}}{a_3 + c_{31}u^+ + c_{32}u + c_{33}u^-} \times \\ & \times \left(a_1y^+ + a_2y + b_1u^+ + b_2u + b_3u^- + c_{11}y^+u^+ + \right. \\ & \left. + c_{12}y^+u + c_{13}y^+u^- + c_{21}yu^+ + c_{22}yu + c_{23}yu^- - y^{++}\right)\Bigg) \mathrm{d}u^+. \end{aligned}$$

Now, it follows from Theorem 2.1 that $\mathcal{H}_3 = \operatorname{span}_{\mathcal{K}^*} \left\{\mathrm{d}y, \mathrm{d}y^+, \mathrm{d}u, \tilde{\tilde{\omega}}_1\right\}$ is completely integrable iff $\mathrm{d}\tilde{\tilde{\omega}}_1 \wedge \tilde{\tilde{\omega}}_1 \wedge \mathrm{d}u \wedge \mathrm{d}y \wedge \mathrm{d}y^+ = 0$. The above condition holds if either

$$b_3 = c_{13} = c_{23} = c_{33} = 0 \tag{A.21}$$

or

$$c_{31} = 0. \tag{A.22}$$

Next, we examine the integrability of $\mathcal{H}_4 = \text{span}_{\mathcal{K}^*}\{\mathrm{d}y, \omega_1, \omega_2\}$ under the conditions (A.21) and (A.22) separately since they will lead to different basis vectors of the subspace $\mathcal{H}_4$.

Case 2.1: *Condition* (A.21) *is satisfied.*

Then, ω_1 and ω_2 are given by (A.7) and (A.8). Since u^- is the independent variable, y^- has to be eliminated like in the computation of $\mathcal{H}_3$. However, now the expression for y^- will be simplified because of (A.21) as follows

$$\begin{aligned} y^- = -\left(a_3 + c_{31}u^+ + c_{32}u\right)^{-1}\big(a_1 y^+ + a_2 y + b_1 u^+ + b_2 u+ \\ + c_{11}y^+u^+ + c_{12}y^+u + c_{21}yu^+ + c_{22}yu - y^{++}\big). \end{aligned} \tag{A.23}$$

Next, we substitute (A.23) into the expressions for ω_1 and ω_2, respectively. In order to guarantee the integrability of $\mathcal{H}_4$, the following conditions have to be satisfied

$$\mathrm{d}\omega_1 \wedge \omega_1 \wedge \omega_2 \wedge \mathrm{d}y = 0 \tag{A.24}$$

and

$$\mathrm{d}\omega_2 \wedge \omega_1 \wedge \omega_2 \wedge \mathrm{d}y = 0. \tag{A.25}$$

One can check that (A.24) is always true, therefore we need only to analyze the condition (A.25), which can be satisfied in two different ways, either by

$$b_2 = c_{12} = c_{22} = c_{32} = 0 \tag{A.26}$$

or by

$$c_{31} = 0. \tag{A.27}$$

Now, we can summarize the obtained conditions:

- (A.21) and (A.26);
- (A.21) and (A.27)

that will yield conditions (iii) and (iv) of Proposition 3.1, respectively. These cases are presented schematically in Figure A.2. Any of these conditions will yield a realizable system.

Case 2.2: *Condition* (A.22) *is satisfied.*

We omit the computations for this case, because they lead to already known conditions (ii) and (v). □

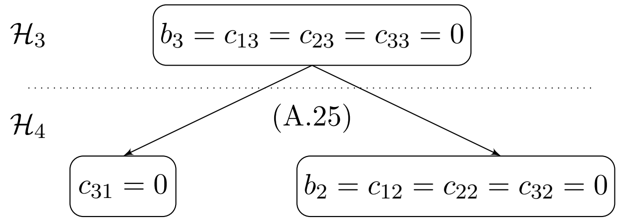

Figure A.2: Realizability conditions for Case 2.1

Proof of Proposition 3.2

The proof of this proposition is based on the same technique as used in the case $n = 3$ and is therefore omitted. Notice only that conditions (i), (ii), (iii), (iv), (vi) and (ix) can be found if we choose $\sigma^{-k}y$, $k \geq 1$ as the independent variables of the inversive closure and the main steps of the proof are represented schematically in Figure A.3.

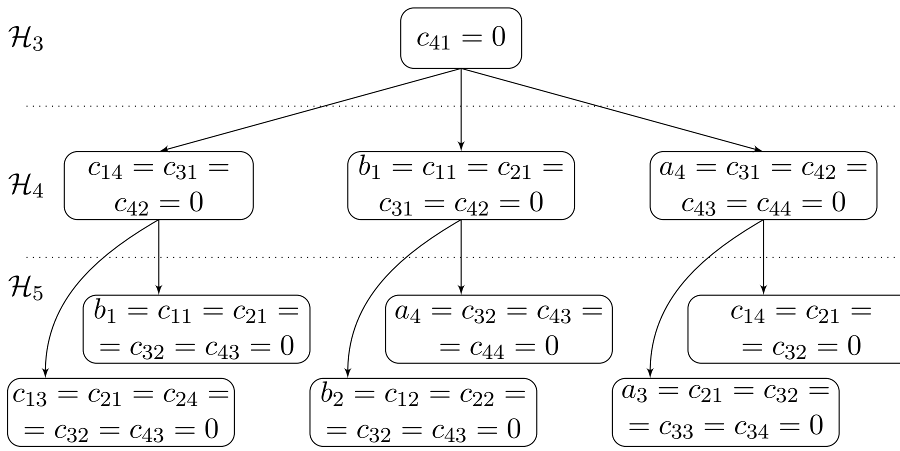

Figure A.3: Realizability conditions: (i)-(iv), (vi) and (ix)

The choice $\sigma^{-k}u$, $k \geq 1$ as independent variables leads to the remaining special cases as summarized in Figure A.4. $\square$

Proof of Proposition 3.3

The result can be checked directly by eliminating the state vector $x(t)$ from the state equations, using for example the technique from [23]. However, we illustrate the basic idea for the second-order system (3.22).

Applying the forward-shift operator σ to the output $y(t)$ of the system and replacing in it $x_1(t)$ by $y(t)$, we get

$$y(t+1) = x_1(t+1) = x_2(t) + (a_1 + c_{11}u(t))y(t) + b_1u(t). \tag{A.28}$$

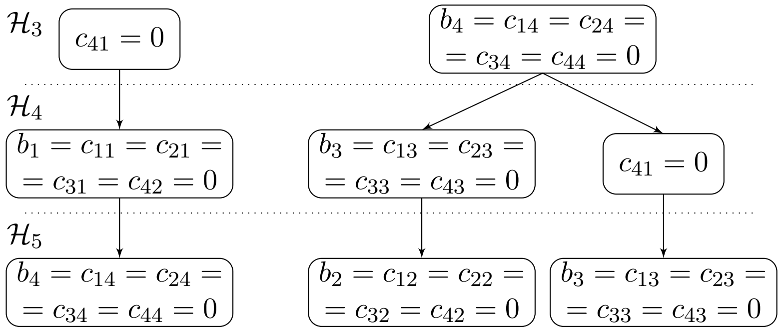

Figure A.4: Realizability conditions: (v), (vii), (viii)

Note that (A.28) can be solved for $x_2(t)$ in terms of the inputs and outputs. Next, we apply σ to (A.28), yielding

$$\begin{aligned} y(t+2) &= x_2(t+1) + (a_1 + c_{11}u(t+1))y(t+1) + b_1u(t+1) = \\ &= (a_2 + c_{22}u(t))y(t) + b_2u(t) + (a_1 + c_{11}u(t+1))y(t+1) + \\ &\quad + b_1u(t+1) + c_{12}u(t)[x_2(t) + (a_1 + c_{11}u(t))y(t) + b_1u(t)], \end{aligned}$$

and substitute $x_2(t)$ from (A.28) into it to get

$$\begin{aligned} y(t+2) &= a_1y(t+1) + a_2y(t) + b_1u(t+1) + b_2u(t) + \\ &\quad + c_{11}u(t+1)y(t+1) + c_{12}u(t)y(t+1) + c_{22}u(t)y(t), \end{aligned}$$

that coincides with (3.21) for $n = 2$.

Note that Proposition 3.3 can be proved in the alternative way. The i/o equations (3.21) may be understood as the special case of the realizable subclass

$$\begin{aligned} y(t+n) &= \varphi_1(y(t+p+l), \ldots, y(t+n-1), u(t+l), \ldots, \\ &\quad u(t+p+l)) + \sum_{i=0}^{n-k-2} \varphi_{n-k-i}(y(t+p+i), \ldots, \\ &\quad y(t+k+i+1), u(t+i), \ldots, u(t+p+i)), \end{aligned} \quad \text{(A.29)}$$

where $k = 0, \ldots, n-1$, $p = 0, \ldots, k$, $l = n-k-1$, given in [60]. Notice that one may take in formula (A.29) parameters mentioned above to be

$k = n - 2$, $p = k$, $l = 1$ and get

$$\begin{aligned}
\varphi_1(\cdot) &= a_1 y(t+n-1) + \sum_{i=1}^{n-1} (b_i + y(t+n-1)c_{1i}) u(t+n-i), \\
\varphi_2(\cdot) &= a_2 y(t+n-2) + b_n u(t) + c_{1n} y(t+n-1) u(t) + \\
&\quad + y(t+n-2) \sum_{i=2}^{n} c_{2i} u(t+n-i),
\end{aligned}$$

in order to find the state-space representation for (3.21). □

Proof of Proposition 3.4

The proof of this proposition is based on the same technique as used for the bilinear case and is therefore omitted. Just notice that conditions (i) and (iii) can be found if we choose $\sigma^{-k}y$, $k \geq 1$ as the independent variables of the inversive closure. The choice $\sigma^{-k}u$, $k \geq 1$ as the independent variables leads to the remaining case (ii). □

Proof of Proposition 3.5

The proof of this proposition is based on the same technique as used for the bilinear case and is therefore omitted. Notice that conditions (i)-(iii) and (vi) can be found if we choose $\sigma^{-k}y$, $k \geq 1$ as the independent variables of the inversive closure. The choice $\sigma^{-k}u$, $k \geq 1$ as the independent variables leads to the remaining special cases. □

Proof of Theorem 3.6

The proof is based on the principle of mathematical induction. Throughout the proof we assume that $i, \nu = 1, \ldots, p$ and $\upsilon = 1, \ldots, m$. First, we show that for $k = 1$ the subspace, calculated according to (3.49), is equivalent to the subspace $\mathcal{H}_1$ defined by (3.44). To show this take in (3.49) $k = 1$ that yields

$$\mathcal{H}_1 = \operatorname{span}_{\mathcal{K}} \left\{ \omega_{i,l}, l = 1, \ldots, n_i, \mathrm{d}u_\upsilon, \ldots, \mathrm{d}u_\upsilon^{(s)} \right\}. \tag{A.30}$$

In order to simplify the presentation note that the recursive formulas (3.48) may be rewritten for $l = 1, \ldots, n_i$ explicitly as

$$\begin{aligned}
p_{i\cdot,0}(z) &= z^l \cdot p_{i\cdot,l}(z) + \Xi_{i\cdot,l}(z), \quad \deg \Xi_{i\cdot,l}(z) < l, \\
q_{i\cdot,0}(z) &= z^l \cdot q_{i\cdot,l}(z) + \Gamma_{i\cdot,l}(z), \quad \deg \Gamma_{i\cdot,l}(z) < l.
\end{aligned} \tag{A.31}$$

Suppose $l = n_i$. According to (3.47), $\omega_{i,n_i} = p_{i\cdot,n_i}(z)\mathrm{d}y + q_{i\cdot,n_i}(z)\mathrm{d}u$. Due to the structure of the i/o equations, $\deg(p_{ii}(z)) = n_i$ and $p_{ii}(z)$

is monic. Then, it follows from (A.31) that $p_{ii,n_i}(z)$ is a left quotient of $p_{ii}(z)$ and z^{n_i}, i.e. $p_{ii,n_i}(z) = 1$. Degrees of all other polynomials $\{p_{i\nu}(z), i \neq \nu, q_{i\cdot}(z)\}$ are strictly less than n_i meaning that the quotient of any polynomial $\{p_{i\nu}(z), i \neq \nu, q_{i\cdot}(z)\}$ and z^{n_i} is zero. Consequently, $\omega_{i,n_i} = \mathrm{d}y_i$. Next, take $l = n_i - 1$ and compute ω_{i,n_i-1}. Now, it follows from (A.31) that $p_{ii,n_i-1}(z)$ is a polynomial of the first degree. Therefore, $\omega_{i,n_i-1} = \mathrm{d}\dot{y}_i + \sum_{\nu=1}^{p} \alpha_{i\nu}\omega_{\nu,n_\nu} + \sum_{\upsilon=1}^{m} \beta_{i\upsilon}\mathrm{d}u_\upsilon$ with $\alpha_{i\nu}, \beta_{i\upsilon} \in \mathcal{K}$. It means that ω_{i,n_i-1} is a linear combination of $\mathrm{d}\dot{y}_i$ and the other elements $\omega_{\nu,n_\nu}, \mathrm{d}u_\upsilon$ from $\mathcal{H}_1$ given by (3.49). Thus, the basis element ω_{i,n_i-1} of $\mathcal{H}_1$ may be replaced by the more simple one-form $\mathrm{d}\dot{y}_i$. Continuing in the similar manner, it is possible to show that the remaining basis one-forms $\omega_{i,l}$, for $l = 1, \ldots, n_i - 2$, in (A.30) may be replaced by more simple ones. As a result, the statement is true for $k = 1$ and (A.30) agrees with (3.44).

Assume now that formula (3.49) holds for r and we prove it to be valid for $r + 1$. We have to prove that

$$\mathcal{H}_{r+1} = \operatorname{span}_{\mathcal{K}} \left\{ \omega_{i,l}, \mathrm{d}u, \ldots, \mathrm{d}u^{(s-k)} \right\}, \tag{A.32}$$

calculated according to formula (3.49), satisfies condition (3.44).

First, the one-forms $\omega_{i,l}, \mathrm{d}u, \ldots,$ $\mathrm{d}u^{(s-r)}$ are in $\mathcal{H}_r$, since we have assumed formula (3.49) to hold for r.

Second, we have to prove that the derivatives of the basis one-forms of (A.32) belong to $\mathcal{H}_r$. By (3.47), we have

$$\dot{\omega}_{i,l} = \begin{bmatrix} z \cdot p_{i\cdot,l}(z) & z \cdot q_{i\cdot,l}(z) \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}$$

for $l = 1, \ldots, n_i$. Next, using relations (3.48), we get

$$\dot{\omega}_{i,l} = \begin{bmatrix} p_{i\cdot,l-1}(z) - \xi_{i\cdot,l} & q_{i\cdot,l-1}(z) - \gamma_{i\cdot,l} \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}$$

or after reordering terms

$$\dot{\omega}_{i,l} = \begin{bmatrix} p_{i\cdot,l-1}(z) & q_{i\cdot,l-1}(z) \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix} - \begin{bmatrix} \xi_{i\cdot,l} & \gamma_{i\cdot,l} \end{bmatrix} \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}. \tag{A.33}$$

From (A.33) it follows that the one-form $\dot{\omega}_{i,l}$ is represented as a sum of two terms. For the first term, we consider two separate cases. In case $l = 1$ the first term yields $p_{i\cdot,0}(z)\mathrm{d}y + q_{i\cdot,0}(z)\mathrm{d}u = p_{i\cdot}(z)\mathrm{d}y + q_{i\cdot}(z)\mathrm{d}u = 0$ due to the polynomial system description (3.46). In case $l = 2, \ldots, n_i$, the first term of (A.33) is equal to $\omega_{i,l-1}$ by (3.47) and, therefore, is in $\mathcal{H}_r$. The second term of (A.33) is a linear combination of $\mathrm{d}y_i, \mathrm{d}u_\upsilon \in \mathcal{H}_r$, since the elements of $\xi_{i\cdot,l}$ and $\gamma_{i\cdot,l}$ are functions from $\mathcal{K}$. Consequently, $\dot{\omega}_{i,l} \in \mathcal{H}_r$ for $l = 1, \ldots, n_i$. Finally, we observe that the derivatives of the rest of the basis one-forms in (A.32) are $\mathrm{d}\dot{u}_v, \ldots, \mathrm{d}u_v^{(s-r+1)}$, which are also in $\mathcal{H}_r$. It should be mentioned that the subspace $\mathcal{H}_{s+2}$ does not contain the elements $\mathrm{d}u^{(j)}$, $j = 1, \ldots, s$. Thus, we have shown that $\mathcal{H}_k$, computed according to (3.49) for $k = 1, \ldots, s+1$ and (3.50) for $k = s+2$, agrees with definition (3.44). □

Proof of Proposition 4.1

The proof is direct consequence from (4.3) yielding

$$R(z) = q_F^{-1}(z) p_F(z) p_G^{-1}(z) q_G(z). \tag{A.34}$$

Alternatively, $R(z)$ is described by the relationship

$$p_R(z)\mathrm{d}u(t) = q_R(z)\mathrm{d}v(t), \tag{A.35}$$

where $p_R(z)$ and $q_R(z)$ are defined by Ore condition (2.9), applied to (A.34). Thus, the i/o equation of the compensator R can be obtained if the one-form (A.35) is integrable. □

Proof of Proposition 4.2

By differentiating equations (4.4), (4.5) and using relations (2.10) and $z^k \mathrm{d}v = \mathrm{d}v(t+k)$, we get (4.1) with

$$p_F(z) = z^{n_F} - \sum_{i=0}^{n_F-1} p_i^F z^i, \qquad p_i^F = \frac{\partial f_1}{\partial y(t+i)},$$
$$q_F(z) = \sum_{j=0}^{s_F} q_j^F z^j, \qquad q_j^F = \frac{\partial f_2}{\partial u(t+j)}$$

and (4.2) with

$$p_G(z) = z^{n_G} - \sum_{i=0}^{n_G-1} p_i^G z^i, \qquad p_i^G = \frac{\partial g_1}{\partial y(t+i)},$$
$$q_G(z) = \sum_{j=0}^{s_G} q_j^G z^j, \qquad q_j^G = \frac{\partial g_2}{\partial v(t+j)},$$

respectively. Note that now in (A.35), $p_R(z) = \beta(z) q_F(z)$ and $q_R(z) = \alpha(z) q_G(z)$, where $\alpha(z)$, $\beta(z)$ are polynomials defined by the left Ore condition as $\beta(z) p_F(z) = \alpha(z) p_G(z)$. According to condition (4.6), the previous equality can be rewritten as $\beta(z)\gamma_F(z)\rho(z) = \alpha(z)\gamma_G(z)\rho(z)$ or $\beta(z)\gamma_F(z) = \alpha(z)\gamma_G(z)$, where $\gamma_F(z), \gamma_G(z)$ can be represented as $\gamma(z) = \sum_{i=0}^{\tau} \gamma_i z^{\tau-i}$,

$\gamma_i \in \mathbb{R}$. So, it follows that $\alpha(z)$ and $\beta(z)$ are also polynomials with real coefficients.

Next, relationship (A.35) can be rewritten as

$$\beta(z) q_F(z) \mathrm{d}u(t) = \alpha(z) q_G(z) \mathrm{d}v(t). \tag{A.36}$$

Notice that the coefficients of the polynomials $q_F(z)$ and $q_G(z)$ do not dependent on $y(t)$ proving the exactness of the one-form (A.36). $\square$

Proof of Proposition 4.3

Necessity: Assume that the proper transfer function $R(z)$ of the compensator R exists that solves the MMP. According to Definition 2.6, it means that

$$\deg p_R(z) \geq \deg q_R(z). \tag{A.37}$$

Next, using the relation

$$G(z) = p_G^{-1}(z) q_G(z) = p_F^{-1}(z) q_F(z) p_R^{-1}(z) q_R(z) = F(z) R(z)$$

and condition (ii) of Proposition 2.3, we get

$$\begin{aligned} \deg q_G(z) &= \deg q_F(z) + \deg q_R(z), \\ \deg p_G(z) &= \deg p_F(z) + \deg p_R(z). \end{aligned} \tag{A.38}$$

Substituting (A.38) into (A.37), we obtain

$$\deg p_G(z) - \deg p_F(z) \geq \deg q_G(z) - \deg q_F(z)$$

or

$$\deg p_G(z) - \deg q_G(z) \geq \deg p_F(z) - \deg q_F(z).$$

Finally, according to Definition 2.7

$$\begin{aligned} \operatorname{rel\,deg} G(z) &= \deg p_G(z) - \deg q_G(z), \\ \operatorname{rel\,deg} F(z) &= \deg p_F(z) - \deg q_F(z) \end{aligned}$$

that yields (4.7).

Sufficiency: Assume that (4.7) holds. Since all the previous steps can be done in the reverse order, we get that the transfer function $R(z)$ is proper. $\square$

Proof of Theorem 4.1

By (4.16),

$$G(z) = (1 - F(z)R_y(z))^{-1}F(z)R_v(z).$$

Next, using (4.11), (4.14) and (4.15), $G(z)$ may be rewritten in the form

$$G(z) = \left(1 - p_F^{-1}(z)q_F(z)p_R^{-1}(z)q_{R_y}(z)\right)^{-1}\left(p_F^{-1}(z)q_F(z)p_R^{-1}(z)q_{R_v}(z)\right)$$

or after multiplying the numerator and denominator by the expression $p_R(z)q_F^{-1}(z)p_F(z)$ from the left

$$G(z) = \left(p_R(z)q_F^{-1}(z)p_F(z) - q_{R_y}(z)\right)^{-1}q_{R_v}(z).$$

Matching the latter to (4.12) results in

$$q_G(z) = q_{R_v}(z), \quad p_G(z) = p_R(z)q_F^{-1}(z)p_F(z) - q_{R_y}(z).$$

One may choose $p_R(z)$ to be $\gamma(z)q_F(z)$, yielding

$$p_G(z) = \gamma(z)p_F(z) - q_{R_y}(z).$$

Under Assumption 4.1, $\gamma(z)$ and $-q_{R_y}(z)$ may be interpreted as right quotient and remainder of skew polynomials $p_G(z)$ and $p_F(z)$, respectively. Thus, from given $p_G(z)$ and $p_F(z)$ one can, by the left division algorithm, determine the infinitesimal description of the compensator

$$\mathrm{d}u(t) = R_v(z)\mathrm{d}v(t) + R_y(z)\mathrm{d}y(t),$$

written alternatively as

$$p_R(z)\mathrm{d}u(t) = q_{R_v}(z)\mathrm{d}v(t) + q_{R_y}(z)\mathrm{d}y(t) \tag{A.39}$$

with $p_G(z) = \gamma(z)p_F(z) - q_{R_y}(z)$, $p_R(z) = \gamma(z)q_F(z)$, $q_{R_v}(z) = q_G(z)$.

Unlike the case of feedforward solution, now the one-form (A.39) is always integrable. Really, equation (A.39) can be rewritten as

$$\gamma(z)q_F(z)\mathrm{d}u(t) = q_G(z)\mathrm{d}v(t) + (\gamma(z)p_F(z) - p_G(z))\mathrm{d}y(t)$$

or

$$\gamma(z)(q_F(z)\mathrm{d}u(t) - p_F(z)\mathrm{d}y(t)) = q_G(z)\mathrm{d}v(t) - p_G(z)\mathrm{d}y(t)$$

yielding that both one-forms $q_F(z)\mathrm{d}u(t) - p_F(z)\mathrm{d}y(t)$ and $q_G(z)\mathrm{d}v(t) - p_G(z)\mathrm{d}y(t)$ are exact. Finally, applying $\gamma(z)$ to an exact one-form results again in an exact one-form. □

Proof of Proposition 4.4

Necessity: Assume that the transfer function $R(z)$, representing the feedback compensator, is proper. According to Definition 2.6, it means that

$$\deg p_R(z) \geq \deg q_{R_v}(z).$$

Next, taking into account that $p_R(z) = \gamma(z)q_F(z)$, $q_G(z) = q_{R_v}(z)$, and using condition (ii) of Proposition 2.3, the previous equation can be rewritten as

$$\deg \gamma(z) + \deg q_F(z) \geq \deg q_G(z).$$

After adding $\deg p_F(z)$ to both sides and regrouping the terms we obtain

$$\deg \gamma(z) + \deg p_F(z) - \deg q_G(z) \geq \deg p_F(z) - \deg q_F(z).$$

Since $p_G(z) = \gamma(z)p_F(z) - q_{R_y}(z)$ and $\deg q_{R_y}(z) = 0$, we get

$$\deg p_G(z) - \deg q_G(z) \geq \deg p_F(z) - \deg q_F(z).$$

Finally, according to Definition 2.7

$$\begin{aligned} \operatorname{rel}\deg G(z) &= \deg p_G(z) - \deg q_G(z), \\ \operatorname{rel}\deg F(z) &= \deg p_F(z) - \deg q_F(z) \end{aligned}$$

that yields (4.17).

Sufficiency: The fact that all the steps in the necessity part of the proof can be done in the reverse order proves the sufficiency.

Proof of Proposition 5.1

The proof is mainly based on the fact recalled in Theorem 5.1. From Assumption 5.1 we know that $v(t)$ is bounded, and using relation (5.5) we can argue that $y(t)$ is bounded too, i.e. $y(t) \subset [a, b]$ and $v(t) \subset [c, d]$. Since $H(y(t), v(t)) : \mathbb{R}^2 \to \mathbb{R}$ is a continuous function, it remains to show that $H(\cdot)$ is defined on a compact set. We know that $[a, b] \times [c, d]$ is compact and the compactness is preserved by continuous function, then it follows that the image $H(y(t), v(t)) \subseteq H([a, b] \times [c, d])$ must also be compact. $\square$

Proof of Theorem 5.4

Necessity: Suppose that the controlled system is bounded-input bounded-output stable. Then the boundedness of the output $y(t)$ is guaranteed by the stable reference model (5.5) under Assumption 5.1. The remaining part of the proof, i.e. the boundedness of the input $u(t)$, relies on the algorithm presented in Chapter 5 for the case of systems linearizable by dynamic output feedback. According to this algorithm, the latter is possible only when $v(t) \in \Omega$.

Sufficiency: Assume that $v(t) \notin \Omega$. The latter means that the functional series (5.11) does not converge, what in turn leads to the fact that the function (5.7) describing the control signal $u(t)$ becomes unbounded. As a result, relying on Definition 5.4 we can conclude that the controlled system is not bounded-input bounded-output stable. $\square$

References

[1] S.A. Abramov, H.Q. Le, and Z. Li. Univariate Ore polynomial rings in computer algebra. *Journal of Mathematical Sciences*, 131(5):5885–5903, 2005.

[2] S.R. Anderson and V. Kadirkamanathan. Modelling and identification of nonlinear deterministic systems in delta-domain. *Automatica*, 43:1859–1868, 2007.

[3] E. Aranda-Bricaire, Ü. Kotta, and C.H. Moog. Linearization of discrete-time systems. *SIAM Journal of Control and Optimization*, 34(6):1999–2023, 1996.

[4] R.S. Baheti, R.R. Mohler, and H.A. Spang III. Second-order correlation method for bilinear system identification. *IEEE Transactions on Automatic Control*, 25(6):1141–1146, 1980.

[5] J.F. Bartee and C. Georgakis. Identification and control of bilinear systems. In *The 11th American Control Conference*, pages 2576–2580, Chicago, IL, USA, June 1992.

[6] Z. Bartosiewicz, Ü. Kotta, E. Pawłuszewicz, and M. Wyrwas. Algebraic formalism of differential one-forms for nonlinear control systems on time scales. *Proceedings of the Estonian Academy of Sciences*, 56(3):264–282, 2007.

[7] S. Beghelli and R. Guidorzi. A new input-output canonical form for multivariable systems. *IEEE Transactions on Automatic Control*, 21(5):692–696, 1976.

[8] J. Belikov, M. Halás, Ü. Kotta, and C.H. Moog. Model matching problem for discrete-time nonlinear systems: Transfer function approach. In *The 9th International Conference on Control and Automation*, pages 360–365, Santiago, Chile, December 2011.

[9] J. Belikov, P. Kotta, and Ü. Kotta. Realizability of bilinear input/output difference equations: Corrections and extensions. In

UKACC International Conference on CONTROL, pages 138–143, Coventry, UK, September 2010.

[10] J. Belikov, P. Kotta, Ü. Kotta, and A.S.I. Zinober. Minimal realization of bilinear and quadratic input-output difference equations in state-space form. *International Journal of Control*, 84(12):2024–2034, 2011.

[11] J. Belikov, Ü. Kotta, and M. Tõnso. State-space realization of nonlinear control systems: Unification and extension via pseudo-linear algebra. *Kybernetika*. Accepted for publication.

[12] J. Belikov, Ü. Kotta, and M. Tõnso. Minimal realization of nonlinear MIMO equations in state-space form: Polynomial approach. In *The 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7735–7740, Orlando, FL, USA, December 2011.

[13] J. Belikov, Ü. Kotta, and M. Tõnso. On applicability of LPV tools for bilinear systems. In *The 9th International Conference on Control and Automation*, pages 366–371, Santiago, Chile, December 2011.

[14] J. Belikov, Ü. Kotta, and M. Tõnso. Symbolic polynomial tools for nonlinear control systems. In *The 7th Vienna International Conference on Mathematical Modelling*, pages 1–6, Vienna, Austria, February 2012.

[15] J. Belikov and E. Petlenkov. Region of admissible values for discrete-time nonlinear control system linearized by output feedback. In *The 18th IFAC World Congress*, pages 209–214, Milan, Italy, August-September 2011.

[16] M.D. Di Benedetto. Nonlinear strong model matching. *IEEE Transactions on Automatic Control*, 35(12):1351–1355, 1990.

[17] M.D. Di Benedetto and J.W. Grizzle. Asymptotic model matching for nonlinear systems. *IEEE Transactions on Automatic Control*, 39(8):1539–1550, 1994.

[18] R.W. Brockett. Feedback invariants for nonlinear systems. In *The 7th IFAC World Congress*, pages 1115–1120, Helsinki, Finland, 1978.

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

[20] Y. Choquet-Bruhat, C. DeWitt-Morette, and M. Dillard-Bleick. *Analysis, Manifolds and Physics, Part I: Basics*. North-Holland, Amsterdam, The Netherlands, 2004.

[21] F. Chyzak, A. Quadrat, and D. Robertz. OreModules: A symbolic package for the study of multidimensional linear systems. In J. Chiasson and J.-J. Loiseau, editors, *Applications of Time Delay Systems, LNCIS*, volume 352, pages 233–264. 2007.

[22] R.M. Cohn. *Difference Algebra.* Wiley-Interscience, New York, USA, 1965.

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

[24] P. Crouch and F. Lamnabhi-Lagarrique. Realizations of input-output differential equations. In *Recent Advances in Mathematical Theory of Systems, Control, Networks and Signal Processing II*, pages 259–264. Mita Press, Tokyo, 1992.

[25] E. Delaleau and W. Respondek. Lowering the orders of derivatives of controls in generalized state space systems. *Journal of Mathematical Systems, Estimation, and Control*, 5(3):1–27, 1995.

[26] H. Diaz and A.A. Desrochers. Modeling of nonlinear discrete-time systems from input-output data. *Automatica*, 24(5):629–641, 1988.

[27] A. Dunoyer, L. Balmer, K.J. Burnham, and D.J.G. James. Systems modelling and control for industrial plant: A bilinear approach. In *The 13th Triennial World Congress*, pages 347–352, San Francisco, CA, USA, June-July 1996.

[28] A. Dzieliński. BIBO stability of NARX models. In *The 7th Mediterranean Conference on Control and Automation*, pages 1198–1206, Haifa, Israel, 1999.

[29] D.L. Elliott. *Bilinear Control Systems: Matricies in Action.* Springer, Dordrecht, Germany, 2009.

[30] B. Farb and R.K. Dennis. *Noncommutative Algebra.* Springer-Verlag, New York, USA, 1993.

[31] D.C. Foley and N. Sadegh. Modelling of nonlinear systems from input-output data for state space realization. In *The 40th IEEE Conference on Decision and Control*, pages 2980–2985, Orlando, FL, USA, December 2001.

[32] K. Forsman. *POLYCON – Computer algebra software for polynomial control systems.* Deptartment of Electrical Engineering, Linköping University, 1993.

[33] S.T. Glad. Nonlinear regulators and Ritt's remainder algorithm. In *Colloque international sur l'analyse des systems dynamiques controlés*, pages 687–692, Lyon, France, July 1990.

[34] J.W. Grizzle. Feedback linearization of discrete-time systems. In *Analysis and Optimization of Systems*, volume 83 of *Lecture Notes in Control and Information Sciences*, pages 271–281. Springer Berlin / Heidelberg, 1986.

[35] J.W. Grizzle. A linear algebraic framework for the analysis of discrete-time nonlinear systems. *SIAM Journal of Control and Optimization*, 31(4):1026–1044, 1991.

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

[37] M. Halás and Ü. Kotta. Transfer functions of discrete-time nonlinear control systems. *Proceedings of the Estonian Academy of Sciences*, 56(4):322–335, 2007.

[38] M. Halás and Ü. Kotta. A note on model matching for nonrealizable nonlinear systems. *Archives of Control Sciences*, 19(4):451–461, 2009.

[39] M. Halás and Ü. Kotta. Realization problem of SISO nonlinear systems: A transfer function approach. In *The 7th International Conference on Control and Automation*, pages 546–551, Christchurch, New Zealand, December 2009.

[40] M. Hálas, Ü. Kotta, Z. Li, H. Wang, and C. Yuan. Submersive rational difference systems and their accessibility. In *The 2009 International Symposium on Symbolic and Algebraic Computation*, pages 175–182, Seoul, Korea, July 2009.

[41] M. Halás, Ü. Kotta, and C.H. Moog. Transfer function approach to the model matching problem of nonlinear systems. In *The 17th IFAC World Congress*, pages 15197–15202, Seoul, Korea, July 2008.

[42] J. Hauser, S. Sastry, and P. Kokotović. Nonlinear control via approximate input-output linearization: The ball and beam example. *IEEE Transactions on Automatic Control*, 37(3):392–398, 1992.

[43] H.J.C. Huijberts. A nonregular solution of the nonlinear dynamic disturbance decoupling problem with an application to a complete solution of the nonlinear model matching problem. *SIAM Journal of Control and Optimization*, 30(2):350–366, 1992.

[44] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, London, UK, 1995.

[45] B. Jakubczyk and W. Respondek. On linearization of control systems. *Bulletin de l'Academie Polonaise des Sciences*, 28:517–522, 1980.

[46] E. Kolchin. *Differential Algebra and Algebraic Groups.* Academic Press, London, UK, 1973.

[47] S. Kotsios. A model matching algorithm for a class of nonlinear discrete systems. A symbolic computational approach. In *The 44th IEEE Conference on Decision and Control, and the European Control Conference*, pages 6603–6607, Seville, Spain, December 2005.

[48] Ü. Kotta. Model matching problem for nonlinear recursive systems. *Proceedings of the Estonian Academy of Sciences*, 46(4):251–261, 1997.

[49] Ü. Kotta. Irreducibility conditions for nonlinear input-output difference equations. In *The 39th IEEE Conference on Decision and Control*, pages 3404–3408, Sydney, NSW, Australia, December 2000.

[50] Ü. Kotta. Decomposition of discrete-time nonlinear control systems. *Proceedings of the Estonian Academy of Sciences*, 54(3):154–161, 2005.

[51] Ü. Kotta, Z. Bartosiewicz, S. Nõmm, and E. Pawłuszewicz. Linear input-output equivalence and row reducedness of discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 56(6):1421–1426, 2011.

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

[53] Ü. Kotta, P. Kotta, and M. Halás. Reduction and transfer equivalence of nonlinear control systems: unification and extension via pseudo-linear algebra. *Kybernetika*, 46(5):831–849, 2010.

[54] Ü. Kotta, P. Kotta, S. Nõmm, and M. Tõnso. Irreducibility conditions for continuous-time multi-input multi-output nonlinear systems. In *The 9th International Conference on Control, Automation, Robotics and Vision*, pages 807–811, Singapore, December 2006.

[55] Ü. Kotta, P. Kotta, M. Tõnso, and M. Halás. State-space realization of nonlinear input-output equations: Unification and extension

via pseudo-linear algebra. In *The 9th International Conference on Control and Automation*, pages 354–359, Santiago, Chile, December 2011.

[56] Ü. Kotta and T. Mullari. Equivalence of different realization methods for higher order nonlinear input-output differential equations. *European Journal of Control*, 11(3):185–193, 2005.

[57] Ü. Kotta and T. Mullari. Equivalence of realizability conditions for nonlinear control systems. *Proceedings of the Estonian Academy of Sciences*, 55(1):24–42, 2006.

[58] Ü. Kotta, S. Nõmm, and A.S.I. Zinober. Classical state space realizability of input-output bilinear models. *International Journal of Control*, 76(12):1224–1232, 2003.

[59] Ü. Kotta and Ü. Nurges. Identification of input-output bilinear systems. In *The 9th World Congress of IFAC: a bridge between control science and technology*, pages 113–117, Budapest, Hungary, July 1984.

[60] Ü. Kotta and N. Sadegh. Two approaches for state space realization of NARMA models: Bridging the gap. *Mathematical and Computer Modelling of Dynamical Systems*, 8(1):21–32, 2002.

[61] Ü. Kotta and M. Tõnso. Removing or lowering the orders of input shifts in discrete-time generalized state-space systems with Mathematica. *Proceedings of the Estonian Academy of Sciences*, 51(4):238–254, 2002.

[62] Ü. Kotta and M. Tõnso. Linear algebraic tools for discrete-time nonlinear control systems with Mathematica. In *Nonlinear and Adaptive Control, NCN4 2001*, Lecture Notes in Control and Information Sciences, pages 195–205. Springer-Verlag Berlin / Heidelberg, 2003.

[63] Ü. Kotta and M. Tõnso. Realization of discrete-time nonlinear input-output equations: Polynomial approach. *Automatica*, 48(2):255–262, 2012.

[64] Ü. Kotta, A.S.I. Zinober, and P. Liu. Transfer equivalence and realization of nonlinear higher order input-output difference equations. *Automatica*, 37(11):1771–1778, 2001.

[65] J.J. Liu, Y.W. Lee, S.Y. Yang, and C.C. Cheng. Non-linear identification and analysis of a HEUI system. In *The 17th IFAC World Congress*, pages 7457–7462, Seoul, Korea, July 2008.

[66] X.Y. Lu and D.J. Bell. Realization theory for differential algebraic input-output systems. *IMA Journal of Mathematical Control & Information*, 10(1):33–47, 1993.

[67] B. Marinescu and H. Bourlés. An intrinsic algebraic setting for poles and zeros of linear time-varying systems. *Systems & Control Letters*, 58(4):248–253, 2009.

[68] R. Marino. On the largest feedback linearizable subsystem. *Systems & Control Letters*, 6(5):345–351, 1986.

[69] J.C. McConnell and J.C. Robson. *Noncommutative Noetherian Rings*. John Wiley and Sons, New York, USA, 1987.

[70] R.R. Mohler. *Nonlinear Systems: Applications to Bilinear Control*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1991.

[71] S. Monaco and D. Normand-Cyrot. The immersion under feedback of a multidimensional discrete-time nonlinear system into a linear system. *International Journal of Control*, 38(1):245–261, 1983.

[72] C.H. Moog, A. Perdon, and G. Conte. Model matching and factorization for nonlinear systems: A structural approach. *SIAM Journal of Control and Optimization*, 29(4):769–785, 1991.

[73] H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, New York, USA, 1990.

[74] Institute of Cybernetics at Tallinn University of Technology. The Nonlinear Control Webpage. Website. `http://webmathematica.cc.ioc.ee/webmathematica/NLControl/`.

[75] M. Ondera and M. Halás. Computer implementation of some recent developments related to generalized transfer functions. In *The 18th IFAC World Congress*, pages 11012–11017, Milano, Italy, August-September 2011.

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

[77] Z.H. Pang, G. Zheng, and C.X. Luo. Augmented state estimation and LQR control for a ball and beam system. In *The 6th IEEE Conference on Industrial Electronics and Applications*, pages 1328–1332, Beijing, China, June 2011.

[78] R. Pothin, Ü. Kotta, and C.H. Moog. Output feedback linearization of nonlinear discrete time systems. In *The IFAC Conference on Control Systems Design*, pages 181–186, Bratislava, Slovak Republic, 2000.

[79] P. Rapisarda and J.C. Willems. State maps for linear systems. *SIAM Journal of Control and Optimization*, 35(3):1053–1091, 1997.

[80] N. Sadegh. Minimal realization of nonlinear systems described by input-output difference equations. *IEEE Transactions on Automatic Control*, 46(5):698–710, 2001.

[81] E. Sontag. *Polynomial Response Maps.* Springer-Verlag, New York, USA, 1979.

[82] E.D. Sontag. On the observability of polynomial systems, I: Finite-time problems. *SIAM Journal of Control and Optimization*, 17(1):139–151, 1979.

[83] R. Starkl and L. del Re. Finite order representations of infinite dimensional bilinear models of non-linear systems. In *The 22nd American Control Conference*, pages 131–136, Denver, CO, USA, June 2003.

[84] K.B. Strömberg, H.T. Toivonen, K.E. Häggiblom, and K.V. Waller. Multivariable nonlinear and adaptive control of a distillation column. *EIChE Journal*, 41(1):195–199, 1995.

[85] M. Tõnso. *Computer algebra tools for modelling, analysis and synthesis for nonlinear control systems.* PhD thesis, Tallinn University of Technology, Faculty of Information Technology, Institute of Cybernetics, Tallinn, Estonia, 2010.

[86] M. Tõnso and Ü. Kotta. Realization of continuous-time nonlinear input-output equations: Polynomial approach. In *Computer Aided Systems Theory - EUROCAST 2009*, Lecture Notes in Control and Information Sciences, pages 633–640. Springer Berlin / Heidelberg, 2009.

[87] M. Tõnso, H. Rennik, J. Belikov, and Ü. Kotta. WebMathematica based tools for continuous-time nonlinear control systems. In *The 17th International Conference on Process Control*, pages 625–633, Štrbské Pleso, Slovak Republic, June 2009.

[88] M. Tõnso, H. Rennik, and Ü. Kotta. WebMathematica based tools for discrete-time nonlinear control systems. *Proceedings of the Estonian Academy of Sciences*, 58(4):224–240, 2009.

[89] R. Tóth. *Modelling and identification of linear parameter-varying systems, an orthonormal basis function approach.* PhD thesis, Delft University of Technology, Delft, The Netherlands, 2008.

[90] R. Tóth, J.C. Willems, P.S.C. Heuberger, and P.M.J. Van den Hof. The behavioral approach to linear parameter-varying systems. *IEEE Transactions on Automatic Control*, 56(11):2499–2514, 2011.

[91] A.J. van der Shaft. On realization of nonlinear systems described by higher-order differential equations. *Mathematical Systems Theory*, 19(1):239–275, 1987.

[92] A.J. van der Shaft. Transformations of nonlinear systems under external equivalence. In *New Trends in Nonlinear Control Theory*, volume 122 of *Lecture Notes in Control and Information Sciences*, pages 33–43. Springer Berlin / Heidelberg, 1989.

[93] S. Willard. *General Topology.* Dover Publications Inc., USA, 2004.

[94] Y.K. Yeo and D.C. Williams. Adaptive bilinear model predictive control. In *The 5th American Control Conference*, pages 1455–1461, Seattle, WA, USA, June 1986.

[95] R. Ylinen. Polynomial systems theory for nonlinear systems. In *The 8th IEEE International Conference on Methods and Models in Automation and Robotics*, pages 95–100, Szczecin, Poland, September 2002.

[96] R. Ylinen. Application of polynomial systems theory to nonlinear systems. In *The 16th IFAC World Congress*, pages 987–992, Prague, Czech Republic, July 2005.

[97] D.L. Yu. Diagnosing simulated faults for an industrial furnance based on bilinear model. *IEEE Transactions on Control Systems Technology*, 8(3):435–442, 2000.

[98] J.I. Yuz and G.C. Goodwin. On sampled-data models for nonlinear systems. *IEEE Transactions on Automatic Control*, 50(10):1477–1489, 2005.

[99] J. Zhang, C.H. Moog, and X. Xia. Realization of multivariable nonlinear systems via the approaches of differential forms and differential algebra. *Kybernetika*, 46(5):799–830, 2010.

[100] Y. Zheng and L. Cao. Transfer function description for nonlinear systems. *Journal of East China Normal University*, 2:15–26, 1995.

[101] Y. Zheng, J.C. Willems, and C. Zhang. A polynomial approach to nonlinear system controllability. *IEEE Transactions on Automatic Control*, 46(11):1782–1788, 2001.

List of Publications

1. J. Belikov, M. Halás, Ü. Kotta, and C.H. Moog. Model matching problem for discrete-time nonlinear systems: Transfer function approach. In *The 9th International Conference on Control and Automation*, pages 360–365, Santiago, Chile, December 2011.
2. J. Belikov, P. Kotta, and Ü. Kotta. Realizability of bilinear input/output difference equations: Corrections and extensions. In *UKACC International Conference on CONTROL*, pages 138–143, Coventry, UK, September 2010.
3. J. Belikov, P. Kotta, Ü. Kotta, and A.S.I. Zinober. Minimal realization of bilinear and quadratic input-output difference equations in state-space form. *International Journal of Control*, 84(12):2024–2034, 2011.
4. J. Belikov, Ü. Kotta, and M. Tõnso. State-space realization of nonlinear control systems: Unification and extension via pseudo-linear algebra. *Kybernetika.* Accepted for publication.
5. J. Belikov, Ü. Kotta, and M. Tõnso. Minimal realization of nonlinear MIMO equations in state-space form: Polynomial approach. In *The 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7735–7740, Orlando, FL, USA, December 2011.
6. J. Belikov, Ü. Kotta, and M. Tõnso. On applicability of LPV tools for bilinear systems. In *The 9th International Conference on Control and Automation*, pages 366–371, Santiago, Chile, December 2011.
7. J. Belikov, Ü. Kotta, and M. Tõnso. Symbolic polynomial tools for nonlinear control systems. In *The 7th Vienna International Conference on Mathematical Modelling*, pages 1–6, Vienna, Austria, February 2012.
8. J. Belikov and E. Petlenkov. Region of admissible values for discrete-time nonlinear control system linearized by output feedback. In *The 18th IFAC World Congress*, pages 209–214, Milan, Italy, August-September 2011.

9. M. Tõnso, H. Rennik, J. Belikov, and Ü. Kotta. WebMathematica based tools for continuous-time nonlinear control systems. In *The 17th International Conference on Process Control*, pages 625–633, Štrbské Pleso, Slovak Republic, June 2009.

Kokkuvõte

Käesolevas väitekirjas on uuritud mittelineaarseid süsteeme kasutades polünoommeetodeid. Töö esimeses ja teises peatükis on esitatud polünoommeetodite teoreetilised alused. Töö põhitulemused on esitatud peatükkides 3-6.

Peatükis 3 lahendatakse mittelineaarsete sisend-väljund süsteemide olekuvõrrandite esitamise probleem, kasutades selleks diferentsiaalvormidel ja mittekommutatiivsetesse ringidesse kuuluvatel polünoomidel baseeruvat algebralist formalismi. On tuletatud valemid, mis võimaldavad välja kirjutada olekuvõrrandite diferentsiaalid otse süsteemi polünoomesitusest. Probleemi on vaadeldud SISO võrrandite juhul, mis on defineeritud pseudolineaarse operaatori teriminites. Antud lähenemine lubab üldistada ja laiendada varem eraldi saadud tulemusi pidevaja- ja diskreetaja süsteemidele. Samuti, on esitatud kolmandat ja neljandat järku bilineaarsete süsteemide realiseeritavate struktuuride täisloend. Kõrgemat järku mudelite jaoks on tuletatud uus realiseeritav alamklass. Lisaks on teist ja kolmandat järku ruutmudelite jaoks leitud realiseeritavuse piisavad timgimused. Polünoomvalemid on üldistatud MIMO süsteemide jaoks.

Peatükis 4 on uuritud mittelineaarsete SISO diskreetaja süsteemide etaloonmudeliga juhtimist. On analüüsitud nii avatud kui ka suletud süsteemide lahendeid. Probleemi uurimisel on kasutatud ülekandefunktsioonide meetodit.

Peatükk 5 on pühendatud tagasisidega lineariseeritud mittelineaarsete SISO diskreetaja süsteemide stabiilsuse analüüsile.

Töö viimane osa kirjeldab ülaltoodud teooreetiliste tulemuste rakendamist tarkvarapaketis NLControl, mis on arendatud *Mathematica* keskkonna jaoks. NLControl võimaldab lahendada mittelineaarsete juhtimissüsteemide modellerimise, analüüsi ja sünteesiga seotud ülesandeid.

Abstract

The present work can be divided into several independent parts. The main contributions are presented in Chapters 3-6. While the first three of them are devoted to the purely theoretical results, the last one is written with a practical orientation.

The algebraic approach based on the theory of differential one-forms together with polynomial formalism are applied in Chapter 3 to realization problem of nonlinear i/o equations in the classical state-space form. The explicit formulas, which allows to write out the differentials of the state equations directly from the polynomial description of the system, are presented. First, the problem is considered for the case of SISO equations, defined in terms of the pseudo-linear operator, generalizing and extending results obtained separately for the continuous- and discrete-time systems. Second, the complete lists of realizable structures for the case of the third- and fourth-order bilinear systems are given. In addition, the new realizable subclass is constructed. Moreover, the sufficient realizability conditions for the second- and third-order quadratic equations are derived. Third, the applicability of the LPV tools for the realizability of nonlinear system is analyzed on the basis of the second-order discrete-time bilinear equations. Finally, in the same manner as in the case of equations defined in terms of pseudo-linear operator, the corresponding polynomial formulas for the MIMO continuous-time systems are presented. Next, in Chapter 4 the model matching problem of nonlinear SISO discrete-time systems is considered. Both feedforward and feedback solutions are analyzed. The problem is studied within the transfer function approach. After that Chapter 5 is devoted to the stability problem of nonlinear SISO discrete-time systems linearized by output feedback. Both static and dynamic cases are studied. Finally, Chapter 6 summarizes the results of the previous chapters linking them together in the form of a practical implementation in the NLControl package.

Elulookirjeldus

1. Isikuandmed

Ees- ja perekonnanimi	Juri Belikov
Sünniaeg ja -koht	15.03.1985, Balta, Ukraina
Kodakondsus	Eesti
Perekonnaseis	abielus

2. Kontaktandmed

Aadress	Õismäe tee 118-44, Tallinn, 13513
Telefon	+372 55923042
E-posti aadress	jbelikov@cc.ioc.ee

3. Hariduskäik

Õppeasutus (nimetus lõpetamise ajal)	Lõpetamise aeg	Haridus (eriala/kraad)
Tallinna Ülikool	2006	matemaatika, B.Sc., cum laude
Tallinna Tehnikaülikool	2008	arvuti- ja süsteemitehnika, M.Sc.

4. Keelteoskus

Keel	Tase
Eesti	kõrgtase
Inglise	kesktase
Vene	emakeel

5. Teenistuskäik

Töötamise aeg	Tööandja nimetus	Ametikoht
2006 – ...	Küberneetika Instituut, TTÜ	insener
2010 – ...	Automaatikainstituut, TTÜ	insener

6. Teadustegevus

   Ajakirja- ja konverentisartiklite loetelu on toodud ingliskeelse CV juures.

7. Kaitstud lõputööd

   Kasumi maksimiseerimine Cobb-Douglase tootmisfunktsiooni korral, B.Sc., Tallinna Ülikool, 2006.

   Mittelineaarsete diskreetaja mudelite identifitseerimine ja süntees mudelil põhineva juhtimise jaoks, M.Sc., Tallinna Tehnikaülikool, 2008.

8. Teadustöö põhisuunad

   Mittelineaarsete juhtimissüsteemide teooria.

Curriculum vitae

1. Personal data

Name	Juri Belikov
Date and place of birth	15.03.1985, Balta, Ukraine
Citizenship	Estonian
Marital status	married

2. Contact information

Address	Õismäe tee 118-44, Tallinn, 13513
Phone	+372 55923042
E-mail	jbelikov@cc.ioc.ee

3. Education

Educational institution	Graduation year	Education (field of study/degree)
Tallinn University	2006	mathematics, B.Sc., cum laude
Tallinn University of Technology	2008	computer and systems engineering, M.Sc.

4. Language competence/skills

Language	Level
Estonian	fluent
English	average
Russian	native

5. Professional Employment

Period	Organization	Position
2006 – ...	Institute of Cybernetics, TUT	engineer
2010 – ...	Department of Computer Control, TUT	engineer

6. Scientific work

1. J. Belikov, M. Halás, Ü. Kotta, and C.H. Moog. Model matching problem for discrete-time nonlinear systems: Transfer function approach. In *The 9th International Conference on Control and Automation*, pages 360–365, Santiago, Chile, December 2011.

2. J. Belikov, P. Kotta, and Ü. Kotta. Realizability of bilinear input/output difference equations: Corrections and extensions. In *UKACC International Conference on CONTROL*, pages 138–143, Coventry, UK, September 2010.

3. J. Belikov, P. Kotta, Ü. Kotta, and A.S.I. Zinober. Minimal realization of bilinear and quadratic input-output difference equations in state-space form. *International Journal of Control*, 84(12):2024–2034, 2011.

4. J. Belikov, Ü. Kotta, and M. Tõnso. State-space realization of nonlinear control systems: Unification and extension via pseudo-linear algebra. *Kybernetika*. Accepted for publication.

5. J. Belikov, Ü. Kotta, and M. Tõnso. An explicit formula for computation of the state coordinates for nonlinear i/o equation. In *The 18th IFAC World Congress*, pages 7221–7226, Milan, Italy, August-September 2011.

6. J. Belikov, Ü. Kotta, and M. Tõnso. Minimal realization of nonlinear MIMO equations in state-space form: Polynomial approach. In *The 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7735–7740, Orlando, FL, USA, December 2011.

7. J. Belikov, Ü. Kotta, and M. Tõnso. On applicability of LPV tools for bilinear systems. In *The 9th International Conference on Control and Automation*, pages 366–371, Santiago, Chile, December 2011.

8. J. Belikov, Ü. Kotta, and M. Tõnso. Symbolic polynomial tools for nonlinear control systems. In *The 7th Vienna International Conference on Mathematical Modelling*, pages 1–6, Vienna, Austria, February 2012.

9. J. Belikov and E. Petlenkov. Calculation of the control signal in MIMO NN-based ANARX models: Analytical approach. In *The 10th International Conference of Control, Automation, Robotics and Vision*, pages 2196–2201, Hanoi, Vietnam, December 2008.

10. J. Belikov and E. Petlenkov. Dynamic pole placement based control of nonlinear discrete-time systems with input delay. In *The 3rd IEEE Multi-conference on Systems and Control*, pages 394–399, St. Petersburg, Russia, July 2009.

11. J. Belikov and E. Petlenkov. Model reference control of nonlinear MIMO systems by dynamic output feedback linearization of ANARX models. In *The 7th IEEE International Conference on Control and Automation*, pages 536–541, Christchurch, New Zealand, December 2009.

12. J. Belikov and E. Petlenkov. Model reference control of nonlinear TITO systems by dynamic output feedback linearization of neural network based ANARX models. In *The 3rd IEEE Multi-conference on Systems and Control*, pages 1820–1825, St. Petersburg, Russia, July 2009.

13. J. Belikov and E. Petlenkov. NN-ANARX model based control of nonlinear discrete-time systems with input delays. In *The 3rd IEEE Multi-conference on Systems and Control*, pages 1039–1044, St. Petersburg, Russia, July 2009.

14. J. Belikov and E. Petlenkov. Neuro-fuzzy dynamic pole placement control of nonlinear discrete-time systems. In *The International Joint Conference on Neural Networks*, pages 1577–1582, San Jose, CA, USA, July-August 2011.

15. J. Belikov and E. Petlenkov. Region of admissible values for discrete-time nonlinear control system linearized by output feedback. In *The 18th IFAC World Congress*, pages 209–214, Milan, Italy, August-September 2011.

16. J. Belikov, K. Vassiljeva, E. Petlenkov, and S. Nõmm. A novel taylor series based approach for control computation in NN-ANARX structure based control of nonlinear systems. In *The 27th Chinese Control Conference*, pages 474–478, Kunming, China, July 2008.

17. S. Nõmm and J. Belikov. Online identification of the system order with ANARX structure. In *The International Conference on Adaptive and Intelligent Systems*, pages 5–15, Klagenfurt, Austria, September 2011.

18. S. Nõmm, E. Petlenkov, J. Vain, F. Miyawaki, K. Yoshimitsu, and J. Belikov. Recognition of the surgeon's motions during endoscopic operation by statistics based algorithm and neural networks based

ANARX models. In *The 17th IFAC World Congress*, pages 14773–14778, Seoul, South Korea, July 2008.

19. S. Nõmm, K. Vassiljeva, E. Petlenkov, and J. Belikov. Structure identification of NN-ANARX model by genetic algorithm with combined cross-correlation-test based evaluation function. In *The 9th International Conference on Control and Automation*, pages 65–70, Santiago, Chile, December 2011.

20. E. Petlenkov and J. Belikov. NN-ANARX structure for dynamic output feedback linearization of nonlinear SISO and MIMO systems: Neural networks based approach. In *The 26th Chinese Control Conference*, pages 138–145, Zhangjiajie, China, July 2007.

21. E. Petlenkov and J. Belikov. Neural network based dynamic pole placement control of nonlinear systems. In *The 7th IEEE International Conference on Control and Automation*, pages 410–415, Christchurch, New Zealand, December 2009.

22. E. Petlenkov, J. Belikov, S. Nõmm, and M. Wyrwas. Dynamic output feedback linearization based adaptive control of nonlinear MIMO systems. In *The 27th American Control Conference*, pages 3446–3451, Seattle, WA, USA, June 2008.

23. A. Tepljakov, E. Petlenkov, and J. Belikov. FOMCON: a MATLAB toolbox for fractional-order system identification and control. *International Journal of Microelectronics and Computer Science*, 2(2):51–62, 2011.

24. A. Tepljakov, E. Petlenkov, and J. Belikov. FOMCON: Fractional order modeling and control toolbox for MATLAB. In *The 18th International Conference Mixed Design of Integrated Circuits and Systems*, pages 684–689, Gliwice, Poland, June 2011.

25. M. Tõnso, H. Rennik, J. Belikov, and Ü. Kotta. WebMathematica based tools for continuous-time nonlinear control systems. In *The 17th International Conference on Process Control*, pages 625–633, Štrbské Pleso, Slovak Republic, June 2009.

26. K. Vassiljeva, J. Belikov, and E. Petlenkov. Genetic algorithm based structure identification for feedback control of nonlinear MIMO systems. In *The International Conference on Adaptive and Intelligent Systems*, pages 215–226, Klagenfurt, Austria, September 2011.

27. K. Vassiljeva, J. Belikov, and E. Petlenkov. Neural networks based minimal or reduced model representation for control of nonlinear MIMO systems. In *The International Joint Conference on Neural Networks*, pages 1706–1713, San Jose, CA, USA, July-August 2011.

28. K. Vassiljeva, E. Petlenkov, and J. Belikov. Automated neural network model selection algorithm for feedback linearization based control. In *The 12th Biennial Baltic Electronics Conference*, pages 235–238, Tallinn, Estonia, October 2010.

29. K. Vassiljeva, E. Petlenkov, and J. Belikov. Neural network based minimal state-space representation of nonlinear MIMO systems for feedback control. In *The 11th International Conference on Control, Automation, Robotics and Vision*, pages 2191–2196, Singapore, December 2010.

30. K. Vassiljeva, E. Petlenkov, and J. Belikov. State-space control of nonlinear systems identified by ANARX and neural network based SANARX models. In *The World Congress on Computational Intelligence*, pages 3816–3823, Barcelona, Spain, July 2010.

7. Defended theses

   Maximization of Profit in Case of Cobb-Douglas Production Function, B.Sc., Tallinn University, 2006.

   Synthesis and identification of nonlinear discrete-time models for model based control, M.Sc., Tallinn University of Technology, 2008.

8. Main areas of scientific work/Current research topics

   Nonlinear control systems.

DISSERTATIONS DEFENDED AT TALLINN UNIVERSITY OF TECHNOLOGY ON INFORMATICS AND SYSTEM ENGINEERING

1. **Lea Elmik**. Informational Modelling of a Communication Office. 1992.
2. **Kalle Tammemäe**. Control Intensive Digital System Synthesis. 1997.
3. **Eerik Lossmann**. Complex Signal Classification Algorithms, Based on the Third-Order Statistical Models. 1999.
4. **Kaido Kikkas**. Using the Internet in Rehabilitation of People with Mobility Impairments - Case Studies and Views from Estonia. 1999.
5. **Nazmun Nahar**. Global Electronic Commerce Process: Business-to-Business. 1999.
6. **Jevgeni Riipulk**. Microwave Radiometry for Medical Applications. 2000.
7. **Alar Kuusik**. Compact Smart Home Systems: Design and Verification of Cost Effective Hardware Solutions. 2001.
8. **Jaan Raik**. Hierarchical Test Generation for Digital Circuits Represented by Decision Diagrams. 2001.
9. **Andri Riid**. Transparent Fuzzy Systems: Model and Control. 2002.
10. **Marina Brik**. Investigation and Development of Test Generation Methods for Control Part of Digital Systems. 2002.
11. **Raul Land**. Synchronous Approximation and Processing of Sampled Data Signals. 2002.
12. **Ants Ronk**. An Extended Block-Adaptive Fourier Analyser for Analysis and Reproduction of Periodic Components of Band-Limited Discrete-Time Signals. 2002.
13. **Toivo Paavle**. System Level Modeling of the Phase Locked Loops: Behavioral Analysis and Parameterization. 2003.
14. **Irina Astrova**. On Integration of Object-Oriented Applications with Relational Databases. 2003.
15. **Kuldar Taveter**. A Multi-Perspective Methodology for Agent-Oriented Business Modelling and Simulation. 2004.

16. **Taivo Kangilaski**. Eesti Energia käiduhaldussüsteem. 2004.
17. **Artur Jutman**. Selected Issues of Modeling, Verification and Testing of Digital Systems. 2004.
18. **Ander Tenno**. Simulation and Estimation of Electro-Chemical Processes in Maintenance-Free Batteries with Fixed Electrolyte. 2004.
19. **Oleg Korolkov**. Formation of Diffusion Welded Al Contacts to Semiconductor Silicon. 2004.
20. **Risto Vaarandi**. Tools and Techniques for Event Log Analysis. 2005.
21. **Marko Koort**. Transmitter Power Control in Wireless Communication Systems. 2005.
22. **Raul Savimaa**. Modelling Emergent Behaviour of Organizations. Time-Aware, UML and Agent Based Approach. 2005.
23. **Raido Kurel**. Investigation of Electrical Characteristics of SiC Based Complementary JBS Structures. 2005.
24. **Rainer Taniloo**. Ökonoomsete negatiivse diferentsiaaltakistusega astmete ja elementide disainimine ja optimeerimine. 2005.
25. **Pauli Lallo**. Adaptive Secure Data Transmission Method for OSI Level I. 2005.
26. **Deniss Kumlander**. Some Practical Algorithms to Solve the Maximum Clique Problem. 2005.
27. **Tarmo Veskioja**. Stable Marriage Problem and College Admission. 2005.
28. **Elena Fomina**. Low Power Finite State Machine Synthesis. 2005.
29. **Eero Ivask**. Digital Test in WEB-Based Environment 2006.
30. **Виктор Войтович**. Разработка технологий выращивания из жидкой фазы эпитаксиальных структур арсенида галлия с высоковольтным p-n переходом и изготовления диодов на их основе. 2006.
31. **Tanel Alumäe**. Methods for Estonian Large Vocabulary Speech Recognition. 2006.
32. **Erki Eessaar**. Relational and Object-Relational Database Management Systems as Platforms for Managing Softwareengineering Artefacts. 2006.

33. **Rauno Gordon**. Modelling of Cardiac Dynamics and Intracardiac Bio-impedance. 2007.
34. **Madis Listak**. A Task-Oriented Design of a Biologically Inspired Underwater Robot. 2007.
35. **Elmet Orasson**. Hybrid Built-in Self-Test. Methods and Tools for Analysis and Optimization of BIST. 2007.
36. **Eduard Petlenkov**. Neural Networks Based Identification and Control of Nonlinear Systems: ANARX Model Based Approach. 2007.
37. **Toomas Kirt**. Concept Formation in Exploratory Data Analysis: Case Studies of Linguistic and Banking Data. 2007.
38. **Juhan-Peep Ernits**. Two State Space Reduction Techniques for Explicit State Model Checking. 2007.
39. **Innar Liiv**. Pattern Discovery Using Seriation and Matrix Reordering: A Unified View, Extensions and an Application to Inventory Management. 2008.
40. **Andrei Pokatilov**. Development of National Standard for Voltage Unit Based on Solid-State References. 2008.
41. **Karin Lindroos**. Mapping Social Structures by Formal Non-Linear Information Processing Methods: Case Studies of Estonian Islands Environments. 2008.
42. **Maksim Jenihhin**. Simulation-Based Hardware Verification with High-Level Decision Diagrams. 2008.
43. **Ando Saabas**. Logics for Low-Level Code and Proof-Preserving Program Transformations. 2008.
44. **Ilja Tšahhirov**. Security Protocols Analysis in the Computational Model - Dependency Flow Graphs-Based Approach. 2008.
45. **Toomas Ruuben**. Wideband Digital Beamforming in Sonar Systems. 2009.
46. **Sergei Devadze**. Fault Simulation of Digital Systems. 2009.
47. **Andrei Krivošei**. Model Based Method for Adaptive Decomposition of the Thoracic Bio-Impedance Variations into Cardiac and Respiratory Components. 2009.
48. **Vineeth Govind**. DfT-Based External Test and Diagnosis of Mesh-like Networks on Chips. 2009.

49. **Andres Kull**. Model-Based Testing of Reactive Systems. 2009.

50. **Ants Torim**. Formal Concepts in the Theory of Monotone Systems. 2009.

51. **Erika Matsak**. Discovering Logical Constructs from Estonian Children Language. 2009.

52. **Paul Annus**. Multichannel Bioimpedance Spectroscopy: Instrumentation Methods and Design Principles. 2009.

53. **Maris Tõnso**. Computer Algebra Tools for Modelling, Analysis and Synthesis for Nonlinear Control Systems. 2010.

54. **Aivo Jürgenson**. Efficient Semantics of Parallel and Serial Models of Attack Trees. 2010.

55. **Erkki Joasoon**. The Tactile Feedback Device for Multi-Touch User Interfaces. 2010.

56. **Jürgo-Sören Preden**. Enhancing Situation - Awareness Cognition and Reasoning of Ad-Hoc Network Agents. 2010.

57. **Pavel Grigorenko**. Higher-Order Attribute Semantics of Flat Languages. 2010.

58. **Anna Rannaste**. Hierarcical Test Pattern Generation and Untestability Identification Techniques for Synchronous Sequential Circuits. 2010.

59. **Sergei Strik**. Battery Charging and Full-Featured Battery Charger Integrated Circuit for Portable Applications. 2011.

60. **Rain Ottis**. A Systematic Approach to Offensive Volunteer Cyber Militia. 2011.

61. **Natalja Sleptšuk**. Investigation of the Intermediate Layer in the Metal-Silicon Carbide Contact Obtained by Diffusion Welding. 2011.

62. **Martin Jaanus**. The Interactive Learning Environment for Mobile Laboratories. 2011.

63. **Argo Kasemaa**. Analog Front End Components for Bio-Impedance Measurement: Current Source Design and Implementation. 2011.

64. **Kenneth Geers**. Strategic Cyber Security: Evaluating Nation-State Cyber Attack Mitigation Strategies. 2011.

65. **Riina Maigre**. Composition of Web Services on Large Service Models. 2011.

66. **Helena Kruus**. Optimization of Built-in Self-Test in Digital Systems. 2011.

67. **Gunnar Piho**. Archetypes Based Techniques for Development of Domains, Requirements and Sofware. 2011.

68. **Juri Gavšin**. Intrinsic Robot Safety Through Reversibility of Actions. 2011.

69. **Dmitri Mihhailov**. Hardware Implementation of Recursive Sorting Algorithms Using Tree-like Structures and HFSM Models. 2012.

70. **Anton Tšertov**. System Modeling for Processor-Centric Test Automation. 2012.

71. **Sergei Kostin**. Self-Diagnosis in Digital Systems. 2012.

72. **Mihkel Tagel**. System-Level Design of Timing-Sensitive Network-on-Chip Based Dependable Systems. 2012.