Dead-Beat Control for Polynomial Systems

Dragan Nešić

Bachelor of Engineering

August 1996

A thesis submitted for the degree of Doctor of Philosophy

of the Australian National University

Department of Systems Engineering

Research School of Information Science and Engineering

The Australian National University

Dead-Beat Control for Polynomial Systems

Dragan Nešić

Bachelor of Engineering

August 1996

A thesis submitted for the degree of Doctor of Philosophy

of the Australian National University

Department of Systems Engineering

Research School of Information Science and Engineering

The Australian National University

To my family

Ksenija and Nina

and my parents

Dragica and Dušan

Statement of originality

The contents of this thesis are the results of the original research unless otherwise stated and have not been submitted for a higher degree at any other university or institution. The material described in this thesis has been obtained under the supervision of Prof. I. M. Y. Mareels. Some results have been obtained in cooperation with Prof. G. Bastin and Dr. R. Mahony. However, the majority of work, approximately 90 %, is my own.

The following journal papers follow from the material presented in the thesis:

1. D. Nešić, "A note on dead-beat controllability of generalised Hammerstein systems", to appear in *Systems and Control Letters*.

2. D. Nešić and I. M. Y. Mareels, "Output dead beat control for a class of planar polynomial systems", submitted in 1995, first revision completed

3. D. Nešić and I. M. Y. Mareels, "Dead beat controllability of polynomial systems: symbolic computation approaches", submitted in 1995, first revision completed.

4. D. Nešić and I. M. Y. Mareels, "Dead beat control of polynomial scalar systems", in revision.

5. D. Nešić and I. M. Y. Mareels, "Dead beat control of simple Hammerstein systems", in revision.

6. D. Nešić and I. M. Y. Mareels, "State dead beat controllability of structured polynomial systems", in preparation.

7. D. Nešić and I. M. Y. Mareels, "Stability of implicit and explicit polynomial systems: symbolic computation approaches", in preparation.

A number of conference papers follows from the results presented in the thesis. Some of the material in these papers overlaps with that covered in the journal papers.

1. D. Nešić, I. M. Y. Mareels, R. Mahony and G. Bastin, " ν-step controllability of scalar polynomial systems", *Proc. 3rd ECC*, Rome, Italy, pp. 277-282, 1995.

2. D. Nešić, I. M. Y. Mareels, G. Bastin and R. Mahony, "Necessary and sufficient conditions for output dead beat controllability for a class of polynomial systems", *Proc. CDC*, New Orleans, pp. 7-13, 1995.

3. D. Nešić and I. M. Y. Mareels, "Invariant sets and output dead beat controllability for odd polynomial systems: the Gröbner basis method", *Proc. 13th IFAC World Congress*, San Francisco, vol. E, pp. 221-226, 1996.
4. D. Nešić, I. M. Y. Mareels, G. Bastin and R. Mahony, "Stability of implicitly defined polynomial dynamics: the scalar case", presented at MTNS, St. Louis, 1996.
5. D. Nešić and I. M. Y. Mareels, "Deciding dead beat controllability using QEPCAD", presented at MTNS, St. Louis, 1996.
6. D. Nešić and I. M. Y. Mareels, "Minimum time dead beat control of simple Hammerstein systems", presented at MTNS, St. Louis, 1996.
7. D. Nešić and I. M. Y. Mareels, "The definition of minimum phase discrete-time nonlinear systems revisited", to appear in Proc. *ICARV '96*, Singapore.
8. D. Nešić and I. M. Y. Mareels, "Scalar polynomial systems, triangular structures and dead-beat controllability", submitted in 1996.
9. D. Nešić and I. M. Y. Mareels, "Stability of high order implicit polynomial dynamics", submitted in 1996.
10. D. Nešić and I. M. Y. Mareels, "An output dead beat controllability test for a class of odd polynomial systems", submitted in 1996.
11. D. Nešić and I. M. Y. Mareels, "On some triangular structures and the state dead beat problem for polynomial systems", submitted in 1996.

Dragan Nešić

Department of Systems Engineering

Research School of Information Science and Engineering

The Australian National University

In Canberra, 29 August 1996

Acknowledgements

I would like to take this opportunity to thank my supervisor Prof. Iven M. Y. Mareels for his encouragement, support and for being such a nice person to work with. Iven has shown a remarkable patience with me in situations when I needed it most. Because of the war in my country, I undergone several personal crises in the last two years. During this time, my supervisors' help and support was invaluable in keeping my interest in research despite the national tragedy which happened in Yugoslavia. Iven's enthusiasm and excitement about new research ideas, his open mind to new and sometimes controversial concepts and his high standards in research made the time spent with him professionally very inspiring and fulfilling. His efforts to introduce me and my ideas to a number of people in the control community resulted in my visits to several conferences, universities and organisations, which was a great experience. Also, thanks to Iven I am one of the lucky students who got a chance to give several lectures on the subject of *Digitally Controlled Systems*. Having said all of this, I would like to thank Iven once again for exposing me to virtually every possible aspect of research life, from which I benefited a lot. Finally, I must admit that, apart from my research, the greatest challenge in the last two years was to beat Iven in table-tennis, which I managed just a few times.

I have cooperated with Prof. G. Bastin, Prof. H. Nijmeijer and Dr. R. Mahony and I would like to thank them for their patience and support. In particular I would like to convey my gratitude to Prof. G. Bastin whose support made possible my visit to Catholic University in Louvaine la Neuve. I express my sincerest gratitude to Prof. G. Bastin, Dr. R. Mahony and Dr. P. Bartlett for carefully reading parts of this manuscript.

I am indebted to Prof. G. E. Collins for his advice on QEPCAD, as well as for his effort in solving some problems that I had sent to him. The CRC for Robust and Adaptive Systems funded most of my visits to universities and attendance to workshops and conferences during my studies and I am grateful for that.

I would also like to thank my wife Nina and daughter Ksenija for their never ending love, support and understanding, without which this work would be much more difficult. The support and encouragement that I got from my family, relatives and friends was also very important to me and I thank them for this.

Last but not least, I would like to thank to all the students and staff for making such a stimulating and friendly atmosphere at the Department of Systems Engineering.

ABSTRACT

This thesis contributes to a better understanding of state and output dead-beat control problems and stability of zero output constrained dynamics for the class of discrete-time polynomial systems. Dead-beat controllability is one of the fundamental notions in control theory since it establishes the existence of control laws which can achieve a desired operating regime in finite time. The class of polynomial systems that we consider is very broad. Indeed, under very mild assumptions any nonlinear input-output map can be realised by a polynomial model.

Symbolic computation methods are exploited to tackle the dead-beat control problems. An algorithm for the design of minimum-time dead-beat controllers follows from our approach. In principle, the proposed method can deal with multi-input multi-output systems and bounds on controls and states can be included in a straightforward manner. The price we pay is the large computational cost, which prevent us from using this method in general.

To reduce the computational requirements for our controllability tests and design methodologies a number of simpler classes of polynomial systems are considered. Mathematical tools, such as algebraic geometry, real algebraic geometry, symbolic computation and convex analysis, are exploited. In this way, a number of analytic results are obtained with which we obtain computationally feasible controllability tests and design methodologies, as well as gain some more geometric insight.

Stability of zero output constrained dynamics and the related minimum phase property play an important role in output dead-beat control. The definitions found in the literature are not general enough to incorporate all behaviours that may occur in the context of polynomial systems. We revisit the definition of a minimum phase system and propose symbolic computation means to test different minimum phase properties for polynomial systems. Our results can be used for testing stability and stabilisability either by definition or by constructing Lyapunov functions.

Notation:

$\mathbb{R}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{C}$	The sets of real, natural, integer, rational and complex numbers.
$\mathbb{R}^n$	The set of all n-tuples (vectors) of real numbers.
$k[x_1, \ldots, x_n]$	Ring of polynomials with coefficients in the field k.
I	Ideal.
$\sqrt{I}$	Radical ideal.
$\langle f_1, \ldots, f_n \rangle$	Ideal generated by polynomials $f_1, \ldots, f_n$.
$V(f_1, \ldots, f_2)$	Variety of the polynomials $f_1, \ldots, f_n$.
$I(V)$	Ideal of a variety V.
$A \subseteq B,\ A \subset B$	A is a subset of B, A is a proper subset of B.
$A \cap B$	Intersection of sets A and B.
$A \cup B$	Union of sets A and B.
$A - B$	The set $\{x : x \in A, x \notin B\}$.
$\wedge$	Conjunction operator (and).
$\vee$	Disjunction operator (or).
$\exists$	Existential quantifier.
$\forall$	Universal quantifier.
$f \vert g$	$f, g \in \mathbb{R}[x_1, \ldots, x_n]$ and g divides f.
$\partial\mathcal{D}$	Boundary of the set $\mathcal{D} \subset \mathbb{R}^n$.
$\mathcal{D}^C$	Complement of the set $\mathcal{D} \subset \mathbb{R}^n$ with respect to $\mathbb{R}^n$.
$\mathring{\mathcal{D}}$	Interior of the set $\mathcal{D} \subset \mathbb{R}^n$.
card$\mathcal{D}$	Cardinal number of the set $\mathcal{D}$.
$S(x)\,,\ \hat{S}(x)$	Defining formulas for semi-algebraic sets S and $\hat{S}$
rank$A$	The rank of the matrix A.
A^T	Transpose of the matrix A.
dim$V$	The dimension of a variety V
im(f)	The image of the function f.
Gbasis$[f_1, \ldots, f_n]$	The reduced Gröbner basis for polynomials $f_1, \ldots, f_n$.
$x \succ y$	x is ranked higher than y using an ordering.

Abbreviations:

ARMAX	Auto-Regressive Moving Average with eXogenous input.
CAD	Cylindrical Algebraic Decomposition
I-O	Input-Output
MI	Multi-Input
MIMO	Multi-Input Multi-Output
NARMAX	Nonlinear Auto-Regressive Moving Average with eXogenous input.
PID	Proportional Integral Differential
PI	Proportional Integral
QE	Quantifier Elimination
QEPCAD	Quantifier Elimination by Partial Cylindrical Algebraic Decomposition
SISO	Single-Input Single-Output

Contents

Part I

Dead-Beat Controllability and Control of Polynomial Systems

Chapter 1

Introduction

The purpose of this chapter is in the first instance to emphasize the importance of the theory of discrete-time nonlinear systems. The main topic of the thesis, dead-beat control for polynomial discrete-time systems, is introduced and motivated. An overview of the existing literature dealing with dead-beat controllability is provided. The chapter is concluded with the outline of the thesis, highlighting the main contributions.

1.1 Nonlinear Discrete-Time Systems

In the last 40 years the control community has witnessed tremendous advances in computer technology which have had a great impact on the control systems theory and applications. Advances in hardware provided the control engineer with more powerful, reliable, faster and above all cheaper computers that could be implemented as process controllers. A good historical account of the genesis of digitally controlled systems is given in [6]. Today, almost all controllers are computer implemented. Consequently, the theory, which is used to design digital controllers and explain the phenomena that occur, is of utmost importance.

The usual configuration of computer controlled closed loop systems is given in Figure 1.1. The output of the process $y(t)$ is a continuous time signal. The measurements of the output signal are fed into an analog-to-digital (A-D) converter, where the continuous time signal is transformed into a digital signal - a sequence of measurements at sampling times t_k. If a measurement device is itself digital, the measurements are taken at sampling times only and there is no need for an A-D converter. The sequence of numbers $y(t_k)$ is used by the control algorithm in order to compute a sequence of controls $u(t_k)$ - the digital control signal. The sequence is converted into a continuous

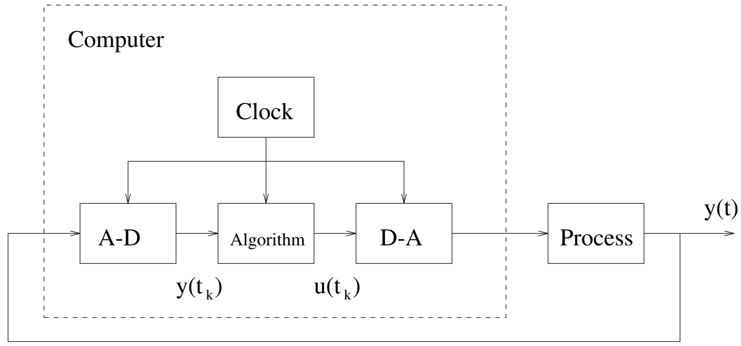

Figure 1.1: A computer controlled process

time signal by a digital-to-analog (D-A) converter. Between the sampling instants the system is in open loop mode. Consequently, the inter-sample behaviour is very often an issue and should not be disregarded. The system is synchronised by a real time clock in the computer.

One could develop a theory in a continuous time setting that takes account of the specific properties of the sampling process. This, however, may lead to undue complications. From an applications point of view, it is often sufficient to understand the system's behaviour at sampling instants only. The response between the sampling instants, being dictated by the open loop response of the system, can then be described in a secondary analysis to obtain a rather complete picture. This approach leads to a simpler analysis and although it neglects to a certain degree the interaction between the continuous time response and the digital control design process, it often suffices to come to a good engineering control design.

The above given approach gives rise to **discrete-time models**, which are used to model the properties of the system at sampling instants t_k. Discrete-time models are described by sets of *difference equations*, which play the same role in discrete-time as differential equations in continuous time.

Modelling of a sampled process given in Figure 1.1 is the main source of discrete-time models. These models may also arise from identification, where we identify a model of a sampled plant. This method of obtaining discrete-time models is also motivated by the prevalence of computer controlled systems. We also point out that a number of processes, such as economic and biologic systems, radars, internal combustion engines, etc. are inherently discrete in time [6]. Last but not least, difference equations arise when we approximate differential equations by an Euler or higher order approximation, such as Runge-Kutta.

In practise we have that *all plants and processes are nonlinear*. The most typical nonlinearity is a saturation. It is present in every system since it is never possible to deliver an infinite amount of energy to any real-world system. Since computer implemented controllers are today a standard configuration, a theory for *discrete-time nonlinear systems* may be of great importance in particular for control design purposes. Basing the controller design on a linearised model may not yield desired performance or even not be possible at all. Indeed, we can not use linear control theory in cases where: large dynamic range of process variables is possible, multiple operating points are required, the process is operating close to its limits, small actuators cause saturation, etc.

The advances in computer technology have provided the control engineer with tools to design and implement better controllers which perform well over a wide range of operating conditions. In order to achieve this, we can not use the traditional linear controllers. As a matter of fact, we normally have to resort to *nonlinear controllers*, which can be easily realised by means of a computer. A common solution to this problem is obtained by using switched linear controllers which are often used to control a plant around a set of different operating points. Yet another technique is to exploit adaptive controllers. A number of control paradigms have been proposed in the literature which modify linear control techniques to deal with a nonlinearity.

However, sometimes it seems more appropriate to start from a nonlinear model of a plant and design a nonlinear controller. Our understanding of nonlinear discrete-time systems is still very modest. The properties of nonlinear controllers are not easily investigated and capabilities understood. Hence, the theory of discrete-time nonlinear systems represents probably one of the most important challenges in control theory.

Because of the complexity of the general discrete-time nonlinear systems one needs to limit the scope of one's investigation in order to carry out an analysis successfully. Accordingly, the investigation of discrete-time nonlinear systems in this thesis is limited in three directions. That is, we consider:

- Class of systems: **discrete-time polynomial systems.** These systems are represented by polynomial input/output and/or state and output difference equations.

- Property: **dead-beat controllability.** Dead-beat controllability is a property of a system which guarantees that we can zero the state (or output) of the system in finite time for any set of initial conditions.

- Control laws: **dead-beat controllers.** Controllers which are such that they zero the state (or output) of the system in finite (or minimum) time starting from any initial state are called dead-beat.

Of course, we intend to be neither rigid nor dogmatic and on certain occasions other problems (we relax some of the above restrictions) are addressed. For instance, the problem of stability of zero dynamics and minimum phase polynomial systems are considered in Chapter 11 since they represent important issues in output dead-beat control.

1.2 Motivation

In this section we summarise the reasons that prompted us to investigate dead-beat controllability in the context of polynomial systems. First, we motivate the consideration of polynomial systems. Next, the importance of dead-beat controllability and dead-beat control is discussed. The minimum phase property is also addressed as an important issue in output dead-beat control. Finally, the available mathematical tools which provide further motivation for considering polynomial systems are discussed.

1.2.1 Polynomial Systems

Linear systems are not general enough to model all systems and processes of interest and very often one needs to resort to a nonlinear model. The trade off between the complexity of a model and its practical value for a design is an art in its own right, which very often depends on the engineer's experience and ingenuity. Hence, classes of models that are general enough to incorporate many plants and that still have "good" structure are invaluable in control theory.

One such class of nonlinear models is the class of discrete-time polynomial systems[1]. These systems are described by input-output

$$y(k+1) = F(y(k), \ldots, y(k-s+1), u(k), u(k-1), \ldots, u(k-t+1)) \tag{1.1}$$

and/or state and output equations,

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= h(x(k)) \end{aligned} \tag{1.2}$$

[1]Hereafter, discrete-time polynomial systems are referred to as polynomial systems.

where F, f and h are polynomials in all their arguments, and $y(k)$, $u(k)$ and $x(k)$ are respectively the output, input and state of the system at time k. The integers s and t in equation (1.1) determine the number of past inputs and outputs that influence the present output. Systems such as linear, bilinear and multi-linear are polynomial. Observe that systems given by (1.1) are a subclass of (1.2).

Discrete-time polynomial systems arise from:

- Modelling (from first principles).
- Identification (from collected data).
- Euler (or higher order) discretisation of continuous time polynomial systems (from first principles and an approximation).

Below we address each of these important sources separately. In Appendix A, we give several examples of polynomial systems, together with a list of applications of polynomial models, which illustrates the versatility of the processes that fall into this class.

Modelling

Polynomials have several important properties that give credit to their use in mathematical modelling for nonlinear dynamical systems. Any static continuous nonlinearity can be approximated with an arbitrary degree of accuracy over a compact domain using polynomials. Consequently, static nonlinearities are very often represented by polynomials. A very general result on good approximating properties of polynomials can be found in [43, Ch. 8] and is often referred to as the Stone-Weierstrass Theorem.

A direct consequence of the Stone-Weierstrass Theorem is that a very general class of discrete-time nonlinear dynamical systems can be approximated by a discrete-time polynomial system on a compact subset of the state space [60, 117]. Indeed, using the following definition:

Definition 1.1 *An input-output map is said to be continuous if, at time k, the output $y(k)$ depends continuously on the inputs $u_1(0), \ldots, u_n(0), \ldots, u_1(k-1), \ldots, u_n(k-1)$.* □

we recall the theorem [60, 117]

Theorem 1.1 *On a finite time interval, with bounded inputs in the discrete-time case, any continuous input-output map can be approximated by a polynomial (more precisely, state-affine) system (1.2) of finite state space dimension.* □

Hence, the class of polynomial systems is very general and, consequently, many of nonlinear phenomena occur in polynomial framework.

Furthermore, some polynomial nonlinearities arise from physical laws and the inherent features of the process that is modelled. For instance, multiplicative terms are often encountered in biochemical reactors [44]; the energy transmitted by radiation between two bodies depends on the difference of the fourth orders of temperatures of the bodies, etc. Notice that sampling usually destroys the polynomial structure of the continuous time system, except in special situations (for example, controlled sampling of bilinear systems [119]). However, discretisations of differential equations, which preserve polynomial structure, sometimes may serve as good approximate models of the sampled system.

Identification

An important feature of input-output polynomial models[2] is that they have a finite Volterra series representation (see [75, 76]), which can be used to identify the structure of a system. Identification techniques for block oriented models yield several important classes of polynomial NARMAX (nonlinear auto regressive moving average with exogenous inputs) systems (1.1). This is obviously a generalisation of ARMAX models for linear systems. A classification of these models is given in [75]. The best known classes of input-output polynomial models are: Wiener, Hammerstein, Wiener-Hammerstein, Uryson, Schetzen and their modifications (see Appendix A). Also, a subclass BARMA (bilinear auto regressive moving average) models were investigated in [119].

Polynomial and rational NARMAX models [184] of the following form were introduced more recently:

$$\begin{aligned} y(k+1) \;\; = \;\; & f(y(k), \ldots, y(k-s), u(k), u(k-1), \ldots, u(k-t), e(k), e(k-1), \ldots, \\ & e(k-l)) + e(k+1) \end{aligned}$$

where e is the disturbance to the system, which also takes into account modelling errors, and the nonlinearity f is a polynomial or rational function in all its arguments. By considering only the part of the system without the disturbance, we obtain polynomial or rational input-output

[2]The classification of polynomial systems with the definitions of some classes of systems that are frequently referred to in the thesis is given in Appendix A.

polynomial difference equations.

A very important result on good approximating properties of NARMAX models, which is similar to Theorem 1.1, can be found in [97, 98]. It was shown in these papers that under some mild assumptions, any controlled dynamic system can be described by a NARMAX model of the form (1.1). This result further motivates the use of polynomial models in control theory. Moreover, it indicates, together with Theorem 1.1, that there is a small loss of generality if instead of general nonlinear systems we consider only polynomial systems (1.1) and (1.2).

In [163] a comprehensive treatment of realisation theory for polynomial systems is presented. In particular, it was investigated when a system described by an explicit or implicit input-output difference equation can be realised by a polynomial model of the form (1.2). Moreover, it was shown in [163] that both polynomial and rational input-output maps (1.2) can be realised by finite dimensional state-affine systems of the form

$$x(k+1) = F(u(k))\,x(k) + g(u(k))$$

where F is an $n \times n$ matrix and g an n vector, whose entries are polynomials in the control vector $u(k)$. Notice that linear, homogeneous and inhomogeneous bilinear systems, simple and generalised Hammerstein systems are just special subclasses of state affine systems. Several practical examples from literature are treated in case studies sections and a long list of reported applications of polynomial systems is given in Appendix A.

Finally, the so called group method of data handling (GMDH) can be used to identify polynomial prediction models [76]. The obtained models usually consist of a number of polynomial subsystems which are interconnected in a special way. The underlying multi-layer structure is usually very complex but very complicated processes can be modelled in this way [76, 85, 175, 176, 177].

Discretisation

Our work is also motivated by the fact that continuous time polynomial systems have polynomial Euler or higher order discrete-time approximations. Although we are not dealing with the exact (sampled) discrete-time model, we may be able to modify dead-beat controllers for the approximation in order to obtain a kind of quasi dead-beat controllers for the original system. Notice that in general it is impossible to obtain a correct discrete-time model of the sampled nonlinear plant since we need to find analytic solutions to a set of ordinary nonlinear differential equations over one sampling period. However, Euler or Runge-Kutta like approximations may provide arbitrary accurate descriptions in discrete-time for the response of the sampled system and controllers based on the approximate models may indeed generate an excellent controlled response. In Chapter 10 we provide a simulation study of a biochemical reactor which shows that this approach may yield a well behaved closed loop system.

The question arises whether it is possible to obtain a systematic procedure for the design of controllers for sampled nonlinear systems, which are based on dead-beat controllers designed for Euler approximate models of the continuous time system. This approach is very often used and proved to be successful in adaptive control [69, 44]. Identifying the conditions and classes of systems for which this approach yields a well behaved closed loop system may offer new design strategies. The control laws obtained in this thesis can be regarded as a first step in this direction.

1.2.2 Dead-Beat Controllability and Control

Controllability is one of the fundamental notions in control theory. There are several different definitions of controllability which are exploited in the literature (see, for example, [81, 90, 167, 48, 56, 151]). We will deal with dead-beat controllability (also known as null controllability) [151].

Loosely speaking, the system is state (output) dead-beat controllable if it is possible to zero the state (output) of the system in finite time from any initial state. In other words, for any set of initial conditions it is possible to find a control sequence of finite length which renders the actual state (output) to be equal to the desired state (output). It is obvious from its definition that dead-beat controllability shows our ability to steer the system to a desired operating regime by means of the actuators. If the system is not controllable we can not always achieve (for certain initial conditions) the control objective. Thus, the physical set-up of the plant should be changed or bigger actuators installed, etc.

Controllability of a plant is necessary condition for a successful design of a controller and a starting point in a design is to check whether the plant is controllable or not. Hence, tests which check controllability are not only theoretically important but also are tools in the control engineer's tool-box. The dead-beat controllability test for linear systems is now a classical result in control theory. Results for dead-beat controllability of polynomial systems are, however, limited by necessity to special classes of polynomial systems (like linear systems).

The significance of the notion of controllability in linear control theory is obvious since many design related questions, such as arbitrary pole placement by state feedback, hinge on the controllability condition. For example, it should be noted that dead-beat controllability is very closely related to the existence of time-optimal control laws.

Note, however, that in a nonlinear context, controllability does not imply stabilisability and hence it does not play the same role for nonlinear systems from stabilisability point of view. Moreover, the definition of dead-beat controllability which we use is very closely related to the question of whether the origin of the system can be made globally attractive[3] by means of controls. Since stability and attractivity are two different notions, the difference of results in this thesis and stabilisability results is obvious. Nevertheless, controllability is still a very important concept in nonlinear control and is very closely related to realisation theory.

Kalman's elegant solution to the minimum-time dead-beat control problem for linear discrete-time systems has generated a large body of research in this area, which resulted in a number of important results and applications. The link between controllability and minimum-time control produced the minimum-time dead-beat controller for linear systems, which is sometimes a very good and easy-to-design option for the control designer. Dead-beat control is also a nice illustration that discrete-time systems offer new design possibilities - finite time settling via feedback - compared to continuous time systems (see [6], Examples 1.4 and 9.5).

It is important to emphasize that minimum-time dead-beat control is the best control law in certain situations. Additionally, even if we do not intend to implement time-optimal control we may gain a good understanding about the limitations to the system's performance if we investigate it. In this sense, dead-beat control represents a kind of a benchmark control law which tells us a lot about the intrinsic properties of the plant.

Dead-beat controllability is a desirable property for any control system to have. It appears to be very important to characterise the structure of classes of polynomial systems which have this property. This information can then be used, for instance, when choosing a class of models which are used to identify a plant. Indeed, one often has some flexibility over the choice of the class of models when identifying the model of a plant [75, 76]. It seems natural to choose the class that is more likely to have some good properties, such as dead-beat controllability. In this sense, our work is important from an identification point of view.

Dead-beat control sometimes requires large magnitudes of control which may lead to loss of

[3] Asymptotic attractivity is not considered since we require that any initial state is transferred to the origin in finite time.

robustness. This is the reason why this paradigm has "an undeservedly bad reputation" [6] in the control community. However, it may uncover limitations to performance for a given plant and can be used as a starting point in the design of a better controller [183]. In [66], Glad made the following remark: "The study of linear dead-beat controllers has given much insight into the properties of linear systems and it seems worthwhile to investigate output dead-beat controllers for nonlinear systems." Indeed, minimum-time control and controllability issues that Kalman solved gave us a much better understanding of capabilities and limitations of linear systems. This thesis is an attempt to contribute towards a better understanding of dead-beat controllability and control of discrete-time polynomial systems.

1.2.3 Minimum Phase Polynomial Systems

An important subproblem of the output dead-beat control problem is that of stability of zero output constrained dynamics (or zero dynamics) [123, 124, 130, 86]. The problem is practically very important since it is related to the question of boundedness of all process variables (states) while the output is kept constant [66, 67]. Systems which have stable zero dynamics are referred to as minimum phase. Note that there is a direct analogy with linear systems. These concepts are very important for some other control theoretic questions, such as input-output linearisation [130, 124, 86].

We emphasize that the concept of stable zero dynamics is directly related to the question of implementability of output dead-beat controllers. Indeed, it is not difficult to see that if we apply a minimum-time output dead-beat controller to a non-minimum phase linear plant, some of the states would grow unbounded while the output is kept at zero. Output dead-beat controllers, therefore, can be implemented only to minimum phase plants. Bearing this in mind, we can say that output dead-beat controllers are feasible only for minimum phase plants.

The notion of minimum phase systems in the nonlinear context has inherited from linear theory not only its name but also the definition which tries to mimic and capture the behaviour which is typical of linear systems. Moreover, it seems that the definition of minimum phase systems as usually found in the literature relies heavily on the methods which are used to investigate the property, but it can not be used in general. Some simple examples that we present in Chapter 11 illustrate our claims. This is the main motivation for considering minimum phase polynomial systems in Chapter 11.

1.2.4 On the Tools that are Used

An important source of motivation for consideration of polynomial systems is a plethora of results in algebraic geometry, real algebraic geometry and symbolic computation that we may exploit in tackling the dead-beat control problem. Polynomials are the most computable nonlinearities. They have a very nice algebraic structure. In algebraic geometry we have an elegant fusion of algebra and geometry which allows us to test algebraically whether a certain geometric condition (in the state space), such as controllability, occurs or not. For example, in Chapter 6 we use a decomposition of a polynomial into irreducible polynomials and a set of polynomial divisions in order to characterise output dead-beat controllability for a class of polynomial systems.

The advances in computer technology, which lead to much faster computers, as well as the emergence of new algorithms, software packages and tool-boxes, provide us with powerful new tools that can be used in control systems design. However, the pace at which this small revolution is taking place over the last 15 years seems to be too fast for the control community to make a best possible use of the incredible computational power and the emerging methods. In [27] some leading researchers in control community pointed out that one of the major challenges in control systems theory is harnessing the vast computational power, which today's computers offer.

One important feature of this thesis is a systematic use of symbolic computation packages, such as Maple, in the investigation of state/output dead-beat controllability and the controller design. In particular, the Gröbner basis method [37], cylindrical algebraic decomposition (CAD) and quantifier elimination (QE) [33, 35, 34], which were respectively discovered in 1960's and 1970's, are used to test dead-beat controllability and design dead-beat controllers.

The advances in the available tools change the way we think about control problems. Algorithmic tests and procedures have become an important way of solving problems. Also, the work of the computer science community has given us a new classification of problems based on computational complexity. It has become clear that we may not be able to compute some problems in a reasonable time with the available hardware. "The curse of computational complexity" warns us that irrespective of the incredible power of today's computers, we can not answer some relevant questions. Accordingly, the understanding of the computational complexity of control theoretic problems is an important feature of the problem itself (see for example [166]).

An understanding of the importance of computational complexity leads to a second important feature of my work. The problem that we consider is proved to be computationally very expensive for general polynomial systems. This strongly indicates that we need severe constraints on the structure of the polynomial system in order to obtain computationally feasible controllability tests and control laws. The results that are obtained show that a classification of polynomial systems according to the computational complexity of dead-beat problem is possible and perhaps more natural than the generally accepted one which uses the structure of the system (linear, bilinear, Wiener-Hammerstein, etc.).

An important consequence of the good approximation properties of polynomial systems is that they capture a large number of nonlinear, as well as all linear phenomena. Consequently, there are subclasses of polynomial systems which can be regarded as a transition between linear and nonlinear systems and for which tools from linear algebra can be successfully used in tackling the dead-beat controllability problem. For instance, a class of bilinear systems allows for controllability tests which are very simple to use and for which we only need linear tools [48, 49]. A large portion of this thesis is dedicated to one such class of polynomial systems, which are called Hammerstein systems. They represent one end of the large spectrum of polynomial systems and they often allow for a successful application of general symbolic computation algorithms because of their simple structure, which effectively reduces the computations.

The polynomial structure allows us to use a number of different tools in tackling the dead-beat problem. However, not all possibilities are explored in this thesis and the powerful methods of the differential geometric approach based on Lie algebras [90] and difference algebra [59], are not used although they may be more appropriate in some situations. We put more emphasis on constructive methods that allow us to solve the minimum-time problem at the same time. In this way, we lose some of the geometric insight but gain an explicit design methodology. It remains to be seen whether a fusion of some of the above mentioned methods into a more comprehensive methodology is possible or not.

In essence, a very important contribution of my work is that the dead-beat control problem is viewed from a constructive/computational perspective. I believe that this is a true engineering approach, made feasible by the immense computational power of computer hardware and the algorithm advances of real algebraic geometry.

1.3 Overview of the Literature

In this section we present an overview of the available results on linear and nonlinear dead-beat control. The survey paper [151] gives a good account of results on linear dead-beat control until 1981. This paper is cited for more classical results on linear dead-beat control and just a short description of the material presented therein is given. I concentrate more on the results that appeared in literature in the last 15 years and which are therefore not discussed in [151]. The overview is by no means comprehensive and it reflects the author's bias to papers addressing problems related to topics to be treated in the bulk of the thesis.

1.3.1 Linear Dead-Beat Control

Time-optimal control of discrete-time linear systems and the dual dead-beat state reconstruction problems have been investigated for more than five decades by many researchers and a number of interesting questions have been answered. The first textbook, where the dead-beat response of sampled data linear systems was noted, appears to be Oldenbourg and Stratorious' *Dynamics of automatic controls* published in Germany in 1944 [25]. This book was translated into English in 1948 and into Russian in 1949, and has helped to disseminate these ideas in both Eastern and Western countries [25]. The linear dead-beat problem has received a lot of attention since then and most of the questions associated with linear dead-beat control have been solved.

Roughly speaking, the minimum-time dead-beat control problem [151] is that of designing a controller which transfers any initial state (or output) of a system to the origin in minimum number of time steps, i.e. minimum-time[4]. Similarly, dead-beat state reconstruction implies the design of an observer that can reconstruct the state of a system in minimum-time. We are concerned here only with the dead-beat control problem; for a good overview of the dead-beat state reconstruction results see [151].

Two great impacts on linear dead-beat control that are discussed in [151] are:

1. Kalman's state space approach and controllability results.
2. Luenberger's canonical forms and arbitrary eigenvalue assignment under state feedback.

The state space approach for MI plants gives a number of possibilities to design dead-beat controllers and O'Reilly classifies them into: the Ludyk-Leden controller, the Kučera controller,

[4]The term "dead-beat" is used by O'Railley to describe minimum-time zeroing of state or output. Precise definitions that we use are given in Chapter 2.

the Tou-Farison-Fu controller and the Kalman controller. All of these methods use different ways of choosing linearly independent columns of the controllability matrix which are then used in the design procedure. The selection procedure is possible only in the MI case since for SISO controllable plants we have a unique minimum-time dead-beat controller.

It is a standard result in linear control theory that the eigenvalues of a closed loop system can be arbitrarily assigned if the system is controllable. The fact that dead-beat control is achieved when all poles of the closed loop system are placed at the origin yielded another set of methods for design. In [151] the following pole placement designs for MI plants are presented: the Ackermann-Prepelită controller, the Patcher-Ichikawa controller and the Fahmy-O'Reilly controller.

Besides the more classical problem of dead-beat control with full state feedback, a number of other related problems are discussed in [151]. The inaccessible state dead-beat problem has two solutions. The first approach is based on the design of a dead-beat observer which reconstructs the state of the system in minimum time and the dead-beat controller with full state information. Modularity of the observer - controller pair guarantees dead-beat behaviour of the closed loop system. The second method is based on the so called linear function observer which reconstructs the minimum-time dead-beat control law directly. In addition to this, output dead-beat control (minimum-time zeroing of output), dead-beat control of time varying systems, dead-beat control using output (non-minimum-time zeroing of state using linear static output feedback), static periodic output dead-beat control, dead-beat state reconstruction, etc. are referred to in [151]. The abundance of related problems indicate that the dead-beat control problem is one of the fundamental problems in control systems theory.

The third great impact on linear dead-beat control theory comes from transfer function factorisation approach and in particular the Youla parametrisation of all stabilising controllers. The parametrisation provided a systematic way of dealing with questions such as: robust dead-beat control and tracking, ripple-free dead-beat control and dead-beat control with smaller magnitudes of control signal. In addition to this, the use of two-degree-of-freedom controllers yielded results which are superior to one-degree-of-freedom controllers. We summarise below in more detail some of these results since they were discovered after [151] was published.

Zhao and Kimura [179, 180] used Youla parameterisation to design robust one-degree-of-freedom dead-beat controllers. It was shown that there is a trade-off between the settling (dead-beat) time and the robustness of the system; the greater the settling time, the better the robustness of the closed loop system with respect to the perturbation of the frequency response curve of the plant. The robustness index is in some sense an averaged sensitivity. They find the bound for improving the robustness by taking the limit of the optimal robustness index as the settling time goes to infinity; then they use it to determine the most appropriate settling time. The same authors used two-degree-of-freedom controllers in [181, 182] to show that better robustness properties of dead-beat controllers can be obtained and prove that no matter how long the settling time of one-degree-of-freedom controllers is, the robustness is always better if we use two-degree-of-freedom controllers with minimum settling time. In [64] robust dead-beat tracking was investigated using two-degree-of-freedom controllers.

By definition, dead-beat control implies finite time settling at sampling time instants whereas there might be an error between the desired and actual state (or output) between sampling instants; this phenomenon is termed “ripple”. The problem of ripple was dealt with in [149] and references therein. Nobuyama gave the parametrisation of all “ripple-free” dead-beat controllers based on the Youla parametrisation. It was shown in the same paper that, in a generic sense, minimum-time dead-beat controller causes ripple when the pulse transfer function of the plant has stable zeroes.

Probably one of the main hindrances to the implementation of dead-beat controllers is their property to produce very large values of control signals. This is natural to expect since we want to drive (if possible) every state of the system to the origin in the shortest possible time. It is proved in [183], however, that a trade-off between the settling time and values of control signals can be achieved. In this paper a transfer function factorisation approach was used in order to parametrise all stabilising two-degree-of-freedom dead-beat controllers using control input error which is defined as the difference between the control signal and its steady state value. The optimal control value is obtained by minimising the control input error in a quadratic sense with the specified settling time. It was shown then that there is a limit of the optimal control cost as the settling time goes to infinity and this was used to choose the most appropriate settling time. It is important to mention that although the paper deals with SISO systems, it is possible to extend the results to MIMO systems.

In addition to the three most prevalent approaches given above, there are a number of other results which use other methods or show connections between dead-beat control and other control paradigms. An interesting connection between minimum variance control and dead-beat control was established in [47]. It was proved that a suitable choice of weighting matrices (based on the Luenberger phase canonical form) in the cost function of the minimum variance control algorithm yields a dead-beat controller. In [84] a new approach was presented which is based on a state transition graph of a matrix and state and output dead-beat control problems were analysed. Kučera [111] solved a dead-beat servo problem using polynomial techniques; optimal dead-beat tracking control is obtained by solving two linear polynomial equations. The connection between state dead-beat control and the solution of the singular Riccati equation was first investigated in [94]; it was shown that the minimisation of a quadratic cost function which penalises only the terminal state leads to solving the singular Riccati equation, which yields a sequence of gain matrices that define a time variable dead-beat controller. It was shown that it is also possible under certain conditions to design a time invariant dead-beat controller. Extensions of results in [94] were given in [100] where it was proved that a time invariant dead-beat controller can always be found using the singular Riccati equation; the link between output dead-beat control and the singular Riccati equation was presented in [99].

1.3.2 Nonlinear Dead-Beat Control

The survey paper [151] gave an overview of about twenty years of research on linear dead-beat problem for linear systems, classified the available methods and gave a unified approach to the classical dead-beat problem. Unfortunately, any attempt to unify the available results for all nonlinear systems is bound to be futile since methods and classes of systems considered in the literature differ considerably. Classification is, however, still possible and it can be based on classes of systems considered or methods that are used. We present below an overview of results and methods on dead-beat control and controllability for nonlinear systems.

Polynomial Systems

We now discuss some results that address controllability of classes of polynomial systems in a manner very similar to ours. The underlying common idea is to define complete and dead-beat controllability in the same way as for linear systems [151] and investigate classes of systems (1.2). Consequently, this subsection is the most relevant for, and closely related to, my work.

A very important class of polynomial systems, whose controllability problem has been completely resolved, is the class of homogeneous bilinear systems of the form (for pioneering works see [70, 127]):

$$x(k+1) = (A + u(k)\,B)\,x(k)\,, \ \text{rank } B = 1 \tag{1.3}$$

where $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}$ are respectively the state and the control variables.

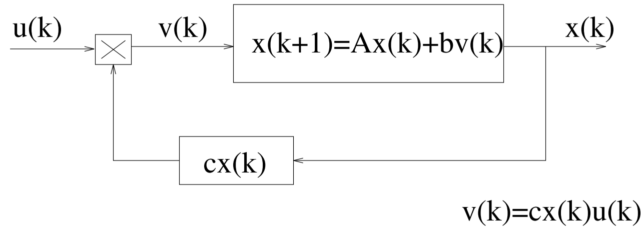

Figure 1.2: Decomposition of the bilinear system into a linear subsystem with multiplicative feedback.

Necessary and sufficient conditions for complete controllability on $\mathbb{R}^n - \{0\}$ for (1.3) are obtained in [48]. Notice that the system (1.3) can be decomposed into a linear subsystem and a multiplicative feedback [70]. The decomposed system is given in Figure 1.2 and its state equation can be rewritten as:

$$x(k+1) = (A + u(k)\,bc)\,x(k)\,,\ b \in \mathbb{R}^{n\times 1},\ c \in \mathbb{R}^{1\times n} \tag{1.4}$$

Some controllability conditions that are obtained in [48] differ considerably from the well known conditions for linear systems but still we only need linear algebra to test them. The structure of the system is very close to linear and this leads to an easy-to-check controllability test.

The solution of this problem has generated a series of results [49, 129, 82, 71, 64] which clarified some aspects of the problem itself or used the result to solve similar problems. Uncontrollable subspaces of (1.3) were investigated in [48, 82] and dead-beat controllability of the same class of systems was solved in [71]. In [64] it was shown that one of the conditions of the controllability test from [48] can be simplified. In [49] controllability of a class of inhomogeneous bilinear systems given by:

$$x(k+1) = (A + u(k)\,B)\,x(k) + du(k)\,,\ \mathrm{rank}(B:d) = 1 \tag{1.5}$$

was resolved. In addition to the very elegant solution and simple controllability tests, the above given papers explained in detail phenomena due to which we may lose controllability. For instance, in [70] it was noticed that the hyper-plane $H = \{x : cx = 0\}$ plays a crucial role for controllability of (1.3). On the hyperplane the system becomes insensitive to control. More importantly, the hyperplane H may contain an invariant subset, which is called in [70] a "free trajectory insensitive to control". If an initial state belongs to the invariant set, the trajectory always stays inside the hyper-plane H irrespective of the applied control. Necessary and sufficient conditions for the existence of the invariant set are given in the same paper.

It is clear that this phenomenon will occur in general polynomial systems and the existence of *invariant sets* is an important consideration in the investigation of certain controllability properties. In chapters 4, 5, 6 and 9 special attention is given to this.

Another class of polynomial systems whose controllability problem has been completely resolved is the class of SISO linear systems with positive controls [50]:

$$x(k+1) = Ax(k) + bu(k)\,, \quad u(k) \geq 0$$

If we introduce the transformation $u(k) = v^2(k)$ we have a class of simple Hammerstein systems. The controllability test is very easy to check and several important properties of this class of systems were observed. Observe that neither of the above given papers addressed the dead-beat controller *design* question.

Non-Polynomial Systems

Probably the first class of nonlinear systems for which the dead-beat control problem was addressed and solved is linear systems with bounded controls ($|u| \leq 1$) [39, 40, 174]. In [174] MIMO systems were considered and the time-optimal control algorithm was derived. The method is based on the construction of sets of initial states from which the origin can be reached in the first, second, etc. steps. Using this construction, a critical hyper-surface (critical line in the case of a second order system), which is crucial in the optimal control policy, is found. The distance between the critical hypersurface and the initial state is measured in an appropriate direction and an appropriate value of control signal is then determined. In [39] the critical line is proved to tend to the switching line of the continuous time-optimal system (bang-bang control) when the sampling period tends to zero. It should be emphasized that my work is going along the same lines. Indeed, the work in this thesis to some extent revisits these ideas that appeared in the literature in 1960's but more recent mathematical tools are used.

A number of generalisations of the above result (just controllability existence) were reported in a series of papers by Evans [52, 55, 56]. The most general MIMO situation is considered when the control signals belong to convex sets. The special cases of this class of systems are linear systems with bounded controls [174] and linear systems with positive controls [50]. However, no design for a dead-beat controller has been reported.

In [14] some interesting examples of output dead-beat control for scalar nonlinear systems were analysed. It was shown that there may be many different control laws that keep the output at zero and the criterion of choice is crucial for dead-beat control of nonlinear systems. In other words, output dead-beat control requires control to a target set on which the output is zero. The dynamics that are constrained to the target set may be realised in general using a large number of different control laws, which may have very different behaviours. The approach taken in [14] is based on predictive control, whose special case is dead-beat control. Note that in predictive control framework usually non-minimum-time dead-beat control is considered.

In [12] long range predictive control for nonlinear systems given by Volterra series is addressed and suboptimal controllers are proposed. This approach seems to be very promising for this class of nonlinear systems. However, a number of questions, such as the effect of changing working points, the sampling time, the output disturbances, etc. need to be addressed in future research. A number of references on the predictive approach to dead-beat control (of polynomial and non-polynomial nonlinear systems) can be found in [78].

In his papers [66, 67] on output dead-beat control, Glad considered the following class of sampled data nonlinear systems:

$$\begin{aligned} \dot{x}(t) &= a(x(t)) + ub(x(t)) \\ y(t) &= c(x(t)) \end{aligned} \tag{1.6}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$, $y \in \mathbb{R}$ and the control u is constant over the time intervals $[0, T)$, $[T, 2T)$, $\ldots$. In [66] systems of the form (1.6) that have one zero at infinity were analysed; in other words, the relative degree of the system is r=1. An extension to systems of an arbitrary relative degree r was presented in [67]. Glad proved that if the system (1.6) is minimum phase, i.e. its zero dynamics are stable, then there exists stabilising dead-beat control which zeroes the output of the system in minimum number of steps which is equal to the relative degree of the system, provided that the sampling period T is sufficiently small. He also proposed a controller which uses the Newton method for computing the value of the control signal, which is a root of a nonlinear algebraic equation. It is important to note the underpinning idea of his method; it is known that the continuous time system (1.6) that has the relative degree r can be input-output linearised [86, 130] using a change of coordinates and an appropriate feedback so that the resulting system consists of linear and nonlinear parts. The input-output relationship can then be described by a transfer function $1/s^r$. The design of a dead-beat controller for linear systems is a trivial problem, but the question arises whether the linearisability is preserved if instead of a continuous we use the sampled control signal. Glad proved that for small sampling periods T it is possible to preserve some of the input-output structure in the sense that the input affects the output through a series of r integrators whereas the actual input-output relationship can be nonlinear. Consequently, in [66] one step dead-beat control was considered and in [67] zeroing of the output in minimum r time steps. In the latter paper, he also shown that the control law (at sampling instants) must be very close to the control that input-output linearises the continuous time system.

Some results on nonlinear non-minimum-time dead-beat control come from the area of moving horizon control [103]. This approach can give control laws which are dead-beat in the sense that all the initial states are transferred to the origin in a finite number of time steps. However, it does not give the solution to the minimum-time problem. This could be expected, since moving horizon implies minimising of the following cost function $J = \sum_{k=0}^{N} l(x(k), u(k))$, where $l(x, u) \geq 0, \forall x, u$ and N is finite and fixed. Hence, the moving horizon cost function penalizes the values of control signals and state variables. On the other hand, state dead-beat control implies a minimisation of a quadratic cost function which penalizes only the terminal state $J = x^T(N)x(N)$ (see [94]). N is not a fixed integer and minimisation of this cost is done over all possible values of N.

A very good overview of accessibility properties of invertible discrete-time nonlinear systems can be found in [163]; a Lie algebraic approach was used to solve different accessibility properties of invertible discrete-time nonlinear systems. The notions of forward and backward accessibility and transitivity that are investigated in [59, 90, 7, 8] differ from the more classical definition of dead-beat controllability that we consider. The papers [90, 7, 8] investigate in particular the situations when the forward accessible (reachable) set from $x(0)$ has a nonempty interior in the orbit of $x(0)$ (this is often referred to as "positive form of Chow's Lemma"). The positive form of Chow's Lemma was proved for the following situations: when the system is smooth, the initial state is an equilibrium and the control value set is connected [90], if the map f in $x(k+1) = f(x(k), u(k))$ is rational, when the state space is compact [7], under a Poisson stability condition [7] and in a generic sense [7].

The reason why orbits (forward-backward accessible sets) are investigated is that they are mathematically much easier objects to deal with, they have the structure of smooth manifolds and they partition the state space into invariant submanifolds that integrate a natural distribution of vector fields. Although orbits have much nicer structure than forward or backward accessible sets, we are usually interested in the latter. Lie algebraic techniques give us a powerful tool but not all questions of interest can be solved using these methods. It is interesting that Lie algebraic methods are less powerful in discrete than in continuous time since it can be shown [90] that the continuous time situation is just a very special case of discrete-time systems. The main difficulty when applying Lie techniques in discrete-time is that instead of algebraic groups, semi-groups appear which leads to loss of algebraic structure.

It should be recognised that the dead-beat problem is closely related to the invertibility of the system. A number of contributions on the invertibility of discrete-time nonlinear systems can be found in [57, 108] and references therein.

1.3.3 Implementations: pro et contra

Finally, it is worth discussing some reported applications of dead-beat control; we refer only to two papers which illustrate pros and cons for the implementation of dead-beat control. Both papers investigate linear dead-beat control. We are not aware of any reported applications of nonlinear dead-beat control.

In [156] dead-beat control was proved to be inappropriate for the situations when the plant dynamics are unknown and had to be obtained via a learning algorithm. The authors use a non-parametric statistical technique termed locally weighted regression, or memory based learning, to learn the model of a plant. Learning is done in closed loop and hence the controller plays a prominent role in the efficiency of this method. The authors described dead-beat controllers as "too aggressive" for the tasks where the model is to be learned using their method since it tries to cancel the plant dynamics entirely, which leads to "an unpredictable, and most often unfavourable behaviour".

On the other hand, in [58] a study of simulation results and their verification on a real hydro power plant was presented for three different control paradigms; besides a dead-beat controller[5], PI and PID controllers with adaptive parameters and a robust controller were designed and their performance compared. First, using identification techniques, very good linearised reduced-order models of the hydro power plant, which are valid around certain operating points, were obtained. The original mathematical model was described by 24 nonlinear differential and several algebraic equations and therefore too complex to apply some of the design techniques. Second,

[5]Dead-beat controller differs a bit from the usual design since the plant is not minimum phase [58].

the above mentioned controllers were designed and their performance was checked by simulations on the computer. Simulations showed that although all three controllers were good, the dead-beat controller yielded a slightly better performance. Finally, the designed controllers were implemented on the real hydro power plant and it was shown that simulations were in a good agreement with experiments, proving that dead-beat control performed better. More surprisingly, it was shown that: " . . . the parameters of the (dead-beat) controller were almost insensitive to changes of load and pressure head. Only the gain had to be adapted to the respective operating point".

These two contradictory results show that one should be careful when implementing a dead-beat controller to plants whose models can not be determined correctly. However, if the plant model is well known, dead-beat control may outperform other control paradigms.

1.4 Outline of the Thesis

We present below the outline of the thesis, summarise our contributions in each chapter and discuss a chart which illustrates the logical dependence of the chapters.

Chapter 2: The main concepts and notation are defined. Some general assumptions are listed and commented on. We use the assumptions in the remainder of the thesis unless otherwise stated. A minimum-time dead-beat controller is designed for a second order linear system and its properties are analysed. Then, several nonlinear dead-beat control examples are considered in order to show what phenomena we may face in the nonlinear context.

Chapter 3: Conditions for state and output dead-beat controllability for a very large class of polynomial systems is given. Polynomial systems with **rational coefficients** are considered:

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

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}$ and $u \in \mathbb{R}$ are state, output and control respectively. We have $f{=}(f_1\ f_2\ \ldots\ f_n)^T$ and f_i, h are polynomials in all their arguments with rational coefficients. The class of systems is very important from applications point of view since it is not difficult to approximate irrational coefficients with rational ones.

The solution hinges on the cylindrical algebraic decomposition algorithm which is implemented using the QEPCAD symbolic computation package. The solution is constructive and it naturally leads to the design of a family of minimum-time state/output dead-beat controllers. The main difficulty with the approach is that it is computationally very demanding. Another hindrance is the occurrence of a non-terminating procedure in the controllability test, which may occur for some systems. This phenomenon is illustrated by Example 2.3 in Chapter 2 and it can be regarded as an intrinsic property of a class of polynomial systems. It is not possible in general to say *a priory* whether the procedure terminates or not. We specify the number of steps that we want to check and if an answer is not obtained within the specified horizon, we can either increase the horizon or stop the procedure. The work in this chapter appears to be completely new and we are not aware of any previous results that either address the problem in its full generality or use the approach that we take.

Although the method based on QEPCAD can in principle be used for any polynomial system (1.7), the computations are very often not possible due to the underlying computational complexity. Moreover, a negative aspect of using QEPCAD for the dead-beat problem is that we may loose some of the geometric and structural insight. Hence, it appears to be very important to identify classes of polynomial systems for which QEPCAD can efficiently be used, analytic dead-beat controllability tests derived or some structural properties uncovered. Consequently, in the remaining chapters we always revisit the same idea of constraining the structure of general polynomial systems (1.7) in order to obtain easier-to-check controllability tests and/or identify systems' characteristics that determine dead-beat controllability.

Chapter 4: Necessary and sufficient conditions for output dead-beat controllability of a class of odd polynomial systems are presented. The controllability test based on QEPCAD (Chapter 3) may fail to provide an answer to the controllability question due to a non-uniform bound on the dead-beat time or computational complexity of the considered system. We constrain the structure of general polynomial systems and then we use another algorithm which computes invariant sets (similar to [70]) of some critical sets in state space. It is shown that the so called maximal invariant set can be computed using an algorithm that stops in finite time. The core of the algorithm is the Gröbner basis approach, which is computationally less expensive than QEPCAD. Although in general we need to resort to the use of QEPCAD, we may sometimes conclude on dead-beat controllability without QEPCAD. This is the first class of systems where a trade-off between the computations and the constraints on the structure naturally arises. Finally, output dead-beat controllability of a column type grain dryer is investigated using the Gröbner basis method.

Chapter 5: Necessary and sufficient conditions for dead-beat controllability of scalar discrete-time polynomial systems are derived. The considered class of systems is described by:

$$x(k+1) = f(x(k), u(k))$$

where x and u are scalars. An algorithm that can decide the dead-beat controllability properties of scalar discrete-time polynomial systems, in the *generic case*, is presented. The non-genericity condition under which the algorithm fails to provide a conclusion is identified. A minimum-time dead-beat controller is designed for the generic class of dead-beat controllable scalar polynomial systems. Also, sufficient conditions for the existence of local and global stabilising dead-beat control laws for unbounded control signals for the same class of systems are presented. Furthermore, sufficient conditions for the existence of locally stabilising dead-beat control with bounded control signals are obtained. A number of interesting phenomena are illustrated by examples. We then investigate dead-beat controllability of a heat exchanger.

Chapter 6: Output dead-beat control for a class of nonlinear discrete-time systems, which are described by a single input-output polynomial difference equation, is considered. The class of systems is given by:

$$y(k+1) = f(y(k), u(k-1), \ldots, u(k-s), u(k))$$

and the highest exponent in $u(k)$ is an odd integer. Necessary and sufficient conditions for the existence of output dead-beat control are obtained. We propose two different output dead-beat controllability tests. It is important to emphasize that the controllability tests *can be checked in finitely many rational operations*, which was not true for any of the classes of systems considered in Chapters 3 and 4. The Gröbner basis method and QEPCAD are used to test output dead-beat controllability of this class of systems.

Chapter 7: Dead-beat controllers for simple Hammerstein systems are investigated. The class of simple Hammerstein systems is given by:

$$\begin{aligned} x(k+1) &= Ax(k) + bf(u(k)) \\ y(k) &= cx(k) + df(u(k)) \end{aligned}$$

where $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}$ are respectively the state and the input of the system at time k.

Also, it is assumed that $f(u) = \pm u^m + c_{m-1}u^{m-1} + \ldots + c_0$ and $m > 0$ is an even integer. Several designs for non-minimum-time state dead-beat controllers are given for certain classes of simple Hammerstein systems. A general minimum-time state dead-beat controller is presented for a class of simple Hammerstein systems. Actually, a family of minimum-time dead-beat control laws is provided. This enables us to shape of transient response via choosing an appropriate control law. Finally, we design an output dead-beat controller for a class of Hammerstein systems that are not necessarily state dead-beat controllable.

Chapter 8: Necessary and sufficient conditions for dead-beat and complete controllability for a class of generalised Hammerstein systems are presented. We consider generalised Hammerstein systems of the form [75, 76]:

$$\begin{aligned} \begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} &= \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \begin{pmatrix} g_1 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ g_2 \end{pmatrix} u^2(k) \\ y(k) &= (c_1^T \; c_2^T) \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + d_0 + d_1 u(k) + d_2 u^2(k) \end{aligned}$$

Since the system's structure is very close to linear, only linear algebra is used for the controllability test. The test is very simple and easy to use.

Chapter 9: Several interconnected polynomial systems are considered and their dead-beat properties investigated. Interconnected systems seem to be very important from a practical point of view since many systems fall into this category and yet dead-beat controllability tests may be very simplified. Using some triangular structures of this chapter QEPCAD based dead-beat controllability tests of Chapter 3 become much more efficient and therefore practically important. Moreover, we give a variety of block oriented models for which it is even possible to obtain analytic dead-beat controllability tests by using some know dead-beat controllability tests. Minimum-time dead-beat controllers can also be easily designed for classes of structured systems and we present one such design. Results of this chapter show that sometimes it is possible to creatively and flexibly use the results of this thesis when dealing with the dead-beat problem for structured polynomial systems.

Chapter 10: A simulation study for a biochemical reactor is presented. The continuous time system (biochemical reactor) is sampled using a sampler and zero order hold. A discrete-time model of the plant is obtained by using the Euler approximation. Then, we design a minimum-time dead-beat controller for the approximation. The dead-beat controller is modified to meet the physical limitations (actuator saturations) and applied to the sampled plant. Simulations reveal a very well behaved closed loop system. This study is aimed at motivating the use of dead-beat controllers proposed in the thesis. Moreover, a more in depth study of implementation issues for minimum-time dead-beat controllers that we obtained seems to be appropriate.

Chapter 11: Minimum phase polynomial systems are considered. First, the known definition of minimum phase systems is shown to be inadequate for general polynomial systems. Several new definitions are proposed and it is shown how QEPCAD can be used to check these properties. We show that stability properties of implicit and explicit polynomial systems can be checked using QEPCAD in a rather unexpected way: by definition. More surprisingly, this method is computationally less expensive than “computing” a Lyapunov function in certain situations. The results of this chapter shed completely new light on the stability problem. Finally, explicit conditions for stability of scalar implicit polynomial dynamics are presented. A case study of a radiator and fan system is also presented.

Chapter 12: A summary of the obtained results and some directions for further research are presented.

Appendix A: Additional practical examples of polynomial models are presented and a list of applications of polynomial systems, together with relevant references are given.

Appendix B: Results from algebraic and semi-algebraic geometry and computer algebra systems, which we use, are presented. The Gröbner basis method, CAD and QE are discussed in more detail. Also, some definitions and notation is defined in this appendix. If unfamiliar with this material one should read Appendix B before reading the rest of the thesis.

We emphasize that a number of examples are presented in all chapters, since they best illustrate the phenomena and behaviours that may occur in this context. We hope that the versatility and richness of the illustrated behaviours would give to the reader a better feeling for the area and a deeper understanding of the presented results.

There are several ways in which the thesis can be read. The logical dependence of the chapters is shown in Figure 1.3. Chapters 3-11 represent the core of the thesis. Having read the first two chapters, one can proceed on to Chapters 3, 7 or 11. If one prefers first to read about the general framework and then about specialisations of the developed theory, Chapters 3 and 4 should be read first. On the other hand, if the approach from simple to general is preferred, the best way is to read Chapter 5. Chapters 3, 4 and 5 should be read before Chapter 9 since a number of results from the former are used in the latter. An interested reader may refer to Appendix A for a long list of applications of polynomial systems which we compiled. Also, the reader not familiar with the Gröbner basis method, quantifier elimination (QE) and cylindrical algebraic decomposition (CAD) should first read Appendix B.

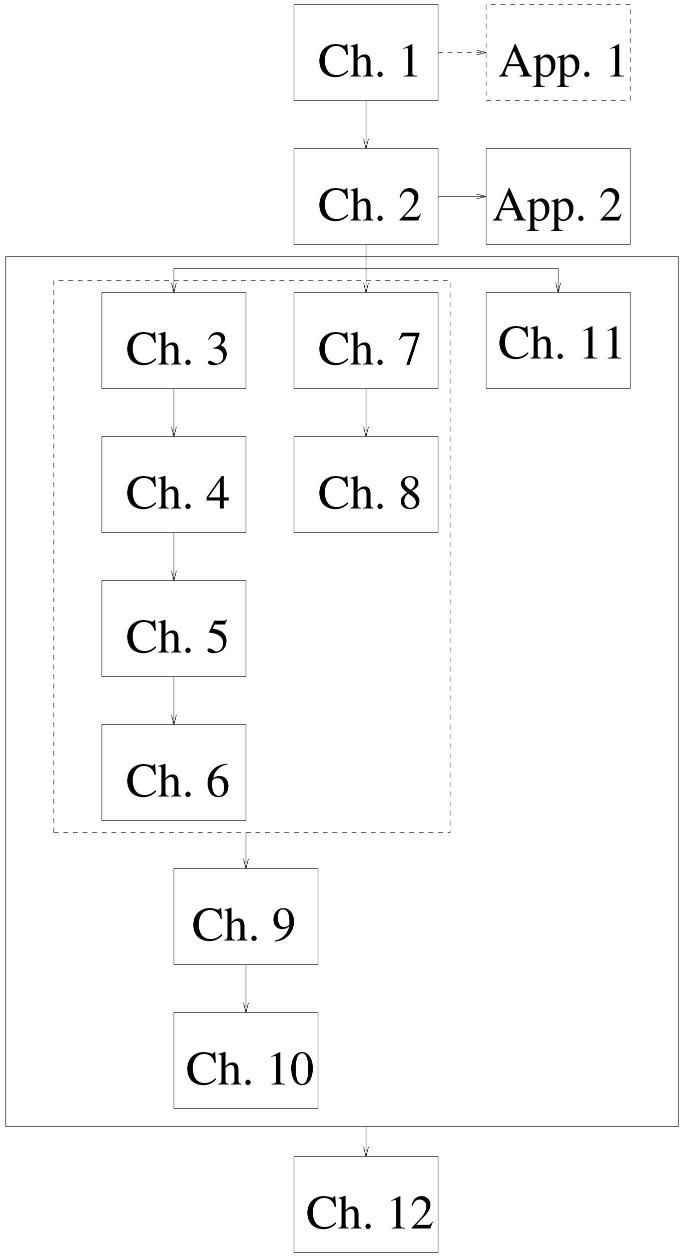

Figure 1.3: Logical dependence of the chapters.

Chapter 2

Preliminaries

The purpose of this chapter is to introduce some general notions, present some notation and main definitions and discuss some of the assumptions that are used throughout most of the thesis. The more mathematical results that we borrow from algebraic geometry and symbolic computation are collected in Appendix B. If the reader is not familiar with the Gröbner bases, cylindrical algebraic decomposition (CAD) and quantifier elimination (QE), it is necessary to read Appendix B before reading Chapters 3, 4 and 6. Some notation and definitions from algebraic and semi-algebraic geometry are listed in Appendix B.

2.1 Notation and Definitions

We use the standard definitions of rings and fields [37, 87]. The ring of polynomials in n variables over a field k is denoted as $k[x_1, x_2, \ldots, x_n]$. Let $f, g \in k[x_1, x_2, \ldots, x_n]$. $f|g$ means that f is divisible by g, that is, there exists a polynomial $h \in k[x_1, x_2, \ldots, x_n]$ such that $f{=}hg$. $f \equiv g|h$ means that f is divisible by h modulo g, that is, given polynomials h and g, multideg$(g) <$ multideg(f) there exists a polynomial $h_1 \in k[x_1, x_2, \ldots, x_n]$ such that $f{=}h_1 h + g$ (for a definition of the multi-degree of a polynomial see Appendix B). Also, $f \not| g$ and $f \equiv g \not| h$ denotes respectively that f is not divisible by g and f is not divisible by h modulo g.

All the systems that are considered in the sequel are subclasses of the following class of polynomial systems:

$$\begin{aligned} x(k+1) &= f(x(k)\,, u(k)\,) \\ y(k) &= h(x(k)\,) \end{aligned} \tag{2.1}$$

where $x(k) \in \mathbb{R}^n,\ y(k) \in \mathbb{R}$ and $u(k) \in \mathbb{R}$ are respectively the state, the output and the input of the system (2.1) at time k. The vector $f(x,u) = (f_1(x,u) \ldots f_n(x,u))^T$ is such that $f_i(x,u) \in \mathbb{R}[x,u] = \mathbb{R}[x_1, x_2, \ldots, x_n, u]$ and $h \in \mathbb{R}[x_1, \ldots, x_n]$.

In [163], it was proved that the systems that satisfy the following input-output polynomial difference equation

$$y(k+1) = F(y(k), \ldots, y(k-s), u(k), \ldots, u(k-t)), \tag{2.2}$$

where $F(y_1, \ldots, y_{s+1}, u_1, \ldots, u_{t+1}) \in \mathbb{R}[y_1, \ldots, y_{s+1}, u_1, \ldots, u_{t+1}]$, allow finite dimensional realisations of the form (2.1). Systems (2.1) are, however, more general than (2.2) [163].

A sequence of controls is denoted as $\mathcal{U} = \{u(0), u(1), \ldots\}$. The truncation of $\mathcal{U}$ to a sequence of length $p+1$ is denoted as $\mathcal{U}_p = \{u(0), u(1), \ldots, u(p)\}$.

The composition of a function $g(x)$ with itself is denoted as

$$g^p(x) = \underbrace{g \circ g \circ \ldots \circ g}_{p+1 \; times}(x)$$

If we have a control action at our disposal, we denote the composition of function f as:

$$f_{u(1)} \circ f_{u(0)}(x(0)) = f(f(x(0), u(0)), u(1)).$$

For longer sequences of controls $\mathcal{U}_p$ we use the notation

$$f^{\mathcal{U}_p}(x(0)) = \underbrace{f(f(\ldots f(f}_{p \; times}(x(0), u(0)), u(1)), \ldots, u(p))$$

The state of system (2.1) that is reached from the initial state $x(0)$ at time step $p+1$ under the action of a control sequence $\mathcal{U}_p$ is denoted as $x(p+1, x(0), \mathcal{U}_p)$. Hence, we can write:

$$x(p+1, x(0), \mathcal{U}_p) = f^{\mathcal{U}_p}(x(0))$$

The following sets are introduced:

$$\begin{aligned} S_0 &= \{x : \exists u \in \mathbb{R} \text{ such that } f(x,u) = 0\} \\ S_k &= \{x : \exists u(0), \ldots, u(k) \in \mathbb{R} \text{ such that } f_{u(k)} \circ \ldots \circ f_{u(0)}(x) = 0\} \end{aligned} \tag{2.3}$$

The set S_k consist of all states in the state space with the following property: the minimum time necessary to transfer any $x(0) \in S_k$ to the origin is *at most* $k + 1$ time steps.

Notice that $S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots$.

We also use the following sets:

$$\begin{aligned} \hat{S}_0 &= \{x : \exists u \in \mathbb{R} \text{ such that } f(x,u) = 0\} \\ \hat{S}_k &= \{x : \exists u \in \mathbb{R} \text{ such that } f(x,u) \in S_{k-1}\} - S_{k-1} \end{aligned} \tag{2.4}$$

The set $\hat{S}_k$ consist of all states in the state space with the property: the minimum time necessary to transfer any $x(0) \in \hat{S}_k$ to the origin is *exactly* $k + 1$ time steps. The following relations are easily verified:

$$\begin{aligned} S_0 &= \hat{S}_0 \\ S_k &= \cup_{i=0}^{i=k} \hat{S}_i, \ \forall k \in \mathbb{N} \\ \hat{S}_k &= S_k - S_{k-1}, \ \forall k \in \mathbb{N} \end{aligned} \tag{2.5}$$

We now give a list of definitions that are used in the sequel.

Definition 2.1 The system (2.1) is state dead-beat controllable if for any initial state $x(0) \in \mathbb{R}^n$ there exists a control sequence $\mathcal{U}$ and $\nu \in \mathbb{N}$ such that $x(p+1, x(0), \mathcal{U}_p) = 0, \ \forall p \geq \nu$. □

Definition 2.2 The system (2.1) is output dead-beat controllable if for any initial state $x(0) \in \mathbb{R}^n$ there exists a control sequence $\mathcal{U}$ and $\nu \in \mathbb{N}$ such that $h(x(p+1, x(0), \mathcal{U}_p)) = 0, \ \forall p \geq \nu$. □

Definition 2.3 The system (2.1) is completely controllable if for any pair of states $x(0), \ x^* \in \mathbb{R}^n$ there exists a control sequence $\mathcal{U}_p$ such that $x(p+1, x(0), \mathcal{U}_p) = x^*$. □

Notice that in Definitions 2.1 and 2.2 we require that we can keep the state/output at zero, once we have zeroed it. However, in Definition 2.3 we do not require that we can stay at x^* once we have reached it.

Definition 2.4 A state dead-beat control law is a mapping $g : \mathbb{R}^n \to \mathbb{R}$ denoted as $u = g(x)$ such that for $r(x) = f(x, g(x))$ there exists an integer ν such that $r^p(x) = 0, \ \forall p \geq \nu, \ \forall x \in \mathbb{R}^n$. □

Definition 2.5 An output dead-beat control law is a mapping $g : \mathbb{R}^n \to \mathbb{R}$ denoted as $u = g(x)$ such that for $r(x) = f(x, g(x))$ there exists an integer ν such that $h(r^p(x)) = 0, \ \forall p \geq \nu, \ \forall x \in \mathbb{R}^n$. □

In general we have $\nu = \nu(x(0), \mathcal{U})$ (see Definition 2.1). We denote

$$T = \max_{x \in \mathbb{R}^n} \min_{\mathcal{U}} (\nu(x, \mathcal{U})) .$$

$T \in \mathbb{N}$ and it is either a fixed positive integer or $+\infty$. If $T \neq +\infty$, we say that *there exists a uniform bound on the dead-beat time.*

In the sequel we investigate dead-beat controllability with a uniform bound on the dead-beat time unless otherwise stated.

Also, in Definition 2.5 we have that $\nu = \nu(x, g)$. If the mapping g is such that for all $x \in \mathbb{R}^n$ it yields minimum value for ν, we say that g is a minimum-time state/output dead-beat controller. On the other hand, if there exists $x^* \in \mathbb{R}^n$ such that g does not yield minimum $\nu(x^*, g)$ in Definition 2.5, we have a non-minimum-time dead-beat controller.

Definition 2.6 A set S_I is (control) invariant if $\forall x(0) \in S_I, \forall \mathcal{U}_p, \forall p = 0, 1, 2, \ldots$, we have that $x(p+1, x(0), \mathcal{U}_p) \in S_I$. □

Invariant sets of different subsets of state space play a major role in our developments.

2.2 *General Assumptions*

The purpose of this section is to discuss the general assumptions that are used and which are not commented on in the sequel. These assumptions are used unless otherwise stated. They may be viewed as a trade-off between the knowledge on the problem that had existed in the literature before and the goals that were set in the beginning. All of these assumptions are motivated by the complexity of the dead-beat problem. It was our intention to understand this artificial problem first and then try to relax some of these assumptions in order to consider the situations that are more practically important. It should be emphasized that the assumptions are the same as the ones that the pioneers of linear dead-beat control used 35 years ago to solve the questions of controllability and minimum-time control. It took a few decades to relax some of these assumptions in the linear context. This is enough motivation for us to use these assumptions in the nonlinear context.

G1. *The model of the plant is known and valid for all operating conditions.*

There are two important problems that we are overlooking here: the structure of the model may vary with different operating conditions and the values of physical coefficients are never known exactly. In other words, we assume that there are no structural and parametric uncertainties. The given model captures all modes of operation with sufficient accuracy. The main reason for this assumption is the simplification of the problem. Although in the real world there are no "exact models", our inability to crack the complexity of the general problem forces us to consider the simpler artificial problem. This aspect of my work does not differ much from the classical approach in control literature.

G2. *The control signal can take on any real value.*

In this way we are not considering the most typical of all nonlinearities - saturation. Most of the results that are obtained can be viewed as necessary for controllability with bounded controls. We have already mentioned that one reason for consideration of polynomial systems is that there exist powerful tools in mathematics that can be used for their analysis. Saturation, however, destroys the polynomial structure and as a result we obtain a class of systems that is highly nonlinear and for which there is much less tools available. It is important to say that the control laws that we present are also designed under Assumption G2 and they can be regarded as a first step in the design of a feasible controller (with bounded controls). We note that the methods of Chapter 3 can incorporate bounded controls. However, the computational requirements for the proposed methods are usually hindering in general.

G3. *We can ignore the inter-sample behaviour.*

In other words, we do not consider what happens between sampling instants, assuming that this behaviour is satisfactory. For inherently discrete-time systems, such as a radar and economic systems, Assumption G3 is irrelevant since the inter-sample behaviour is either not measured or not defined at all. For sampled continuous time systems Assumption G3 should be carefully checked but it is very often satisfied under reasonable conditions (see, for example [74]).

G4. *Full state feedback is available for control.*

In most situations this is not true and we need to reconstruct the state of the systems from output measurements. Design of observers for polynomial systems is an important issue that needs to be addressed in future research.

G5. *All measurements are noise free.*

We do not investigate theoretically how noise affects the controllers that we design. However, on certain occasions the effect of noise is tested using simulations.

2.3 A Prelude

Although the main topic of the thesis is nonlinear dead-beat control, it is very important to address some aspects of linear dead-beat control which are important for our work. A purpose of this section is to show by an example the salient features of linear dead-beat control. Some of the introduced definitions are illustrated in this way. Next, we present some examples of nonlinear systems with their dead-beat controllers. In this way we introduce some of the phenomena typical for the nonlinear systems. A comparison between nonlinear and linear dead-beat control completes the section.

2.3.1 Linear Dead-Beat Control

Let us consider a dead-beat controller for a linear, discrete-time, planar, deterministic system:

$$\begin{aligned} x_1(k+1) &= x_1(k) - x_2(k) + u(k) \\ x_2(k+1) &= 2x_1(k) + u(k) \end{aligned} \tag{2.6}$$

where $x_1(k)$, $x_2(k)$ and $u(k)$ are scalar state variables and control at time k. Suppose that we want to design a control law $u(k) = g(x(k))$, $x(k) = (x_1(k) \quad x_2(k))^T$, which transfers any initial state to the origin in minimum time. Let us first find the set of states $S_0 \subseteq \mathbb{R}^2$ that can be mapped to the origin in one step. Hence, we consider:

$$\begin{aligned} 0 &= x_1(0) - x_2(0) + u(0) \\ 0 &= 2x_1(0) + u(0) \end{aligned} \tag{2.7}$$

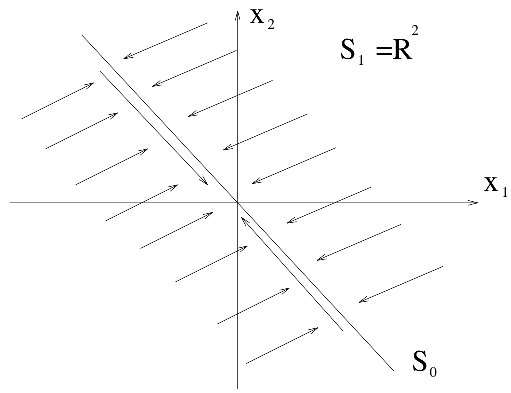

Figure 2.1: Sets S_0 and S_1 for the linear system.

It is immediate that the set of states for which there exists $u(0) \in \mathbb{R}$ which renders (2.7) zero simultaneously is defined by:

$$S_0 = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 - x_2 = 2x_1\} = \{x \in \mathbb{R}^2 : x_1 + x_2 = 0\} \tag{2.8}$$

This set is a line through the origin, see Figure 2.1. Let us now try to find the set of states that can be mapped in one step to S_0. Upon substituting $x_1(1) = x_1(0) - x_2(0) + u(0)$ and $x_2(1) = 2x_1(0) + u(0)$ into

$$x_1(1) + x_2(1) = 0$$

we obtain

$$S_1 = \{x \in \mathbb{R}^2 : \exists u \in \mathbb{R} \text{ such that } 3x_1 - x_2 + 2u = 0\} = \mathbb{R}^2 \tag{2.9}$$

Hence, for any initial state $x(0) \in \mathbb{R}^2$ we can find a sequence of two controls that transfer the state to the origin in two steps. We say that the system is *dead-beat controllable*.

Notice that any initial state $x(0) \in S_1$ needs to be transferred first to S_0 and then to the origin, see Figure 2.1. The control law which has this property is given by:

$$u(k) = -3x_1(k)/2 + x_2(k)/2 \tag{2.10}$$

Suppose now that we apply the control law (2.10) to the system (2.6). We obtain the closed loop system:

$$x(k+1) = A_{cl}x(k) = \begin{pmatrix} -1/2 & -1/2 \\ 1/2 & 1/2 \end{pmatrix} x(k) \tag{2.11}$$

The closed loop system has the property that $\forall x(0) \in \mathbb{R}^2$ we have that $x(k) = 0, \forall k \geq 2$, since

$A_{cl}^2=0$. Thus we have dead-beat behaviour and the control law (2.10) is called the *dead-beat control law*.

It is not difficult to see that the control law (2.10) is actually **time-optimal** since it transfers any initial state to the origin in minimum time.

The properties of the control law (2.10), which are in fact true for general linear controllable systems, are summarised below:

LP1 It is minimum-time control law.

LP2 The minimum-time control law is unique (for SISO linear controllable systems).

LP3 It is a linear static state feedback controller, which is a continuous function of state.

LP4 All eigenvalues of A_{cl} are zero and A_{cl} is nilpotent. Hence, the closed loop system is stable.

LP5 Sets S_0 and S_1 are linear subspaces of the state space.

LP6 There is a uniform bound on the dead-beat time. More precisely, any state can be transferred to the origin in the number of steps which is not greater than the order of the system.

LP7 The obtained control law is “feasible” in the sense that the magnitude of control is bounded on any bounded subset of the state space.

Actually, it can be shown that by placing the poles of the closed loop system at zero (when it is possible to do so) we always obtain a static linear state feedback controller which yields time-optimal performance. For a good overview of linear dead-beat control, see [151].

Let us consider more general (nonlinear) polynomial systems. If we follow the same idea of first computing the sets of states that can be transferred to the origin in k steps and then computing the control law which maps states from S_{k+1} to S_k, we have a design procedure for a minimum-time dead-beat controller. The difficulty in this approach is in finding an efficient way to compute these sets for classes of nonlinear systems. The simplicity of linear dead-beat controller comes from the linear structure of the system. The nonlinear structure introduces computational difficulties and the solution, even if obtained, is not so simple and elegant.

2.3.2 Nonlinear Dead-Beat Control

We now illustrate that none of linear dead-beat control properties **LP1**-**LP7** extends in general to polynomial nonlinear systems. We consider some examples of scalar polynomial systems, which are not too difficult to analyse and yet they seem to exhibit quite a few interesting features of general polynomial systems.

Example 2.1 (Properties **LP3, LP5** and **LP7** do not hold.) Consider the system $x(k+1) = (x(k)+1)\,u(k) + 2x(k)$. A dead-beat control law can be constructed as follows:

$$u(k) = \frac{-2x(k)}{(1+x(k))},\ x(k) \neq -1$$

and

$$u(k) = 0,\ x(k) = -1.$$

In this case we have that $u(k) \to \pm\infty$ as $x(k) \to -1$. Therefore, the control law is practically not feasible in the sense of **LP7**. Notice that this can not happen in the case of linear dead-beat control. However, we can modify the obtained control law:

$$u(k) = \frac{-2x(k)}{(1+x(k))},\ x(k) \notin [-0.5, -1.5]$$

and

$$u(k) = -2,\ x(k) \in [-0.5, -1.5].$$

This example shows the phenomenon of a trade-off between the minimum number of steps and magnitudes of control signals. This property is an important issue in general dead-beat control for polynomial systems. By considering unbounded control $u \in]-\infty, +\infty[$ in the early design phase, a more practical bounded control action can be designed as a second phase by a similar modifications. We will assume in most situations that $u \in]-\infty, +\infty[$. □

Example 2.2 (Properties **LP2, LP3, LP4** and **LP5** do not hold.) Consider the system $x(k+1) = u^2(k) + x(k)^2(1 - x^2(k))$. We can see that a possible control law which drives any initial state to the origin is

$$\begin{aligned}
u(k) &= \sqrt{x^2(k)\,(x^2(k)-1)} \text{ if } |x(k)| \geq 1 \\
u(k) &= \sqrt{x^2(k)\,(x^2(k)-1) + K},\ \text{if } |x(k)| < 1,\ \text{where } K > 1. \\
u(k) &= 0 \text{ if } x(k) = 0.
\end{aligned}$$

Obviously, any initial state such that $|x| \geq 1$ is driven to the origin in the first step and any other state is zeroed in two steps. It is clear that the minimum number of steps is two and that the control law is not continuous. In this case, it is not difficult to see that there does not exist a stabilising dead-beat control law, but if we apply u=0, $\forall |x(k)| < 1$ we do obtain an asymptotically stable system, in the sense of Lyapunov. This example shows that minimum-time dead-beat control does not imply stability for nonlinear systems. Notice also that by choosing different values for K in the control law, we obtain different minimum-time dead-beat controllers. There are infinitely many solutions for the time-optimal problem. □

Example 2.3 (Properties **LP3, LP5** and **LP6** do not hold.) Consider the system $x(k+1) = x(k) + u(k) + u^2(k)$. Let us introduce the function $\lfloor x \rfloor : \mathbb{R} \to \mathbb{N}$, defined as follows:

$$\lfloor x \rfloor = k, k \in \mathbb{N}, \text{ if } x \in]k, k+1]$$

It is not difficult to see that a minimum time to transfer any initial state $x(0) \leq 0.25$ to the origin is one step. On the other hand, for $x(0) > 0.25$ we have that the minimum time is given by $\nu(x(0)) = \lfloor 4x(0) \rfloor + 1$. Obviously, as $x(0) \to +\infty$ we have that $\nu(x(0)) \to \infty$. In other words, there is no uniform bound on the dead-beat time. □

We summarise below the properties of the given control laws:

NLP1 The control laws are static and discontinuous (state feedback) in all three examples.

NLP2 Minimum-time dead-beat controller does not guarantee stability (Example 2.2).

NLP3 Sets S_k are not linear subspaces of the state space (all three examples). Actually, they are semi-algebraic sets (sets defined by polynomial equations and inequalities).

NLP4 There may be no uniform bound on the dead-beat time (Example 2.3).

NLP5 The obtained control law may not be feasible in the sense of **LP7** (Example 2.1).

NLP6 There may exist an infinite number of minimum-time dead-beat controllers (Example 2.2).

When we compare properties of linear dead-beat control **LP1**-**LP7** with properties of nonlinear dead-beat control **NLP1**-**NLP6**, we see that no single property extends from linear to nonlinear context.

Chapter 3

Deciding Dead-Beat Controllability Using QEPCAD

3.1 Introduction

A constructive approach to the state and output dead-beat controllability problems for polynomial systems with rational coefficients is presented in this chapter. The controllability tests make use of the Cylindrical Algebraic Decomposition (CAD) algorithm [5, 33, 35, 34], which represents a part of a Quantifier Elimination (QE) procedure for real closed fields. CAD and QE are implemented in a symbolic computation package called QEPCAD (Quantifier Elimination by Partial Cylindrical Algebraic Decomposition) [35].

The construction of state/output dead-beat controllers is implicit in our approach. More precisely, we design a family of minimum-time state/output dead-beat controllers, which come in the form of discontinuous static state feedback controllers. The method gives a systematic way to stabilise a plant in certain situations (see also Chapter 11). Because of the generality of the approach, the obtained controllability tests and design methods are computationally demanding.

In general, the computational cost of our approach may be prohibitive. It is argued that the computational cost of solving a particular dead-beat problem may be a more appropriate way of defining the complexity of a class of polynomial systems than the structure of the system (such as bilinear, scalar, etc.). We pay some attention to these aspects. The method is efficient for polynomial systems of moderate order whose defining polynomials have small multi-degrees, such as classes of bilinear and Hammerstein systems.

First, we shortly explain the QEPCAD algorithm. For more details on real algebraic geometry and QEPCAD we refer to Appendix B, although due to the space limitations we do not present a complete theory. A very good introduction to QEPCAD is given in [92] and a more detailed presentation is given in [5, 33, 35, 34]. Second, we discuss the computation of sets S_k and $\hat{S}_k$ using QEPCAD (see equations (2.3) and (2.4) in Chapter 2). Some properties of these sets are shown. This leads to two different procedures that can be used to compute these sets. The difference comes from the formulation of decision and quantifier elimination problems that are used in the computation of the sets. Next, we present the state and output dead-beat controllability tests for the class of polynomial systems with rational coefficients. A number of properties and interesting phenomena are commented on. In particular, the phenomenon which leads to a non-terminating procedure in controllability tests is identified. Finally, examples that we present in the last section illustrate our approach.

At this stage, we point out that the outline of the thesis does not follow the chronological evolution of our research. Indeed, a large number of results on simpler classes of systems had been obtained before it was realised that QEPCAD can in principle be used to solve the dead-beat control problems in a very standard way. Having found the solution, it was realised then that constraining the structure of the system is a good way to obtain more insights about the geometry of the problem. In subsequent chapters we pursue this same idea of exploiting the structure to simplify considerations and understand the underlying phenomena.

3.2 Class of Systems

In this chapter we consider the class of polynomial systems given by:

$$
\begin{aligned}
x(k+1) &= f(x(k), u(k)), \\
y(k) &= h(x(k))
\end{aligned}
\tag{3.1}
$$

where $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}$ and $u(k) \in \mathbb{R}$ are respectively the state, the output and the input of the system (3.1) at time k. The vector $f(x, u) = (f_1(x, u) \dots f_n(x, u))^T$ is such that $f_i(x, u) \in \mathbb{Q}[x, u] = \mathbb{Q}[x_1, x_2, \dots, x_n, u]$ and $h \in \mathbb{Q}[x_1, \dots, x_n]$, which means that we assume f_i and h have rational coefficients.

The class of systems (3.1) represents a very large subclass of general polynomial systems. In fact, in practice we always deal with (3.1) since any irrational coefficients are approximated with a desired accuracy by rational coefficients. Indeed, all applications of polynomial models that are given in Appendix A fall into this category.

3.3 A Short Introduction to QEPCAD

It is a well known fact that given the equation $a_2u^2 + a_1u + a_0{=}0$ in four variables u, a_0, a_1, a_2, there exists a real solution for u if and only if the discriminant $a_1^2 - 4a_2a_0$ is not negative. Hence, we have a condition on the variables a_0, a_1, a_2 alone, which guarantees the existence of a real solution u to the original equation. The Sturm Theorem [87] establishes a similar result for any univariate polynomial $f(u)$ to have a real root.

A set of formulas which consist of polynomial inequalities, equations, Boolean operators ($\wedge, \vee, \neg, \rightarrow$) and quantifiers ($\exists, \forall$), represent a sentence in the so called first-order theory of real closed fields. A very important question is: given any such sentence, is it possible to find an equivalent formula without any quantified variables? In other words, is *quantifier elimination* possible in the first-order theory of real closed fields? Using our example we can see that the sentence

$$(\exists u)\ [a_2u^2 + a_1u + a_0{=}0]$$

has the solution (quantifier free formula)

$$a_1^2 - 4a_2a_0 \geq 0.$$

It should be emphasized that all variables in the above given formulas are assumed to be *real*.

Tarski proved that QE is always possible in the first order-theory of real closed fields. In other words, he proved that given any sentence, it is possible to obtain a quantifier free formula in *finite time*. Moreover, he also presented an algorithm to carry out such elimination. Unfortunately, the algorithm was highly impractical since the time bound for computing a quantifier free formula can not be estimated by any tower of exponential functions [73]. Nevertheless, Tarski's result solved the existence problem and this generated a large body of results in the search for more efficient QE algorithms (see for instance [4]). It was not until 1973 that a more practical method for QE was found. The QE algorithm hinges on the CAD algorithm [33, 35], which was first discovered by Collins in 1973. Since then a number of improvements have been reported in literature.

There are several known methods for QE [5, 73, 87], but the most important one (from a computational viewpoint) is based on CAD [5, 33, 35, 34]. It is important to mention that the Grigor'ev algorithm for quantifier elimination [73] has a sub-exponential computation time but we are not aware whether it has a computer implementation. For more details on CAD and QE see Appendix B.

QEPCAD[1] [35] is the name of a software program where CAD and QE are implemented. CAD represents a part of a QE procedure for real closed fields and is based on the SACLIB package which was developed by G. E. Collins and a number of other researchers (for a more detailed description of the algorithm see [33, 35, 34, 92]). QEPCAD is probably the only general method for QE, which has a software implementation, available at this time.

The input to the QEPCAD algorithm is any sentence of the above mentioned form and the output of the algorithm is a quantifier free expression. It is important to emphasize that QEPCAD is an interactive program and solving non-trivial problems requires a detailed knowledge of its operation (see, for instance [34, 92]).

QEPCAD was found to be useful in motion planning [35], output stabilisability of linear systems [3], bang-bang control [68]. Recently I have become aware of a reference [92], where a number of linear and nonlinear continuous time control problems, are tackled using QEPCAD. We show below that QEPCAD can also be used in deciding state and output dead-beat controllability of polynomial systems (3.1).

3.4 State Dead-Beat Control

In Chapter 2 we showed using a linear and three scalar polynomial systems that the sets S_k (see equations (2.3)) play a very important role for the state dead-beat control problem. Computation of these sets leads naturally to a constructive way of checking whether a system is state dead-beat controllable or not. By constructive, we mean that a minimum-time dead-beat controller is obtained easily once the sets S_k have been computed. It is shown below how it is possible to use QEPCAD to compute the sets S_k and how this information can be used to test whether a polynomial system (3.1) is state dead-beat controllable.

In this section we use the following:

Assumption 3.1 We assume that the system (3.1) satisfies $f(0,0)=0$. □

[1]QEPCAD is still not available in any of the commercial computation packages and is still being perfected.

3.4.1 Computation of Sets S_k and $\hat{S}_k$

The main purpose of this section is to show how the sets S_k and $\hat{S}_k$ can be computed using QEPCAD. We present two different procedures that can be used to obtain the sets S_k. It is shown that both procedures lead to the same solution - the defining formulas for the sets S_k. In Chapter 7 we show that for simple Hammerstein systems the sets S_k and $\hat{S}_k$ may be possible to compute without resorting to QEPCAD. However, the class of simple Hammerstein system is but a small subclass of general polynomial systems with rational coefficients.

Procedure 1: A straightforward application of the definitions of set S_0 leads to the following input formula to QEPCAD, which is used to compute the expression that defines the set S_0 (we omit the time index for the state, that is we write x instead $x(0)$).

$$(\exists u(0))\,[f(x, u(0))=0]$$

QEPCAD computes a quantifier free formula, which depends only on x, for which the above given formula is true. We emphasize that all variables in the above given formula are assumed to be real. The output formula is of the following form:

$$\bigvee_{i=1}^{p^0}\Big(\bigwedge_{j=1}^{r_i^0} t_{i,j}^0(x) \quad m_{i,j}^0 \;\; 0\Big) \tag{3.2}$$

where $t_{i,j}^0 \in \mathbb{Q}[x_1, \ldots, x_n]$ and

$$m_{i,j}^0 \in \{>, <, =\}, \;\; j=1, \ldots, r_i^0, \; i=1, \ldots, p^0$$

For example, the defining expression for S_0 may have the form $((x_1^2 + x_2=0) \bigwedge (x_2 \geq 0)) \bigvee (x_1 - x_2 + 1 < 0)$. We denote the obtained formula for the set S_0 given by (3.2) as $S_0(x)$. We use the same notation $S_k(x)$ to denote the defining formulas for the set S_k. Hence, we write by definition:

$$S_0=\{x \in \mathbb{R}^n : S_0(x)\}$$

The computation of sets $S_k, \forall k$ can be done in the same manner. The input formula to QEPCAD used to compute the set S_k is:

$$(\exists u(0))(\exists u(1))\ldots(\exists u(k))\,[f_{u(k)} \circ \ldots \circ f_{u(0)}(x)=0] \tag{3.3}$$

and the output formula is of the form:

$$S_k(x) = \bigvee_{i=1}^{p^k} (\bigwedge_{j=1}^{r_i^k} t_{i,j}^k(x) \quad m_{i,j}^k \quad 0) \, , \; k=0,1,\ldots$$

where $t_{i,j}^k \in \mathbb{Q}[x_1,\ldots,x_n]$ and $m_{i,j}^k \in \{>,<,=\}$, $j=1,\ldots,r_i^k$, $i=1,\ldots,p^k$, $k=0,1,\ldots$. In other words, QEPCAD is used to project the variety $V(f_{u(k)} \circ \ldots f_{u(0)}(x))$ in the space $\mathbb{R}^{n+k+1}$ (n states and $k+1$ controls) onto the space $\mathbb{R}^n$ which is the ambient space of the vector x. An important consequence of the definition of the sets S_k (see equations (2.3)) is

$$S_k \subseteq S_{k+1}, \;\; k=0,1,2,\ldots$$

The sets $\hat{S}_k$ are easily obtained from the formulas:

$$\hat{S}_0 = S_0, \text{ and } \hat{S}_k = S_k - S_{k-1}, \; k=1,2,\ldots$$

Comment 3.1 Procedure 1 is computationally very expensive for general polynomial systems (3.1). Notice that each of the sets S_k (and therefore $\hat{S}_k$) can be computed in finite time. However, the computation time increases rapidly as the number of variables in the input formulas increases. The computation time depends roughly doubly exponentially with respect to the number of variables in the input polynomials (see Appendix B). The bounds on the computation time for the original algorithm can also be found in [33] and the improvements are discussed using some examples in [35]. We need more time to compute the set S_{k+1} than the set S_k using Procedure 1. However, the examples that are presented below show that for moderate multi-degrees of polynomials and low order polynomial systems this approach may still yield satisfactory answers. We note that in general it is very difficult to tackle more than 4 variables with QEPCAD. □

Procedure 1 does not exploit the recursive nature (compositions of a map) of the formulas that are used to compute the sets S_k. As a result, to compute $S_k(x)$ it is necessary to compute CAD[2] of $\mathbb{R}^{n+k+1}$. This is undesirable because the computation time depends roughly double exponentially in the number of variables in the input polynomials. We show below that it is possible to keep the number of input variables at $n+1$ in the computation of all sets S_k. This may improve the efficiency of the algorithm considerably.

[2]See the definition of CAD in Appendix B.

Procedure 2: First, $S_0(x)$ (3.2) is computed in the same way as in Procedure 1. Notice that the number of variables in the input polynomials is $n+1$, that is $x{=}(x_1 \ \dots \ x_n)^T$ and $u(0)$ (for Procedure 2 we omit the time index for the control u). We introduce the following notation:

$$S_0 \circ f_u(x) = S_0(f(x,u))$$

to denote the set of formulas obtained from $S_0(x)$ when $f(x,u)$ is substituted for x. Consider the following QE problem:

$$(\exists u)\,[S_0 \circ f_u(x)] \tag{3.4}$$

We show that the quantifier free formula of this problem is equivalent to $S_1(x)$. Indeed, notice first that the formula can be interpreted as follows: "find all initial states x in $\mathbb{R}^n$ for which there exists a real control u which transfers the initial state to the set S_0." We show that this is equivalent to saying that: "find all the states which are such that the minimum time to transfer them to the origin is at most 2 time steps". Notice that the second formulation is exactly the definition of the set S_1 (2.3). Denote the formulas obtained by considering the quantifier elimination problem (3.4) as $S^*(x)$ (the set is denoted as S^*) and let us prove that $S_1(x) = S^*(x)$ (that is $S_1{=}S^*$).

Indeed, since $f(0,0) = 0$ and $0 \in S_0$ by definition, it follows that for any initial state $x \in S_0$ there is a control action u which maps the state in the next step to S_0, that is to the origin itself. Hence, we have $S_0 \subset S^*$ States in S_0^C that are mapped to S_0 in one step constitute exactly $\hat{S}_1$. Hence $S^*{=}S_0 \cup \hat{S}_1$, and using equations (2.3) we have $S_1{=}S^*$.

In a very similar manner, we can show that by considering the quantifier elimination problems:

$$(\exists u)\,[S_{j-1} \circ f_u(x)]\,,\ j{=}1,2,\dots$$

we obtain, using QEPCAD, the defining expressions $S_j(x)\,, \forall j$.

Comment 3.2 It is essential to notice that for the computation of formulas $S_j(x)$, using Procedure 2, we are dealing with $n+1$ variables at each step, that is $\forall j$. In other words, in order to compute any of the sets S_j, a CAD for $\mathbb{R}^{n+1}$ should only be computed. Note that if we use Procedure 1, we need to compute a CAD of $\mathbb{R}^{n+1+j}$ in order to obtain the set S_j. However, in Procedure 2 the computations are done sequentially and in order to compute $S_j(x)$ we *have to compute* $S_s(x)\,, s{=}0,1,\dots,j-1$. On the other hand, in Procedure 1, we could compute $S_j(x)$ *without having to compute* any of $S_s(x)\,, s{=}0,1,\dots,j-1$. □

3.4.2 State Dead-Beat Controllability Tests

The below given theorem follows easily from the above construction:

Theorem 3.1 *Suppose that there exists an integer N such that $S_N = S_{N+1}$. The system (3.1) is state dead-beat controllable if and only if $S_N = \mathbb{R}^n$.* □

From Theorem 3.1 and Procedures 1 and 2 we derive the following state dead-beat controllability tests. We use $(\exists x)$ to denote $(\exists x_1)(\exists x_2)\ldots(\exists x_n)$.

TEST 1:

0. Input: $f(x, u)$

1. Let $j=0$. Find the set S_0 using the following input formula to QEPCAD

$$(\exists u)\,[f(x, u) = 0]\,.$$

   The resulting formula is $S_j(x) = S_0(x)$.

2. $j = j + 1$

3. Find the composition $S_{j-1} \circ f_u(x)$. Compute $S_j(x)$ by considering

$$(\exists u)\,[S_{j-1} \circ f_u(x)]\,.$$

   Compare whether $S_j = S_{j-1}$. In other words, check whether the following formula is true or not

$$(\exists x)\,[S_j(x) \wedge \neg S_{j-1}(x)]$$

   If it is true, go to 2. If not, go to 4.

4. Check whether $S_j = \mathbb{R}^n$. That is, check whether the following formula is true or not

$$(\exists x)\,[\neg S_j(x)]$$

   If it is true, the system is not state dead-beat controllable and vice versa.

TEST 2:

0. Input: $f(x, u)$

1. Let j=0. Find the set S_0 using the following input formula to QEPCAD

$$(\exists u(0))\,[f(x,u(0))=0]\,.$$

2. j=$j+1$

3. Find the composition $f_{u(j)} \circ \ldots f_{u(0)}(x)$. Compute $S_j(x)$ using QEPCAD by considering

$$(\exists u(j))\ldots(\exists u(0))\,[f_{u(j)} \circ \ldots f_{u(0)}(x)=0]\,.$$

   Compare whether S_j=S_{j-1}. In other words, using QEPCAD check whether the following formula is true or not

$$(\exists x)\,[S_j(x) \wedge \neg S_{j-1}(x)]$$

   If it is true, go to 2. If not, go to 4.

4. Check whether S_j=$\mathbb{R}^n$. That is, using QEPCAD check whether the following formula is true or not

$$(\exists x)\,[\neg S_j(x)]$$

   If it is true, the system is not state dead-beat controllable and vice versa.

Comment 3.3 Notice that if for some N we have that S_N=S_{N+1} and $S_N \neq \mathbb{R}^n$, then there exists an invariant set S_N^C which is such that we can not escape from it no matter which control sequence we apply. This means that we *can not generate trajectories* using the control signal which are such that they start in S_N^C and have a non-empty intersection with the set S_N. Since the system is not dead-beat controllable, we have a nice analogy with uncontrollable subspaces of linear systems. However, the "dead-beat uncontrollable set" of the state space, that is S_N^C, is a geometrically more complex object than the subspaces in the linear case. Note that even if the system is not dead-beat controllable, the uncontrollable subsets of the state space are still important objects and their investigation reveals some structural properties of the systems [48, 82]. □

Comment 3.4 The problem with this approach is that there may be some systems for which the chain of sets $S_0 \subset S_1 \subset \ldots$ may not terminate (see Example 2.3). That is $S_k \neq S_{k-1}, \forall k$. However, even when the chain does not terminate, obtaining a characterisation of the sets S_j is important in its own right and may be used in the design of control laws, such as minimum-time dead-beat controllers.

We emphasize that for any given decision or quantifier elimination problem QEPCAD is guaranteed to find its solution in a finite number of steps. That is, the algorithm can compute any of the sets S_j in a finite time. However, for the dead-beat problem we need to use QEPCAD recursively and hence the above given controllability test may not stop in a finite time. An infinite loop may occur in the controllability test if $S_{j+1} \neq S_j, \forall j$.

The existence of the infinite loop in TESTS 1 and 2 is one of the main problems that may arise when using the tests. However, it reflects an inherent property of a class of polynomial systems and it can not be regarded as a drawback of our method. Recognising the classes of systems for which we can say *a priory* whether there exists a dead-beat controllability test that stops in finite time appears to be very important. If this was possible, we would know for which systems we may expect to have problems when dealing with the dead-beat controllability problem.

Several classes of polynomial systems for which there exist finitely computable dead-beat controllability tests have been found in the literature. They are (besides linear discrete-time systems) a class of bilinear systems [48, 71, 70], a class of linear systems with positive controls [50] and a class of inhomogeneous bilinear systems [49]. In subsequent chapters three similar results are presented. In Chapter 5 we find for scalar polynomial systems a dead-beat controllability test which stops in finite time in a generic sense. In Chapter 6 we show for a class of NARMAX models that the output dead-beat controllability test stops always in finite time. Finally, in Chapter 8 we prove that the state dead-beat controllability test for generalised Hammerstein systems is also finitely computable. $\square$

Comment 3.5 It is not difficult to include bounds on controls in the QEPCAD based state dead-beat controllability test. In other words, controllability with bounded ($|u(k)| \leq C$) or positive ($u(k) \geq 0$) controls can be checked in the same way. We just need to add several equations in the input set of equations to QEPCAD. For example, in the case of positive controls $u \geq 0$, we compute $S_0(x)$ using the formula:

$$(\exists u)\ [(u \geq 0) \wedge (f(x, u) = 0)]$$

The computation of $S_j(x)$ can be carried out by considering $(\exists u)\ [(u \geq 0) \wedge (S_{j-1} \circ f_u(x))]$. In the case of linear systems with bounded controls the chain $S_0 \subset S_1 \subset \ldots$ may not terminate for dead-beat controllable systems [174]. In this case, however, instead of checking the controllability on $\mathbb{R}^n$ we may need to work on a bounded subset of the state space $\mathcal{B} \subset \mathbb{R}^n$. This generalises the approach of Desoer and Wing for minimum-time dead-beat control of linear systems with bounded controls [174].

Notice that we can include constraints on states as well. We check whether $S_N \supseteq \mathcal{B}$. There may be no uniform bound on the dead-beat time. Our method is constructive and it may be an alternative to some known non-constructive methods, such as Evans' controllability results for linear systems with positive controls [51] or for a class of bilinear systems [48].

Therefore, straightforward changes to the procedure for computing sets S_j are needed to include very general constraints on state and controls at the same time. It is essential to notice that the constraints do not have to be convex and they are given by polynomial expressions. Very general dead-beat problems, for which there does not exist any other method, can be tackled in this way. $\square$

Comment 3.6 The power of the QEPCAD based controllability tests is the generality of the dead-beat problems that can be solved in this way without having to tackle one class of polynomial systems at a time. However, the generality of the method is at the same time its main drawback since it implies that it can solve very difficult problems as well. This means that in some cases the computation time would be too large for the algorithm to be of practical value.

One way of reducing the required computations is to introduce some additional assumptions on the structure of the system. This is the main reason why we investigate in the subsequent chapters a number of simpler classes of polynomial systems. By creatively using some structural assumptions and QEPCAD, one may obtain feasible dead-beat controllability tests for non-trivial classes of systems. Good examples of the application of such ideas are Chapters 4, 5, 6 and 9.

Another way to reduce the computations is to modify the algorithm itself but we have not pursued it in this thesis. It appears that the computation of a complete CAD may not be necessary and that simpler procedures my be possible in certain cases. For example, there have been reports on QE methods which can deal only with certain classes of problems, such as for formulas defined by linear or quadratic polynomials, but the number of variables that can be tackled can be very high (see, for instance, [83, 101, 173, 112] and references therein). Identifying classes of polynomial systems for which such more efficient methods can be used seems to be a very important question. We conclude by saying that computational real algebraic geometry is a rapidly changing field and it is very difficult to speculate on the possible advances that may be relevant to applications. □

Comment 3.7 In view of the last comment, it seems worthwhile identifying different ways in which the computational complexity of the dead-beat problem may be reduced. In subsequent chapters we revisit this idea frequently. Here, we give several possible ways in which we might tackle complexity, which are not addressed elsewhere in the thesis.

We point out that QEPCAD is an important tool which can be used flexibly in answering different controllability questions. To illustrate our claim consider the state linear systems (see, for instance [117]) of the form:

$$x(k+1) = F(u(k))\,x(k) \tag{3.5}$$

where F is an $n \times n$ matrix whose entries are polynomials in control u. Applications of this class of models in the identification of certain power systems has been reported in [117]. Notice that discrete-time bilinear systems are a subclass of (3.5).

The structure of this class of systems is very suitable for the implementation of *periodic or quasi periodic* open loop controllers. In other words, by applying a periodic sequence of controls of finite length ($u(k), u(k+1), \ldots, u(k+M), \forall k = j(M+1), j \in \mathbb{N}$), which does not depend on the state (open loop), we may achieve dead-beat behaviour. This control scheme may be used to investigate dead-beat controllability but it is unlikely that it would perform well if applied to a real system, since it is an open loop scheme. The first reference that we are aware of which uses these ideas for controllability of linear systems with output feedback is [127].

This technique is also closely related to the problem of arbitrary eigenvalue assignment by means of periodic static output feedback for linear systems [1, 2, 72] since the underlying structure of this problem is actually bilinear.

Consider the system (3.5) when the sequence $u(k), u(k+1), \ldots, u(k+M), k = j(M+1), j \in \mathbb{N}$ is applied to it:

$$x(k+M+1) = \underbrace{F(u(k+M))\,F(u(k+M-1))\ldots F(u(k))}_{\mathcal{F}}\,x(k)$$

The entries of matrix $\mathcal{F}$ are polynomials in controls $u(k), u(k+1), \ldots, u(k+M)$. Therefore, the coefficients $l_i, i=0, 1, \ldots, n-1$ of the characteristic polynomial of matrix $\mathcal{F}$, which is denoted as $\det(\lambda I - \mathcal{F}) = \lambda^n + l_{n-1}\lambda^{n-1} + \ldots + l_0$, are also polynomials in controls $u(k), u(k+$

$1), \ldots, u(k+M)$. Hence, we can try to assign the eigenvalues of $\mathcal{F}$ by means of controls. If we assign all the eigenvalues of the characteristic polynomial at zero, we obtain dead-beat behaviour since the matrix $\mathcal{F}$ is nilpotent. In other words, if there exist controls $u(k), u(k+1), \ldots, u(k+M)$ which yield l_i=0, $\forall i$=0, $1, \ldots, n-1$, the system (3.5) is dead-beat controllable. This problem is solved by QEPCAD by considering:

$$(\exists u(k))\ldots(\exists u(k+M))\,[l_0{=}0 \wedge \ldots \wedge l_{n-1}{=}0]$$

Additionally, in [167] a concept of *asymptotic controllability* was introduced. By using, for example, the Jury criterion for stability we can find a set of expressions $L(l_0, l_1, \ldots, l_{n-1})$ in l_i, i=0, $1, \ldots, l_{n-1}$ which guarantee that the matrix $\mathcal{F}$ is stable (has all eigenvalues in the open unit disc). Actually, since l_i are polynomials in controls, the expressions $L(l_0, l_1, \ldots, l_{n-1})$ also depend in a polynomial fashion on the controls $u(k), u(k+1), \ldots, u(k+M)$. Consequently, one can easily check the asymptotic controllability property using QEPCAD by considering:

$$(\exists u(k))\ldots(\exists u(k+M))\,[L(l_0, l_1, \ldots, l_{n-1})]\,.$$

Notice that if no solution exists for a control sequence of length $M+1$, it still may exist for a longer sequence.

Let us consider another example of how computations may be reduced when using QEPCAD. Suppose that the set S_{n-1} has been computed using QEPCAD but we could not compute the set S_n due to the complexity of the underlying QE problem (this often happens since the QE problem for computation of S_{k-1} may be "much easier" than the corresponding one for S_k). We have at our disposal the defining expression $S_{n-1}(x)$. Suppose that dim S_{n-1}=n and that there exists a set $\mathcal{V}_{n-1}$ such that it is described by a very simple expression $\mathcal{V}_{n-1}(x)$. For example, $\mathcal{V}_{n-1}$ may be a hyperball (defined by quadratic inequalities) or hypercube (defined by linear inequalities) centred at a point x^*. It is not difficult to formulate a decision problem which would solve the existence of such a set. If QEPCAD can compute the expression $\mathcal{V}_{n-1}(x)$, we may be able to compute a subset of S_n using the following quantifier elimination problem:

$$(\exists u)\,[\mathcal{V}_{n-1} \circ f_u(x)]$$

Suppose that we solved the above problem and obtained the set S'_n. We may be able to "fit" a set $\mathcal{V}_n$ inside S'_n and repeat the same procedure, etc. It is surprising that this procedure may yield $\mathcal{V}_N = \mathbb{R}^n$. In Example 3.1 we illustrate this procedure and show that using this method we can gain huge savings in computations. □

Comment 3.8 It is easy to see that a family of state dead-beat controllers can be designed using this approach. Indeed, assume that the system under consideration is state dead-beat controllable, that is $S_N = \mathbb{R}^n$. The sets $S_j, j=0,1,\ldots,N$ have been obtained using QEPCAD and are defined by $S_j(x)$. Having found the defining expressions $S_j(x)$, it is straightforward to obtain formulas $\hat{S}_j$. From the definition of sets $\hat{S}_j$ it follows that $\forall x \in \hat{S}_{j+1}, \forall k=0,\ldots,N-1,\ \exists u \in \Re$ such that $\hat{S}_j \circ f_u(x)$ is satisfied. Hence, once we have obtained the defining expressions for the sets $\hat{S}_j$ we can easily find controllers which are such that they map $\hat{S}_{j+1}$ to $\hat{S}_j$, $\forall j=0,1,\ldots,N-1$. Indeed, $x(k)$ is measured at each time step k and it is input to the the feedback controller, given by:

$$u(k) = \text{any real root } u \text{ to } \begin{cases} f(x(k),u)=0 & \text{, if } x(k) \in \hat{S}_0 \\ \hat{S}_0 \circ f_u(x(k)) & \text{, if } x(k) \in \hat{S}_1 \\ \ldots & \ldots \\ \hat{S}_{N-1} \circ f_u(x(k)) & \text{, if } x(k) \in \hat{S}_N \end{cases}$$

Notice that the control $u(k)$, which transfers the state $x(0)$ from $\hat{S}_{j+1}$ to $\hat{S}_j$, may not be unique. In this way we obtain a family of all minimum-time dead-beat controllers, which are by construction discontinuous and nonlinear state feedback controllers. Checking the conditions $x(k) \in \hat{S}_j,\ j=0,1,\ldots,N$ is done by checking whether the expressions $\hat{S}_j(x)$ are satisfied for $x=x(k)$.

It is important to say that we can also use the controller, which has the structure:

$$u(k) = \text{any real root } u \text{ to } \begin{cases} f(x(k),u)=0 & \text{, if } x(k) \in S_0 \\ S_0 \circ f_u(x(k)) & \text{, if } x(k) \in S_1 \\ \ldots & \ldots \\ S_{N-1} \circ f_u(x(k)) & \text{, if } x(k) \in S_N \end{cases}$$

Notice, however, that in this case the order in which we check the condition $x(k) \in S_j, j=0,1,\ldots,N$ becomes crucial for the controller to operate properly. Indeed, we *have to* first check if $x(k) \in S_0$. If this is true, apply the corresponding control. If not, check if $x(k) \in S_1$, etc. The above given controller might not yield minimum-time behaviour if we first checked whether $x(k) \in S_j, j > 0$.

Indeed, by definition $S_0 \subset S_j, j$=$1, 2, \ldots$ and suppose we implement the controller such that we first check if $x(k) \in S_1$. If it happens that $x(k) \in S_0$ then automatically $x(k) \in S_1$. The controller would apply the control which transfers the state $x(k)$ to the set S_0, but not necessarily the one which would transfer it to the origin! This is the main motivation for introducing sets $\hat{S}_j$ since in this case the structure of the controller is more precisely defined. □

Comment 3.9 Notice that dead-beat controllability of implicit polynomial systems, can be tackled in the same way. These systems are defined as:

$$F(x(k+1), x(k), u(k)) = 0$$

where F is a polynomial with rational coefficients. In the paper [184] the authors provide an identification scheme for identifying the so called rational NARMAX models. These models can be regarded as implicit polynomial. □

3.5 *Output Dead-Beat Control*

A very similar procedure can be used to deal with output dead-beat controllability and control of (3.1). In the case of state dead-beat control our goal was to control the system to the origin, and since we assumed that $f(0, 0) = 0$ we could keep the state at the origin for all future time steps. We say that the target set, to which we need to steer any other state, is the origin itself.

In the case of output dead-beat control we are interested in zeroing the output of the system in finite time. Moreover, we want to keep the output at zero for all future time steps once we have zeroed it (see Definition 2.2). In order to achieve this, it is necessary to first compute the target set $T \subset \mathbb{R}^n$, which is such that the output of the system is zero for any state in T and moreover it is possible to find a control sequence $\mathcal{U}$ which keeps the trajectories emanating from any initial state in T in the set T itself.

3.5.1 *Computation of Sets* T_j *and* S_j^O

In order to compute the target set T we need to introduce sets T_j, which are used in the computation of T.

Definition 3.1 Sets T_j are defined as follows:

$$\begin{aligned} T_0 &= V(h) = \{x : h(x) = 0\} \\ T_j &= \{x \in V(h) : \exists u(0), \ldots, u(j-1) \in \mathbb{R} \text{ such that } f_{u(l)} \circ \ldots f_{u(0)}(x) \in V(h) \\ &\quad , \forall l = 0, \ldots, j-1\} \end{aligned} \tag{3.6}$$

In other words, the set T_j consists of all states $x \in V(h)$ for which there exists a sequence of controls $u(0), \ldots, u(j-1)$ which keeps the trajectory emanating from x in the variety $V(h)$ for j consecutive time steps.

We use the same notation as in the previous section: $T_j(x)$ represents the defining expression for the set T_j and $T_j \circ f_u(x) = T_j(f(x,u))$. By definition we have that $T_0(x) \equiv (h(x) = 0)$. $T_1(x)$ can be computed using the following QE problem

$$(\exists u)\,[(h \circ f_u(x) = 0) \wedge (h = 0)]\,.$$

Moreover, straightforward calculations show that we have in general that $T_j(x)$ can be computed by considering the following QE problem:

$$(\exists u)\,[T_{j-1} \circ f_u(x) \wedge T_{j-1}(x)]\,.$$

The sets T_j are crucial in computing the target set T. On the target set T we have that the output is zero and for any initial state in T we can find a sequence of controls which keeps the state in T for all future time steps. If there exists an integer N^* such that $T_{N^*} = T_{N^*+1}$ we have that $T = T_{N^*}$.

Suppose that the target set has been computed. It is given by

$$T = \{x : \bigvee_{i=1}^{P} (\bigwedge_{j=1}^{R_i} t_{i,j}(x) \quad m_{i,j} \ \ 0)\}$$

where $m_{i,j} \in \{<, >, =\}$ and $t_{ij} \in \mathbb{Q}[x_1, \ldots, x_n]$. We use the usual shorthand writing $T = \{x : T(x)\}$.

We now need to define the set S_j^O.

Definition 3.2 Sets T_j are defined as follows:

$$S_0^O = \{x : T(x)\}$$

$$S_j^O = \{x : \exists u(0), \ldots, u(j-1) \in \mathbb{R} \text{ such that } T \circ f_{u(j-1)} \circ \ldots f_{u(0)}(x)\} \tag{3.7}$$

In other words, the sets $S_j^O, j{=}1, 2, \ldots$ are sets of states that can be transferred to the target set in one, two, etc. time steps. We have denoted $S_0^O(x) = T(x)$. Defining expressions $S_j^O(x)$ for sets $S_j^O, j{=}1, 2, \ldots$ can be computed by considering the QE problems:

$$(\exists u)\,[S_{j-1}^O \circ f_u(x)]$$

Comment 3.10 Notice that we could find another set of QE formulas in order to compute $T_j(x)$ and $S_j^O(x)$, similar to Procedures 1 and 2 in the previous section. However, we presented only the ones which exploit the recursive nature of these sets for reasons presented in Comment 3.2. □

3.5.2 Output Dead-Beat Controllability Test

Using the introduced sets, we can state the following

Theorem 3.2 *Suppose that the target set T has been computed and that there exists L such that $S_L^O = S_{L+1}^O$. The polynomial system is output dead-beat controllable if and only if $S_L^O = \mathbb{R}^n$.* □

The proof of Theorem 3.2 is obvious. The following output dead-beat controllability test is obtained from the previous subsection and the above given theorem.

TEST 3

1. (a) Let $j{=}0$ and $T_0(x) \equiv (h{=}0)$.

   (b) $j{=}j+1$

   (c) Find composition $T_{j-1} \circ f_u(x)$ and compute $T_j(x)$ by considering:

   $$(\exists u)\,[T_{j-1} \circ f_u(x) \wedge T_{j-1}(x)]$$

   Consider now whether $T_j{=}T_{j-1}$. Hence we consider if the following decision problem is true

   $$(\exists x)\,[T_{j-1}(x) \wedge \neg T_j(x)]$$

   If it is not true, go to 2 and define $T(x) = T_j(x)$. If it is true, go to 1.(b).

2. We have computed $T(x)$.

(a) Let j=0 and define $S_0^O(x)$ =$T(x)$.

(b) j=$j+1$

(c) Find composition $S_{j-1}^O \circ f_u(x)$. Compute $S_j^O(x)$ by considering the QE problem:

$$(\exists u)\,[S_{j-1}^O \circ f_u(x)]$$

Check if S_j^O=S_{j-1}^O by considering whether the decision problem:

$$(\exists x)\,[\neg S_{j-1}^O(x) \wedge S_j^O(x)]$$

is true or not. If it is true go to 2.(d). If it is not true, go to 2.(b).

(d) Check if S_j^O=$\mathbb{R}^n$ by considering whether the decision problem

$$(\exists x)\,[\neg S_j^O(x)]$$

is true or not. If it is not true, the system is output dead-beat controllable and vice versa.

Comment 3.11 Notice that the procedure used for computing the target set T may not terminate in finitely many steps. In other words, we may have that $T_0 \supset T_1 \supset T_2 \supset \ldots$. However, we can still compute a subset of the target set as follows:

$$T^*=\{x : h(x)=0 \text{ and } \exists u \in \mathbb{R} \text{ such that } x=f(x,u)\}$$

and investigate sets of states that are controllable to T^* in one, two, etc. time steps. Notice also that if we assume that $f(0,0)=0 \wedge h(0)=0$, the origin is always contained in T^* and therefore state dead-beat controllability implies output dead-beat controllability whereas the opposite is not true. In general, we do not need this assumption when considering output dead-beat controllability. □

From the above given test and comments we can see that deciding output dead-beat controllability is usually more difficult than deciding state dead-beat controllability. We emphasize that two infinite loops may occur in the above algorithm. One may occur when computing the target set T, that is $T_j \neq T_{j+1}, \forall j$ and another when computing the set S_j^O when it happens that $S_j^O \neq S_{j+1}^O, \forall j$.

QEPCAD based approach can be regarded as a unified approach to state/output dead-beat controllability and control of polynomial systems (3.1). However, the main hindrance to its implementation is the computational complexity of the problem (for explicit bounds on the computation time refer to Appendix B). It is possible to reduce the complexity of the problem by either requiring less information about S_j (not a complete description) or by constraining the structure of the system (3.1). Although it is plausible in certain situations to require less information about sets S_j, the nature of the time-optimal problem does not allow us to exploit it. The inherent complexity of the class of systems that we consider, as well as the question that we want to answer, forces us to select a class of simpler systems which can be tackled more efficiently in order to obtain more explicit conditions and easier to check controllability tests. In the sequel we show how constraining the structure of (3.1) may reduce the computational complexity of the controllability test or even be used to obtain finitely computable conditions for controllability.

3.6 Examples

We present below several examples.

Example 3.1 Consider the scalar polynomial system:

$$x(k+1) = x(k)\,u^6(k) + (x(k)+1)\,u^3(k) - 2u^2(k) + 3x(k)\,u(k) + 2x(k) \tag{3.8}$$

The set S_0 is computed by using QEPCAD. We compute $S_0(x)$ by considering the QE problem:

$$(\exists u)\,[xu^6 + (x+1)\,u^3 - 2u^2 + 3xu + 2x = 0]\,.$$

QEPCAD computed $S_0(x)$ in 1.2 sec[3]:

$$\begin{aligned} S_0(x) \quad = \quad & (4123953x^7 + 13719780x^6 + 7007148x^5 - 2009664x^4 + 382968x^3 + \\ & 901620x^2 - 130208x - 1728 \le 0) \lor (x \ge 0) \end{aligned}$$

We used Procedure 1 to compute $S_1(x)$ and the following QE problem is considered

$$(\exists u(0))\,(\exists u(1))\,[f_{u(1)} \circ f_{u(0)}(x) = 0]$$

[3]All examples are computed using a DECstation 5000/240 with a 40 MHz R3400 risc-processor.

When the control $u(0)$ is eliminated a polynomial of degree 42 in $u(1)$ and of degree 7 in $x(0)$ is obtained. The same polynomial is obtained when we take the composition of polynomials that define S_0 with f. QEPCAD could compute that the set S_1 consists of all of $\mathbb{R}$ except possibly for 14 algebraic numbers, which are the real roots of some univariate polynomials that were computed. 8 of them have degree 56, 3 have degree 7, 2 have degree 8 and one is rational. In order to obtain this result QEPCAD took 68 minutes of processor time. However, QEPCAD could not complete the computation of $S_1(x)$ after more than 9 hours.

In Chapter 5 we show that for most scalar polynomial systems we could decide on dead-beat controllability after computing the set S_0 only, which took only 1.2 seconds to compute. This shows that instead of using straight forward computation of all S_k's, that is proposed in TESTS 1 and 2, we sometimes may require less information to conclude on dead-beat controllability. This strongly supports our claim (see Comment 3.6) that by combining the structural properties of some classes of systems with QEPCAD we can reduce computations drastically and hence feasible controllability tests can be obtained.

In this case it is not too difficult to see that the interval $]-\infty,-3] \subset S_0$. Let us compute which states can be transferred to this interval in one step by considering the QE problem $(\exists)\ [xu^6 + (x+1)\,u^3 - 2u^2 + 3xu + 2x < -3]$. It was computed that this is true for any $x \in \mathbb{R}$. Hence, $S_1{=}\mathbb{R}$. The answer was obtained in 0.333 seconds. Hence, by reformulating the problem of computing S_1 (it is the set of states that can be transferred to the set $]-\infty,-3]$, which is a subset of S_0) dead-beat controllability could be tested using QEPCAD. This approach was described in Comment 3.7. Although this case-by-case approach is not plausible to use in general, for certain classes of systems it may be successfully imbedded in the controllability test.

In particular, scalar polynomial systems and triangular systems of Chapter 9 (Class 1) seem to be suitable for the application of this method since any of the sets S_j is a finite union of intervals. The reformulation of the dead-beat controllability test becomes extremely simple in these cases. Indeed, we can choose only one interval, which is a subset of S_k, to compute the set S_{k+1}. Note that we do not need to resort to QEPCAD when choosing the interval and this enhances the practicality of the described method for these systems. □

Example 3.2 Consider the generalised Hammerstein system:

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & -2 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u^2(k) \tag{3.9}$$

By using Procedure 1 to compute $S_j(x)$ we obtain:

$$\begin{aligned} S_0(x) &= (x_2 - x_1^2{=}0) \ \wedge (x_3{=}0) \\ S_1(x) &= (2x_3 + x_2 \geq 0) \ \wedge (2x_2x_3 + x_3^2 - 6x_1^2x_3 + x_2^2 - 2x_2x_1^2 + x_1^4{=}0) \\ S_2(x) &= (0{=}0) \end{aligned}$$

The computation time for $S_0(x)$, $S_1(x)$ and $S_3(x)$ is respectively 0.34 sec, 0.517 sec and 133 sec. The sets S_0 and S_1 are given by

$$\begin{aligned} S_0 &= \{x :(x_2 - x_1^2{=}0) \ \wedge (x_3{=}0)\} \\ S_1 &= \{x :(2x_3 + x_2 \geq 0) \ \wedge (2x_2x_3 + x_3^2 - 6x_1^2x_3 + x_2^2 - 2x_2x_1^2 + x_1^4{=}0)\} \\ S_2 &= \mathbb{R}^3 \end{aligned}$$

A minimum-time state dead-beat (feedback) controller is given below:

$$u(x) = \text{any real root } u \text{ to} \begin{cases} f(x,u) = 0 & \text{,if } x \in S_0 \\ S_0 \circ f_u(x) & \text{,if } x \in S_1 - S_0 \\ S_1 \circ f_u(x) & \text{,if } x \in \mathbb{R}^3 - S_1 \end{cases} \tag{3.10}$$

where

$$\begin{aligned} f(x,u) &= (x_1 + u{=}0) \wedge (-x_2 - 2x_3 + u^2{=}0) \\ S_0 \circ f_u(x) &= (x_3 - (x_1+u)^2) {=}0) \wedge (-x_2 - 2x_3 + u^2{=}0) \\ S_1 \circ f_u(x) &= (2(-x_2 - 2x_3 + u^2) + x_3 \geq 0) \wedge (2x_3(-x_2 - 2x_3 + u^2) + (-x_2 - 2x_3 + \\ &\quad + u^2)^2 - 6(x_1+u)^2(-x_2 - 2x_3 + u^2) + x_3^2 - 2x_3(x_1+u)^2 + (x_1+u)^4 {=}0) \end{aligned}$$

The control u is obtained as a real solution to different sets of polynomial equations for $x \in S_1$. On the other hand, a polynomial equation and an inequality should be solved for $x \in \mathbb{R}^3 - S_1$. We can first solve the equation and then check which solutions satisfy the inequality. Since we may have non-unique solutions, the above given minimum-time controller actually represents a family of minimum-time dead-beat control laws. By specifying the rule according to which we choose a solution, different minimum-time state dead-beat controllers are obtained. □

Example 3.3 Consider the third order bilinear systems:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_3(k) \\ x_3(k+1) &= x_1(k) + x_3(k) - (x_1(k) + x_2(k) - x_3(k)\,)\,u(k) \end{aligned} \tag{3.11}$$

The computed $S_j(x)$ are given below:

$$\begin{aligned} S_0(x) &= (x_2{=}0) \,\wedge\, (x_3{=}0) \\ S_1(x) &= (x_3{=}0) \,\wedge\, (x_3 - x_2 - x_1{=}0) \\ S_2(x) &= (x_3 - x_2 - x_1{\neq}0) \,\wedge\, (x_3 + x_2{\neq}0) \\ S_3(x) &= ((x_2 - x_1{\neq}0) \,\wedge\, (2x_3 + x_1{\neq}0)\,) \,\vee\, (x_3 - x_2 - x_1{\neq}0) \\ S_4(x) &= (0{=}0) \end{aligned} \tag{3.12}$$

And hence the system is state dead-beat controllable. Notice that we could conclude on dead-beat controllability using the results in [48, 71] but for the first time we could obtain explicit description of the sets S_k. All of the sets S_k were computed in just a few seconds.

An interesting phenomenon occurs in this example. Namely, the set S_2 consists of the whole state space except for two planes. The union of the two planes is an algebraic variety defined by the polynomial $(x_3 - x_2 - x_1)\,(x_3 + x_2)$. Obviously, the variety consists of all *critical states* that may not be controllable to the origin. In the next chapter we present an approach based on the Gröbner basis method which may be used to check state dead-beat controllability of this class of systems. In Chapter 9, we present (Class 3) systems, which have the same property that the set S_{n-1} is a complement of a real variety. These systems are in principle easier to deal with than the general polynomial systems (3.1). □

Example 3.4 Check whether the system:

$$\begin{aligned} x_1(k+1) &= x_2(k) + u(k) \\ x_2(k+1) &= -x_1(k) + u^2(k) \\ y(k) &= x_2(k) \end{aligned} \tag{3.13}$$

is output dead-beat controllable.

PART 1: The first step is to find the target set T. We denote $T_0{=}V(h){=}\{x : x_2{=}0\}$. Then we compute the set $T_1,\ T_1 \subseteq T_0$ of states that can be mapped back to T_0 in one step. We can easily find $T_1{=}\{x : x_2{=}0 \ \wedge \ x_1 \geq 0\}$ and hence $T_0 \subset T_1$, which means that we need to compute T_2. We have that $T_2{=}\{x : x_2{=}0 \ \wedge \ x_1 \geq 0\}$ and hence $T_2{=}T_1{=}T$.

PART 2: Let us find the set of states in $\mathbb{R}^2$ that can be transferred to T in one step:

$$S_1^O \ = \ \{x : \exists u \in \mathbb{R} \text{ such that } x_2 + u \geq 0,\ -x_1 + u^2{=}0\}{=}\{x : x_2^2 \geq x_1\}$$

Similarly, we have that

$$S_1^O \ = \ \{x : \exists u \in \mathbb{R} \text{ such that } (-x_1 + u^2)^2 \geq x_2 + u\}{=}\mathbb{R}^2$$

and therefore the system is output dead-beat controllable.

Suppose that an output dead-beat controller has been implemented and let us consider what happens once we have have reached the target set. We need to consider the zero output constrained dynamics ($x_2(k){=}0, \forall k$), which are shortly called *zero dynamics*:

$$\begin{aligned} x_1(k+1) &= 0 + u(k) \\ 0 &= -x_1(k) + u^2(k) \\ 0 &= x_2(k) \end{aligned}$$

Straightforward calculations show that the control signal must satisfy:

$$u(k+1){=}+\sqrt{u(k)},\, u(0) \geq 0$$

Simple considerations show that there are two equilibria u=0 and u=1. The equilibrium u=1 is globally asymptotically stable on the interval $u \in]\,0, +\infty[$. The stability of zero output constrained dynamics, which is also called zero dynamics, is crucial for the implementation of output dead-beat control laws. If the zero dynamics are not stable, no output dead-beat controller can be implemented since controls grow unbounded. In Chapter 11 we present a methodology based on the use of QEPCAD which can be used to check when the zero dynamics are stable for systems (3.1). □

3.7 Conclusion

We presented state/output dead-beat controllability tests for a very general class of polynomial systems, which are based on QEPCAD. Furthermore, the methods can be used for dead-beat controllability/control problems of polynomial systems with bounds on controls and states, as well as MIMO systems. Moreover, implicit polynomial systems can be tackled in the same way. We use symbolic computation software in a systematic design of minimum-time dead-beat controllers.

Computational complexity of the dead-beat problems may indeed be formidable. Consequently, it is necessary to constrain the structure of general polynomial systems in order to reduce the computations. Nevertheless, the method that we propose appears to be applicable to the most general class of polynomial systems that is available in the literature. It can be regarded as a unified approach to dead-beat controllability of polynomial systems. In the sequel, we show how it is possible to exploit the structure of subclasses of polynomial systems in order to obtain easier-to-check controllability tests and/or simpler dead-beat controllers. We emphasize that a trade off between the generality and feasibility of the proposed methods forces us to investigate simpler systems in order to reduce the required computations.

Chapter 4

Odd Polynomial Systems

4.1 Introduction

The methodology in Chapter 3 gives a unified approach to the problem of state/output dead-beat controllability for a large class of polynomial systems. However, the computational requirements may be formidable. If the structure of general polynomial systems is constrained, we may obtain computationally less expensive controllability tests using the same methodology. The purpose of this and the following chapters is to exhibit some situations where this is possible. Linear systems are a good example of how the general dead-beat controllability tests presented in the previous chapter can be simplified to reduce computations and obtain explicit controllability tests. In the subsequent chapters, we illustrate the tradeoff between the generality of the proposed methods and the computational resources required using several classes of polynomial systems.

In this chapter, we investigate a class of discrete-time nonlinear systems which allow both a state space and output representation in a polynomial format. More precisely, we consider the class of polynomial systems:

$$x(k+1) = f(x(k), u(k)), \quad y(k) = h(x(k))$$

where $x(k)$, $y(k)$ and $u(k)$ are respectively state, output and input of the system at time k. f and h are polynomials in all their arguments and we assume *inter alia* that the highest exponent of the control u in the polynomial $h(f(x, u))$ is an odd integer.

We concentrate on the output dead-beat controllability properties for this class of polynomial systems. We demonstrate that it is possible to use a combination of the Gröbner basis method in conjunction with QEPCAD to test for output dead-beat controllability. The emphasis is on the existence of the so called invariant sets. The existence of a special class of invariant sets in the context of state controllability for a class of bilinear systems was first considered in [70]. In this paper, the invariant sets were referred to as *trajectories insensitive to control*. Our notion of invariant sets is more general than the one considered in [70], but retains its flavour.

The method that we use illustrates how it is possible to determine the union of all invariant sets, which is crucial for output dead-beat controllability. The controllability tests of this chapter still may suffer from computational inefficiency but non-trivial problems can be solved using symbolic manipulation software packages, as e.g. Maple and QEPCAD. The idea of using the Gröbner basis method in simplifying some problems in first order theory of real closed fields can be found in [80] and approach taken in this chapter goes along the same lines. We also present a number of easier-to-check necessary conditions and sufficient conditions for output dead-beat controllability.

Some results, definitions and notation from algebraic geometry, which we use in this chapter, are given in Appendix B.

4.2 *Definition of the System*

The systems that we consider are given by:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= h(x(k)) \end{aligned} \tag{4.1}$$

where $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}$ and $u(k) \in \mathbb{R}$ are respectively the state, the output and the input of the system (4.1) at time k. The vector $f(x,u) = (f_1(x,u) \ldots f_n(x,u))^T$ is such that $f_i(x,u) \in \mathbb{Q}[x,u] = \mathbb{Q}[x_1, x_2, \ldots, x_n, u]$ and $h \in \mathbb{Q}[x_1, \ldots, x_n]$. Consider the composition

$$h \circ f_u(x) = h(f(x,u)) = a_m(x)\, u^m + \ldots + a_0(x) \tag{4.2}$$

Assumption 4.1 The integer m in the equation (4.2) is odd, that is $m = 2s+1$, $s \in \mathbb{N}$. Systems of this form will be termed odd systems. □

By using Assumption 4.1 we restrict our consideration to systems whose output $y(k+1)$ is affected by $u(k)$ (we have one time delay from input to output). We emphasize that a generalisation of our results to systems of arbitrary time delay is straightforward. Notice that the output is equal to zero for states that belong to the real variety $V(h)$ (for the definition of real varieties see Appendix B).

Assumption 4.2 $\forall x \in V(h)$, $\exists u \in \mathbb{R}$ such that $h \circ f_u(x)$ =0. □

Assumption 4.2 is technical and there are systems of interest that do not satisfy it. However, it simplifies considerably the consideration of output dead-beat controllability. It implies that the target set T (see the previous chapter) is T=$V(h)$. Assumption 4.2 may be very restrictive for some classes of polynomial systems, such as bilinear homogeneous systems. However, it is very often satisfied for odd polynomial systems found in applications. Indeed, we have found in the literature the following odd systems for which Assumption 4.2 is satisfied: an industrial diesel generator [23], a fan and radiator system [21], a grain dryer [107], a heat exchanger [75] and a liquid level system [24]. An odd system for which Assumption 4.2 does not hold is the model of the effects of a drug on the blood pressure of a dog, which was considered in [42].

Notice that the variety $V(a_m)$ contains initial states from which it may not be possible to zero the output. The highest degree m of control u in the expression (4.2) is odd for all states $x \notin V(a_m)$. Hence, for all states outside the variety $V(a_m)$ there exists at least one real value of the control u which renders the expression (4.2) zero. This motivates the following definition.

Definition 4.1 The target set T=$V(h)$ is denoted in this chapter as V_O and is called the *zero output variety*. V_C=$V(a_m)$ is called the *critical variety*. □

Definition 4.2 A set $V_{I_j} \subseteq V_C$ is termed *invariant* if it is such that $\forall x \in V_{I_j}$, $\forall u \in \mathbb{R}$ we have $f(x,u) \in V_{I_j}$. The set V_I is called the *maximal invariant set* if it has the following property: if V_I is a subset of another invariant set V_{I_j} then V_I=V_{I_j}. □

Notice that the maximal invariant set can be viewed as a union of all invariant subsets, that is V_I=$\cup_j V_{I_j}$. Given a set of polynomials $f_1, f_2, \ldots, f_n \in \mathbb{Q}[x_1, \ldots, x_2]$ we denote their reduced Gröbner basis as Gbasis$[f_1, f_2, \ldots, f_n]$ (see Appendix B).

4.3 Invariant Sets and Output Dead-Beat Controllability

In this section we show how it is possible to determine invariant sets of V_C using the Gröbner basis method and how this information can be used to decide on output dead-beat controllability of odd polynomial systems. As we have already indicated, the set of states from which it may not be possible to zero the output is contained in the critical variety V_C. The fact that V_C is a lower dimensional subset of the state space, simplifies the analysis of odd systems considerably.

It is not difficult to show that the critical variety may contain invariant subsets, that is for some states in V_C there may not exist a control sequence $u(0)\,,u(1)\,,\ldots$ which can transfer them to the complement of V_C. The following theorem shows how $V_I \subseteq V_C$ may be computed. Notice that the following compositions can be regarded as polynomials in $u(0)\,,\ldots,u(k)$ whose coefficients are polynomials in x:

$$\begin{aligned}
a_m \circ f_{u(0)} &= \sum_{i=0}^{m_1} b^1_{i_1}(x)\, u(0)^i \\
a_m \circ f_{u(1)} \circ f_{u(0)} &= \sum_{i_1=0,i_2=0}^{m_2,p_2} b^2_{i_1,i_2}(x)\, u(0)^{i_1}\, u(1)^{i_2} \\
\ldots & \quad \ldots \\
a_m \circ f_{u(k)} \circ \ldots \circ f_{u(0)} &= \sum_{i_1=0,\ldots,i_{k+1}=0}^{m_{k+1},p_{k+1},\ldots,l_{k+1}} b^{k+1}_{i_1,\ldots,i_{k+1}}(x)\, u(0)^{i_1} \ldots u(k)^{i_{k+1}}
\end{aligned} \tag{4.3}$$

where $b^1_{i_1}(x)\,, b^2_{i_1,i_2}(x)\,,\ldots, b^{k+1}_{i_1,\ldots,i_{k+1}}(x) \in \mathbb{Q}[x]$.

Theorem 4.1 *The maximal invariant set $V_I \subseteq V_C$ can be computed by an algorithm that stops in finite time.* $\square$

Proof of Theorem 4.1: Notice that by definition $V_I \subseteq V_C$. The set of all critical states is defined by the ideal $I_1 = \langle a_m \rangle$. Consider now the initial states that are in V_C and which are mapped to V_C in one step irrespective of the applied control $u(0)$. These states are characterised by $a_m \circ f_{u(0)}(x) = 0, \forall u(0) \in \mathbb{R}$. The composition of two polynomials is a polynomial and therefore we have $a_m \circ f_{u(0)}(x) = b^1_{m_1}(x)\, u(0)^{m_1} + \ldots + b^1_1(x)\, u(0) + b^1_0(x)$. This polynomial is identically equal to zero for all $u(0)$ if and only if $b^1_{i_1}(x) = 0, \forall i_1 = 0, 1, \ldots, m_1$. Therefore, the points that are mapped to V_C in the first step, regardless of the control action taken, are defined by the ideal $I_2 = \langle a_m, b^1_{m_1}, \ldots, b^1_0 \rangle$. Notice that $I_1 \subseteq I_2$. If $I_1 = I_2$, the critical variety is equal to the maximal invariant set, that is $V_C = V_I$ and the ideal I_1 defines V_I. Suppose that $I_1 \subset I_2$.

Consider now the set of initial states that are mapped in the first and second steps to V_C irrespective of controls $u(0)$ and $u(1)$. The composition $a_m \circ f_{u(1)} \circ f_{u(0)}(x) = b^2_{m_2p_2}(x)\, u(0)^{m_2}\, u(1)^{p_2} + \ldots + b^2_{00}(x)$ is a polynomial in all its arguments and is identically equal to zero $\forall u(0)\,, u(1) \in \mathbb{R}$

if and only if $b_{ij}^2(x)=0, \forall i=0,\ldots,m_2,\ j=0,\ldots,p_2$. Therefore, we have the ideal:

$$I_3=\langle a_n, b_{m_1}^1,\ldots,b_0^1, b_{00}^2,\ldots,b_{m_2p_2}^2\rangle$$

which defines the set of states that stay after two steps inside V_C irrespective of the applied sequence $u(0),\ u(1)$. Observe that $I_2 \subseteq I_3$. If $I_2=I_3$, the maximal invariant set is defined by I_2. If we suppose that $I_2 \subset I_3$, we have that $I_1 \subset I_2 \subset I_3$. Continuing the same construction of ideals $I_1, I_2, I_3,\ldots$ we obtain an ascending chain of ideals which has to stabilise after a finite number of steps. Therefore, we have $I_N=I_{N+1}=\ldots$ and I_N defines the maximal invariant set V_I. Q.E.D.

Notice that the above given proof is constructive in its nature since we form an ascending chain of ideals, which necessarily terminates. All ideals I_j in the chain are determined by polynomials given in (4.3). However, the question arises of how we can compare whether two ideals I_{k-1} and I_k are the same. The Gröbner basis method gives us the tool to do this (see Appendix B). Notice that for a given monomial ordering an ideal may have many Gröbner bases. However, there is a special Gröbner basis which is termed *reduced* and which is *unique* for a given monomial ordering. Therefore, we can compare whether two ideals are the same by comparing whether their reduced Gröbner bases are the same for a chosen monomial ordering. A more explicit algorithm for the computation of the maximal invariant set is presented below.

Theorem 4.2 *The maximal invariant set $V_I \subseteq V_C$ can be computed by the following finite algorithm.*

1. *Initialise: $a_m(x),\ f(x,u)$; $G_0=\{a_m\}$; $k=0$; Fix a monomial ordering.*
2. *Iterate: $k=k+1$*
3. *Compute $a_m \circ f_{u(k-1)} \circ \ldots \circ f_{u(0)}(x)$.*
4. *Compute the reduced Gröbner basis G_k:*

$$G_k=\textit{Gbasis}[a_m, b_0^1,\ldots,b_{m_1}^1, b_{00}^2,\ldots,b_{m_2,p_2}^2,\ldots,b_{m_k,p_k,\ldots,l_k}^k]$$

   where the polynomials $b_{i_1,\ldots,i_s}^s \in \mathbb{Q}[x], s=1,\ldots,k$ are defined in (4.3).

5. *If $G_k=G_{k-1}$ stop. $\langle G_k\rangle$ defines the maximal invariant set V_I. If $G_k\neq G_{k-1}$ go to 2.* □

Proof of Theorem 4.2: Two sets of polynomials define the same ideal if and only if their reduced Gröbner basis is the same [37]. In step k we need to compute the reduced Gröbner basis G_k of I_k (see the proof of Theorem 4.1) and compare it with the reduced Gröbner basis of the ideal I_{k-1} in the previous step G_{k-1}. From Theorem 4.1 we know that any chain of ideals necessarily has got finite length, say N. Hence, it is necessary to compute a reduced Gröbner basis finitely many times.

A reduced Gröbner basis of any set of polynomials can be computed in finite time [37, pg. 89]. Since points 4 and 5 of the above given algorithm compute the reduced Gröbner basis of a set of polynomials, we conclude that the algorithm stated in Theorem 4.2 terminates after a finite number of iterations. Q.E.D.

Comment 4.1 We emphasize that the algorithm in Theorem 4.1 can be used to find an invariant set of any variety defined by $V(f_1,\ldots,f_c)$, $f_i\in\mathbb{Q}[x_1,\ldots,x_n]$, which we denote as $V_I(f_1,\ldots,f_c)$. Notice that the dimension of the variety $V(f_1,\ldots,f_c)$ may be arbitrary, that is $\dim V(f_1,\ldots,f_c)\in\{0,1,\ldots,n\}$. For instance, if $f_1\equiv 0$ trivial calculations show that $V_I(f_1)=V(0)=\mathbb{R}^n$ is invariant. However, in this chapter we are interested only in the invariant subsets of V_C since they can be used to characterise output dead-beat controllability of odd systems with Assumption 4.2. $\square$

The maximal invariant sets of varieties $V(a_m),\ldots,V(a_m,\ldots,a_1)$ are respectively denoted as

$$V_I(a_m),\ldots,\ V_I(a_m,\ldots,a_1)\,.$$

Hence, by definition $V_I=V_I(a_m)$.

The proof of Theorem 4.1 displays several important aspects which we emphasize and summarise below:

1. The algorithm for computing V_I, that is I_N, can be implemented in Maple

2. V_I is a variety, whose dimension is less than n

3. The same method can be used to construct invariant subsets $V_I(a_m,a_{m-1},a_{m-2}),\ldots$, $V_I(a_m,\ldots,a_1)$ of the varieties $V(a_m,a_{m-1},a_{m-2}),\ldots$, $V(a_m,\ldots,a_1)$ and therefore we can find a number of invariant sets V_{I_j} which are contained in V_I.

4. Observe the nested structure:

$$V_I(a_m,\ldots,a_1) \subseteq \ldots \subseteq V_I(a_m,a_{m-1},a_{m-2}) \subseteq V_I(a_m) = V_I$$

   Since all of these sets are invariant, they need to intersect V_O or output controllability is not possible. Hence, we can reduce computations by ordering our calculations in such a way that we compute $V_I(a_m,\ldots,a_1)$ first. The computations may be reduced considerably because the variety $V_I(a_m,\ldots,a_1)$ is computed using more polynomials than V_I and hence we expect to have a shorter chain.

We introduce the following definition:

Definition 4.3 The *trivial invariant set* $V_T \subseteq V_I$ is such that for any initial state $x(0) \in V_T$ there exists a finite sequence of controls which transfers the initial state $x(0)$ to the zero output variety V_O in finite time. □

The trivial invariant set V_T and the maximal invariant set V_I determine output dead-beat controllability of odd systems for which Assumption 4.2 holds. The following theorem follows directly from the definitions of trivial and invariant sets:

Theorem 4.3 *An odd polynomial system (4.1), which satisfies Assumption 4.2, is output dead-beat controllable if and only if* $V_I = V_T$. □

Comment 4.2 The trivial invariant set can be computed using the QEPCAD. Suppose that the maximal invariant set is not empty and that $V_I = V(f_1, f_2, \ldots, f_s)$. Notice that the states that belong to the variety $V_I \cap V_O = V(h, f_1, f_2, \ldots, f_s)$ are already in V_T and we denote this set as S_0^T. Also, we write $S_0^T(x)$ to denote the expression:

$$h(x) = 0,\ f_1(x) = 0, \ldots, f_s(x) = 0$$

We can compute using QEPCAD (see Chapter 3) the subset of V_I from which we can reach the zero output variety in one step:

$$\begin{aligned} S_1^T \quad = \quad & \{x \in \mathbb{R}^n : \exists u \in \mathbb{R},\ h=0, f_1=0, \ldots, f_s=0, \\ & h \circ f_u(x) = 0, f_1 \circ f_u(x) = 0, \ldots, f_s \circ f_u(x) = 0\} \end{aligned}$$

and denote the obtained expression as $S_1^T(x)$. We can continue computing the sets $S_k^T, k{=}1,2,\ldots$ and if we have that $S_k^T{=}S_{k+1}^T$ for some k then the trivial invariant set is $V_T{=}S_k^T$. Notice, that we have $S_k^T \subseteq S_{k+1}^T$ and that the chain of sets $S_0^T \subset S_1^T \subset \ldots$ may not terminate, in which case we can not compute V_T.

The expressions $S_k^T(x)$ that define sets S_k^T are obtained using QEPCAD by considering the QE problems:

$$(\exists u)\ [S_{k-1}^T \circ f_u(x)\,]$$

and checking whether $S_k^T{=}S_{k-1}^T$ is done by considering the QE problems:

$$(\exists u)\ [S_{k-1}^T(x) \wedge \neg S_k^T(x)\,]\ .$$

Notice that although this procedure is almost identical to the one in Chapter 3, we reduced our consideration to a subset of V_C, which reduces computations. □

Below we present an output dead-beat controllability test for odd polynomial systems.

TEST:

1. Check whether Assumptions 4.1 and 4.2 are satisfied. Assumption 4.2 is checked using QEPCAD by considering the decision problem:

   $$(\exists u)\ [h(x)=0 \rightarrow h \circ f_u(x)=0]$$

   If the above given decision problem is true, Assumption 4.2 is satisfied and vice versa.

2. Compute defining equations for V_I using the procedure presented in Theorem 4.2.

3. If $V_I{=}\emptyset$ the system is output dead-beat controllable. If not, go to step 4.

4. Find the trivial invariant set V_T using QEPCAD (see Comment 4.2). If $V_T{=}V_I$, the system is output dead-beat controllable. If $V_T{\neq}V_I$ the system is not output dead-beat controllable.

Comment 4.3 It was shown in Chapter 3 (Example 3.2) that we may use the same approach when tackling state dead-beat controllability. It may happen that the set S_k for some number k is the complement of a real variety whose dimension is lower than the dimension of the state space. If we denote this variety as the critical one and apply the same procedure, we might find all its invariant sets. Hence, we can combine QEPCAD and the Gröbner basis methods in deciding on dead-beat controllability. The motivation for this is that the Gröbner basis method is computationally less expensive than QEPCAD (see Appendix B). □

Comment 4.4 Notice that checking output dead-beat controllability can be done using QEPCAD without resorting to the Gröbner basis method, as it was shown in Chapter 3. In this case we do not compute the maximal invariant set V_I. However, it appears that the maximal invarinat set is an important object in its own right and it seems to be important to provide a method for its computation. □

Comment 4.5 Step 4 of the above given test is very difficult to check in general, since the set V_T is difficult to compute (we may have a non-terminating procedure due to the infinite length of the the chain of S_k^T's). We need to use QEPCAD with all its deficiencies. We remark that each of the sets S_k^T is finitely computable [33, 34] but in general the trivial invariant set is not finitely computable.

Notice that in Step 1 we also need to use QEPCAD, but in this case the computations are performed only once, which leads to a procedure which always stops after finitely many steps. Moreover, it can be expected for Step 1 (checking Assumption 4.2) that the computational requirements are not prohibitive as not many compositions of polynomials are required and multi-degrees of input polynomials are small. Observe that the number of variables in "initial" polynomials for Step 1 and 4 is $n + 1$.

We emphasize that the class of odd systems is inherently simpler than the systems with rational coefficients considered in the previous section since QEPCAD only need to be used for a much smaller subset of the state space. Indeed, notice that $V_I \subseteq V_C$ and $\dim V_C \leq n - 1$. □

Comment 4.6 We can use the above given procedure to check output dead-beat controllability to any fixed output $y{=}y^*$, $y^*{\neq}0$. The modifications to the controllability test are obvious. For instance, the target set ("zero output variety") is in this case defined as $V_O{=}V(h(x) - y^*)$. □

If $V_I(a_m, \ldots, a_1) = V_I(a_m)$, we can find V_T (if it exists) using the Gröbner basis method. The algorithm is presented below:

1. Let the ideal which defines $V_I(a_m, \ldots, a_0)$ be given by $J_0{=}\langle c_1, c_2, \ldots, c_p\rangle$. Consider the following ideals:

$$J_1{=}\langle c_1, \ldots, c_p, a_0\rangle,$$

$$\ldots,$$

$$J_k = \langle c_1, \ldots, c_p, a_0 \circ f_0 \circ \ldots \circ f_0(x) \rangle$$

where $f_0(x) = f(x, 0)$. The sets of states that can be transferred to V_O in one, two, ...,k steps are given respectively by ideals $J_1,\ J_1 \cdot J_2,\ \ldots,\ J_1 \cdot \ldots \cdot J_k$, where $J_1 \cdot J_2$ represents the product of ideals [37].

2. Find reduced Gröbner bases of J_0 and J_1, compute the varieties that are defined by the Gröbner bases and compare them. If they are the same, the trivial invariant set is equal to the maximal invariant set V_I. If not, proceed to 4.

3. Find the Gröbner basis of $J_1 \cdot J_2$ and J_1, compute the varieties defined by the bases and compare them. If they are the same, we found $V_T = V(J_1)$ and if not continue by computing $J_1 \cdot J_2 \cdot J_3$, etc.

4. If the algorithm does not stop after N steps stop the computation. We have not computed V_T.

The problem with the above given algorithm is that if it does not give an answer after N steps (we determine N), we can not say anything about dead-beat controllability. This is because we form a descending chain of ideals (ascending chain of varieties), which does not have to stabilise after finitely many steps. However, the algorithm often gives an answer after a few steps.

The following two corollaries are direct consequences of Theorem 4.3.

Corollary 4.1 Assume $V_I \neq \emptyset$. The odd system (4.1) with Assumption 4.2 is output dead-beat controllable only if $V_O \cap V_I \neq \emptyset$. □

Corollary 4.2 The odd system (4.1) with Assumption 4.2 is output dead-beat controllable if $V_I = \emptyset$. □

4.4 Examples

In all the examples we used the lexicographic monomial ordering $x_1 \succ \ldots \succ x_n$ (see Appendix B).

Example 4.1 Test output dead-beat controllability of the bilinear system:

$$x(k+1)=\begin{pmatrix} 1 & 0 \\ 1 & -2 \end{pmatrix} x(k)+u(k)\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} x(k)$$

$$y(k)=(1 \quad -1)\, x(k)$$

This system is odd, since:

$$y(k+1)=h\circ f_{u(k)}(x(k))=2x_2(k)+2x_1(k)\,u(k)$$

We now need to check if Assumption 4.2 is satisfied. The zero output variety is given by:

$$V_O=V(h)=\{(x_1,x_2)\in\mathbb{R}^2 : x_1-x_2=0\}$$

Therefore, $\forall x\in V_O$ the control $u=-1$ keeps the output at zero for all future steps. Assumption 4.2 is satisfied. We now check if the critical variety, which is defined by $V_C=\{(x_1,x_2)\in\mathbb{R}^2 : x_1=0\}$, has invariant subsets. We consider the following polynomials:

$$a_m=x_1,\ a_m\circ f_{u(0)}(x)=x_1+(2x_1+x_2)\,u(0)$$

Therefore, $I_0=\langle x_1\rangle$ and using Maple we find the Gröbner basis of the ideal $\langle x_1,x_1,2x_1+x_2\rangle$ is $G_1=\{x_1,x_2\}$. Next, consider

$$a_m\circ f_{u(1)}\circ f_{u(0)}(x)=x_1+(2x_1+x_2)\,u(0)+(3x_1-2x_2)\,u(1)+4(x_1+x_2)\,u(0)\,u(1)$$

and the we find that the Gröbner basis G_2 of the ideal $\langle x_1,2x_1+x_2,3x_1-2x_2,x_1+x_2\rangle$ is $G_2=\{x_1,x_2\}$. It follows that $G_1=G_2$ and $V_I=V(G_1)=\{(0,0)\}$. Since $V_I\subset V_O$, it follows that the system is output dead-beat controllable. □

Example 4.2 Consider the inhomogeneous bilinear system:

$$x(k+1)=\begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} x(k)+u(k)\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} x(k)+\begin{pmatrix} 1 \\ 0 \end{pmatrix} u(k)$$

$$y(k)=(2 \quad 0)\, x(k)$$

Compute $h \circ f_{u(k)}(x(k))$:

$$y(k+1) = 2(x_1(k) - x_2(k)) + (4x_1(k) + 2)\, u(k)$$

Since $V_O = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 = 0\}$, we see that $a_m = 4x_1 + 2 \neq 0$, $\forall x \in V_O$ and therefore Assumption 4.2 is satisfied. Find compositions $a_m \circ f_{u(0)}(x)$ and $a_m \circ f_{u(1)} \circ f_{u(0)}(x)$ and compute Gröbner bases of the corresponding coefficients:

$$\begin{aligned}
a_m \circ f_{u(0)}(x) &= 4(x_1 - x_2) + 2 + (2x_1 + 1)\, u(0) \\
a_m \circ f_{u(1)} \circ f_{u(0)}(x) &= (2x_1 - 4x_2 + 1) + (4x_1 - 2x_2 + 2)\, u(0) + (4x_1 - 4x_2 + 2)\, u(1) \\
&\quad + 2(2x_1 + 1)\, u(0)\, u(1)
\end{aligned}$$

$G_1 = G_2 = \{2x_1 + 1, x_2\}$ and therefore $V_I = \{(-1/2, 0)\}$. Since $V_I \cap V_O = \emptyset$ and $V_I \neq \emptyset$ we conclude that the system is not output dead-beat controllable. $\square$

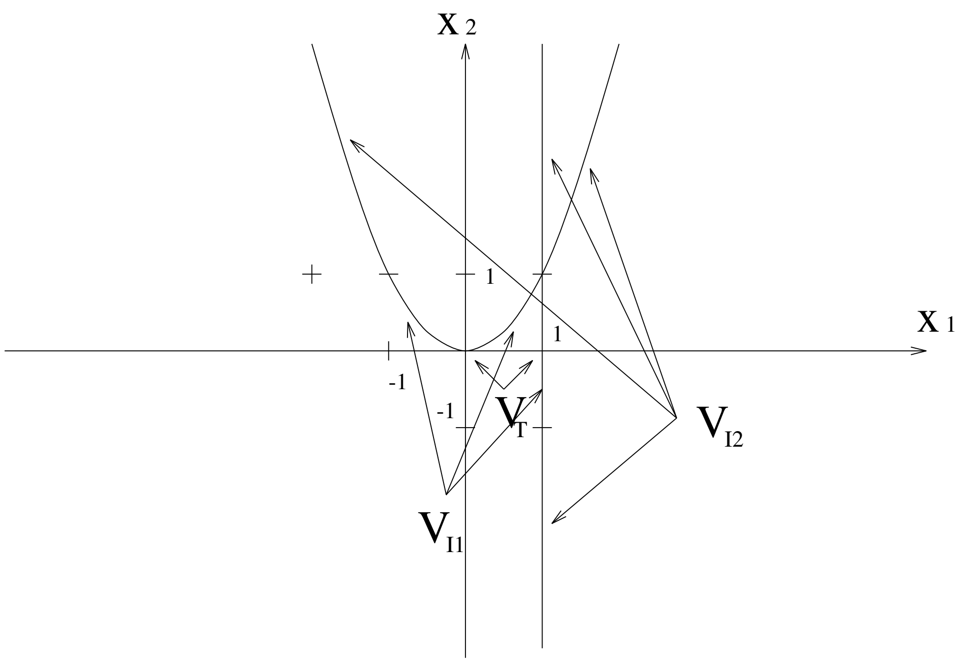

Figure 4.1: Set V_I and asymptotic behaviour invariant to control

Example 4.3 Consider the system:

$$\begin{aligned}
x_1(k+1) &= x_2(k) \qquad (4.4) \\
x_2(k+1) &= x_2^2(k) + (x_1(k) - 1)\,(x_2(k) - x_1^2(k))\, u^2(k) +
\end{aligned}$$

$$(x_1(k)-1)\,(x_2(k)-x_1^2(k)\,)\,u^3(k)$$

$$y(k) \;=\; x_2(k) \tag{4.5}$$

In this case we have:

$$a_3{=}(x_1-1)\,(x_2-x_1^2)$$

$$a_3 \circ f_{u(0)}(x)={(x_2-1)}\,((x_1-1)\,(x_2-x_1^2)\,u^2(0)+(x_1-1)\,(x_2-x_1^2)\,u^3(0)\,)$$

Gröbner basis of $\langle (x_1-1)\,(x_2-x_1^2)\,, (x_2-1)\,(x_1-1)\,(x_2-x_1^2)\,, (x_2-1)\,(x_1-1)\,(x_2-x_1^2)\rangle$ is $G_1{=}\{(x_1-1)\,(x_2-x_1^2)\,\}$ and therefore $V_C{=}V_I$. Trivial invariant set can be found by considering the system on the maximal invariant set V_I.

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_2^2(k) \\ y(k) &= x_2(k) \end{aligned} \tag{4.6}$$

Set of points that are transferred to $V_O{=}\{(x_1, x_2) :\, x_2{=}0\}$ in one step is defined by $\langle x_2^2, (x_1-1)\,(x_2-x_1^2)\,\rangle$ and its Gröbner basis is $G_1{=}\{x_2^2, x_1x_2-x_1^3-x_2+x_1^2\}$. The set of points transferred to V_O in two steps is defined by $\langle x_2^4, (x_1-1)\,(x_2-x_1^2)\,\rangle$ and its Gröbner basis is $\{x_2^4, x_1x_2-x_1^3-x_2+x_1^2\}$. The real varieties defined by G_1 and G_2 are identical. Therefore, the trivial invariant set $V_T{=}\{(0,0)\,,(1,0)\,\}$ and the maximal invariant set is $V_I{=}\{(x_1,x_2) \in \mathbb{R}^2 : (x_1-1)\,(x_2-x_1^2)\,{=}0\}$. The system is not output dead-beat controllable since $V_T{\neq}V_I$.

It can easily be checked that there are two equilibria that are insensitive to control, that is, if the initial state is at an equilibrium we can not get out of it no matter what control is applied to the system. They are $(0,0)$ and $(1,1)$. States $(-1,1)$ and $(1,-1)$ are mapped in one step to $(1,1)$ and then in all future steps is mapped back to $(1,1)$.

There are two invariant subsets that exhibit asymptotic behaviour. Any initial state that belongs to the set $V_{I1}{=}\{(x_1,x_2) \in \mathbb{R}^2 : x_2{=}x_1^2,\ |x_1| < 1,\ x_1{\neq}0\} \cup \{(x_1,x_2) \in \mathbb{R}^2 : x_1{=}1,\ |x_2| < 1,\ x_1{\neq}0\}$ asymptotically converges to the origin. On the other hand, any initial state in $V_{I2}{=}\{(x_1,x_2) \in \mathbb{R}^2 : x_2{=}x_1^2,\ |x_1| > 1\} \cup \{(x_1,x_2) \in \mathbb{R}^2 : x_1{=}1,\ |x_2| > 1\}$ diverges from the origin. However, it is impossible to zero the output in finite time for any initial state in either set V_{I1} or V_{I2}. □

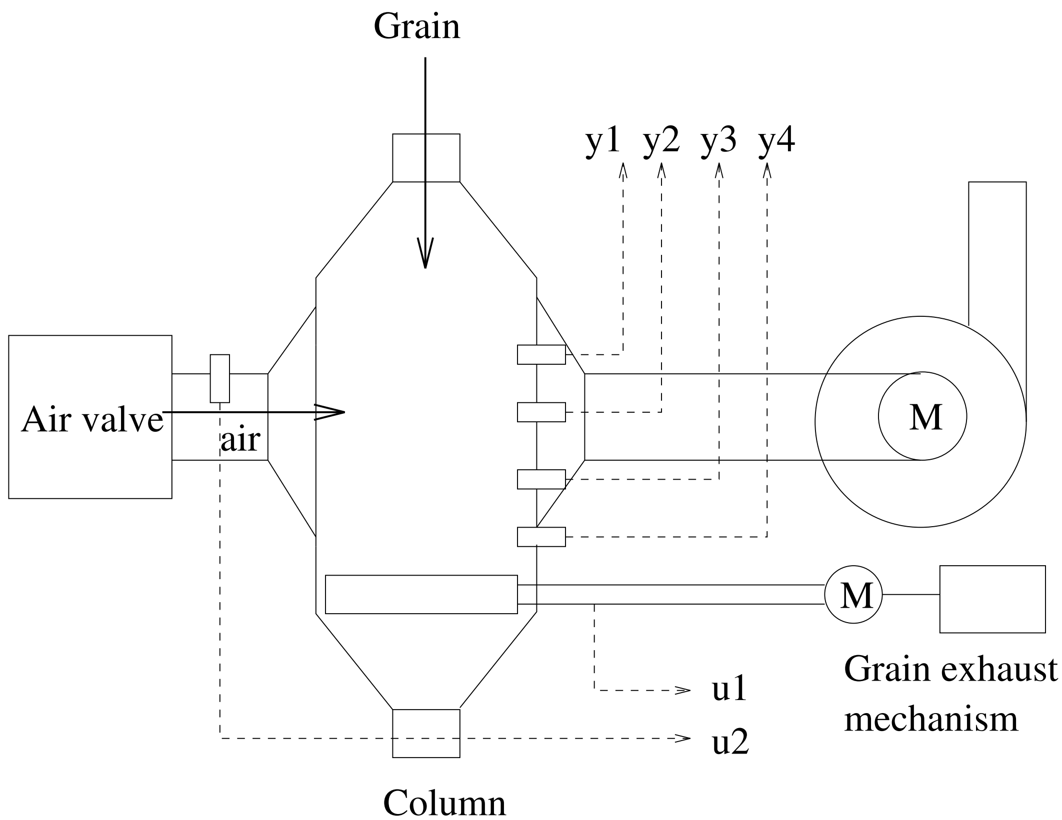

Figure 4.2: A column type grain dryer.

4.5 Case Study 1: Column-Type Grain Dryer

The purpose of grain drying processes is to produce grain (rye, oats, wheat, barley, mixed grain, etc.) with a desired (lower) content of water. We use [89] as a main reference for the features and description of grain drying processes. Usually, a number of grain properties determine its quality. If grain drying is done semi-automatically, which is usually the case, the product quality relies heavily on the experience and qualifications of the operator. Consequently, it often happens that the grain is either over dried or with higher water content than required. The input disturbances, such as the initial humidity of grain, produce large oscillations in the quality of the product and this leads to over-expenditures in energy and wages and a decrease in the drier's productivity. These problems motivate the use of automatic control.

An automatically controlled column type grain dryer is presented in Figure 4.2. The grain is fed into the top part of the column. A fan blows hot air into the column, which dries the grain. Dried grain is extracted from the bottom of the column by means of a grain exhaust mechanism. Two control variables are the productivity of the grain exhaust mechanism u_1 and the temperature of the inlet hot air u_2. The output variables are the humidity z_1 and temperature z_2 of the outlet

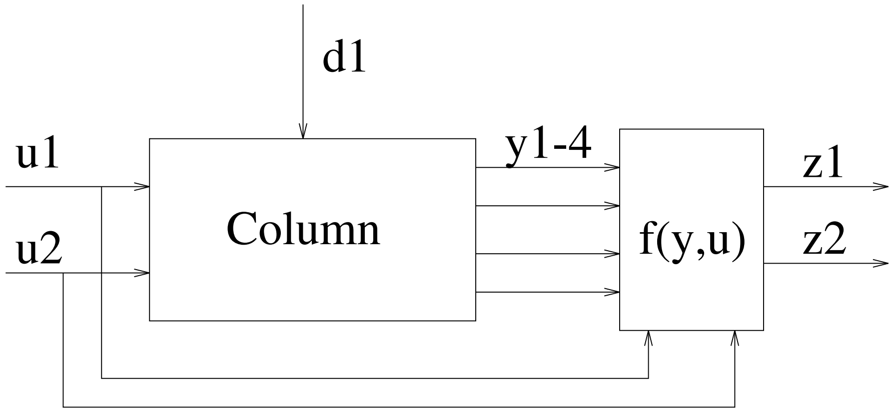

Figure 4.3: A block diagram of the column type grain dryer.

grain but they can not be measured. Hence, we measure the temperatures of the exhaust heating air at the layers $1, 2, \ldots, n$ which are denoted as $y_1, y_2, \ldots, y_n$ (in Figure 4.2 we have four measured temperatures). The block diagram of the system is given in Figure 4.3.

In [89], a controller for the described plant was designed and a functional dependence $Z{=}Z(Y, U)$ investigated. We are, however, not interested in the overall system but just in its subsystem which relates the uppermost temperature $y_1{=}y$ and the productivity of the exhaust grain mechanism $u_1{=}u$. The mathematical model of the subsystem was identified in [107] and is given by:

$$
\begin{aligned}
y(k+1) \;=\; & 1.6389y(k) - 0.4397y(k-1) - 0.1803y(k-2) \\
& - 0.0082u(k)\,y(k) - 0.0042u(k-1)\,y(k-1) - 0.0074u(k-2)\,y(k-2) \\
& + 0.0019u(k) - 0.0041u(k-1) + 0.0021u(k-2)
\end{aligned}
\tag{4.7}
$$

which is called BARMAX[1] (bilinear ARMAX) model [119].

The purpose of this case study is to investigate output dead-beat controllability of this subsystem using the methodology developed in this chapter. For this purpose we introduce state variables:

$$
\begin{aligned}
x_1(k) \;=\; & y(k) \\
x_2(k) \;=\; & -0.4397y(k-1) - 0.1803y(k-2) - 0.0042u(k-1)\,y(k-1) \\
& - 0.0074u(k-2)\,y(k-2) - 0.0041u(k-1) + 0.0021u(k-2)
\end{aligned}
$$

[1] Some authors refer to these models as BARMA.

$$x_3(k) = -0.1803y(k-1) - 0.0074u(k-1)\,y(k-1) + 0.0021u(k-1)$$

and we obtain an inhomogeneous bilinear system:

$$\begin{aligned} x(k+1) &= \begin{pmatrix} 1.6389 & 1 & 0 \\ -0.4397 & 0 & 1 \\ -0.1803 & 0 & 0 \end{pmatrix} x(k) + u(k) \begin{pmatrix} -0.0082 & 0 & 0 \\ -0.0042 & 0 & 0 \\ -0.0074 & 0 & 0 \end{pmatrix} x(k) \\ &\quad + \begin{pmatrix} 0.0019 \\ -0.0041 \\ 0.0021 \end{pmatrix} u(k) \\ y(k) &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x(k) \end{aligned} \tag{4.8}$$

We use the notation:

$$\begin{aligned} x(k+1) &= Ax(k) + u(k)\,Bx(k) + cu(k) \\ y(k) &= dx(k) \end{aligned} \tag{4.9}$$

The first step in the investigation of output dead-beat controllability of the system (4.8) is to check Assumptions 4.1 and 4.2. Consider the expression

$$\begin{aligned} y(k+1) &= d(Ax(k) + u(k)\,Bx(k) + cu(k)) = 1.6389x_1(k) + x_2(k) \\ &\quad + (0.0019 - 0.0082x_1(k))\,u(k)\,. \end{aligned}$$

Assumption 4.1 is satisfied and the system is odd. If we assume that we want to control the output of the system to the point y^*, "zero" output variety is defined as $V_O = V(x_1 - y^*)$ and hence we have that $\forall x(0) \in V_O$ there exists control $u(0)$ which yields $x(1) \in V_O$ if $x_1^* \neq 19/82$. Therefore, Assumption 4.2 is also satisfied for all set points $y^* \in \mathbb{R} - \{19/82\}$ and we can apply the methods from this chapter. Thus, it is assumed that we want to control the temperature y to a set point $y^* \neq 19/82$. We have that $a_m(x) = 0.0019 - 0.0082x_1$.

In order to compute the maximal invariant set, we compute the compositions:

$$a_m \circ f_{u(0)}(x(0)) = 0.0019 - 0.0134x_1(0) - 0.0082x_2(0)$$

$$
\begin{aligned}
& +(\,-1.558\cdot 10^{-5}+6.724\cdot 10^{-5}x_1(0)\,)\,u(0)\,) \\
a_m\circ f_{u(1)}\circ f_{u(0)}(x(0)\,) \;=\; & 0.0019-0.0184x_1(0)\,-0.0134x_2(0)\,-0.0082x_3(0) \\
& +(\,-1.443\cdot 10^{-4}x_1(0)+8.16\cdot 10^{-6})\,u(0) \\
& +(\,-1.558\cdot 10^{-5}+1.102\cdot 10^{-4}x_1(0)+6.72\cdot 10^{-5}x_2(0)\,)\,u(1) \\
& +(1.2776\cdot 10^{-7}-5.514\cdot 10^{-7}x_1(0)\,)\,u(0)\,u(1)
\end{aligned}
$$

Notice that we must scale the coefficients (multiply them with 10^N, where N is the number of decimals that we are working with) in order to use the Gröbner basis method. Hence we have to use the following ideals:

$$
\begin{aligned}
I_0 &= \langle 19-82x_1\rangle \\
I_1 &= \langle 19-82x_1, 19-134x_1-82x_2, 1558-6724x_1\rangle \\
I_2 &= \langle 19-82x_1, 19-134x_1-82x_2, 1558-6724x_1, 19-184x_1 \\
&\quad -134x_2-82x_3, 14430x_1+816, -5514x_1+12776, 1558+11020x_1 \\
&\quad +6720x_2\rangle
\end{aligned}
$$

Their corresponding Gröbner bases with LEX ordering $x_1\succ x_2\succ x_3$ are:

$$
\begin{aligned}
G_0 &= \{19-82x_1\} \\
G_1 &= \{-19+82x_1, 247+1681x_2\} \\
G_2 &= \{1\}
\end{aligned}
$$

Hence, $V_2{=}V(G_2)\,{=}\emptyset$ and as a result $V_I{=}\emptyset$. According to Corollary 4.2 the subsystem (4.8) is output dead-beat controllable, which means that we can achieve any temperature $y^*\in\mathbb{R}-\{19/82\}$ at the uppermost layer of the column by means of the productivity of grain exhaust mechanism $u_1\in\mathbb{R}$.

4.6 Conclusion

We presented an algebro-geometric approach to output dead-beat controllability for a class of odd polynomial systems. The output dead-beat controllability test that we propose is based on the use of the Gröbner basis method and QEPCAD.

Although odd polynomial systems are easier to deal with because the critical variety V_C is a lower dimensional subset of the state space, in general they have the same computational complexity as even system (Assumption 4.2 is not satisfied) as far as deciding dead-beat control is concerned. This is due to the fact that on the critical variety the system may degenerate into an even system and the same phenomena occur. We concentrated our investigation on invariant sets and hence Assumption 4.2. However, not all systems of interest satisfy this Assumption and in general we are limited to use the method presented in Chapter 3.

Chapter 5

Scalar Polynomial Systems

5.1 Introduction

In this chapter we consider one of the simplest possible dead-beat control problems in a specific nonlinear setting. We consider dead-beat control of scalar polynomial systems, described by the difference equation

$$x(k+1) = f(x(k), u(k)) \tag{5.1}$$

where f is a polynomial in all its arguments satisfying

$$f(0,0) = 0 \tag{5.2}$$

$x(k)$ and $u(k)$ are scalars. The state $x(k)$ is assumed to be available for control. The requirement $f(0,0) = 0$ imposes no fundamental restriction on the class of scalar polynomial systems that we consider since it is always possible to achieve it by a suitable change of coordinates (translation of the origin to a new point).

In this chapter we give necessary and sufficient conditions for the existence of dead-beat control of scalar polynomial systems (5.1) and also give sufficient conditions for the existence of local and global stabilising dead-beat control. The conditions for dead-beat controllability are hard to verify. Nevertheless, we present a test, which can be generically implemented using Maple and Matlab. This computer assisted test is able to decide the dead-beat controllability properties of systems (5.1) except for a non generic subset of systems (5.1) which is to be characterised. As an alternative to the presented controllability test, one can use the QEPCAD based tests described in Chapter 3.

The emphasis of the chapter is on the existence of different forms of dead-beat control. We present an algorithm which yields minimum-time control and which can be implemented for a generic class of scalar polynomial systems.

That scalar polynomial systems (5.1) are not completely trivial from a practical point of view may be seen from the simple scalar polynomial model derived for a heat exchanger (see [75]).

5.2 *Notation and Definitions*

We consider discrete, time-invariant, nonlinear controlled systems of the form

$$x(k+1) = f(x(k), u(k)) \tag{5.3}$$

Here $f(x(k), u(k))$ is a polynomial in the variables $x(k)$ and $u(k)$. It may be represented in the following form:

$$f(x(k), u(k)) = a_n(x(k))\, u(k)^n + \ldots + a_1(x(k))\, u(k) + a_0(x(k)) \tag{5.4}$$

where $x(k) \in \mathbb{R}$ is the state at the kth iteration, $u(k) \in \mathbb{R}$ is the control at the kth iteration. The a_i are polynomial functions of x:

$$a_i(x(k)) \quad = \sum_{j=0}^{j=m_i} a_{ij} x(k)^j\,, \tag{5.5}$$

$$a_{ij} \in \mathbb{Q}, \qquad \forall i = 0, 1, \ldots, n, \qquad m_i \in \mathbb{N},\; a_{im_i} \neq 0 \tag{5.6}$$

are polynomial functions.

The following set is introduced:

$$\bar{S} = \{x \in \mathbb{R} : a_i(x) = 0,\; \forall i = 1, 2, \ldots, n\} \tag{5.7}$$

The set $\bar{S}$ is called the control independent set. Observe that the control independent set $\bar{S}$ only contains a finite number of points, the common real roots of a set of polynomials. On the control independent set we can not influence the behaviour of the system with the control signal and the system evolves according to:

$$x(k+1) = a_0(x(k)) \tag{5.8}$$

The importance of the set $\bar{S}$ in the investigation of controllability of a class of bilinear systems was noted in [70].

We now give a list of definitions that are used in this chapter.

Definition 5.1 If n in equation (5.3) is an odd integer, we call the system **odd**. If n is even, the term **even system** is used. □

In this chapter we only consider subsets of $\bar{S}$ when discussing invariance (see Chapter 2). In other words, we have that any invariant set S_I satisfies $S_I \subseteq \bar{S}$. Invariance necessarily implies that the control u can not influence the update. The two following special forms of invariant sets deserve to be defined separately.

Definition 5.2 An equilibrium point of the system (5.3) is a state $\bar{x}$ such that $\forall u \in \mathbb{R}$, $f^i(\bar{x}, u) = \bar{x}$, $\forall i = 0, 1, \ldots$, that is $a_0(\bar{x}) = \bar{x}$. □

Definition 5.3 The point $\hat{x}$ is a periodic point of period p of the system (5.3) [38] if $\forall u \in \mathbb{R}$, $f^p(\hat{x}, u) = \hat{x}$, $f^i(\hat{x}, u) \neq \hat{x}$ for $1 \leq i < p$. In other words, $\hat{x} = a_0^p(\hat{x})$ and $a_0^i(\hat{x}) \neq \hat{x}$, $\forall 1 \leq i < p$. The set $\{a_0^i(\hat{x}),\ i = 1, 2, \ldots, p\}$ is called a periodic orbit. □

All equilibria and periodic points necessarily belong to the set $\bar{S}$.

The following sets are used in the sequel:

$$\mathcal{P}_1 =]-\infty, P_1];\ \mathcal{P}_2 = [P_2, +\infty[,\ P_1, P_2 \in \mathbb{R},\ P_1 < 0,\ P_2 > 0 \tag{5.9}$$

5.3 A Necessary Condition for Dead-Beat Controllability

It is not difficult to show that there may exist several invariant subsets of the control independent set $\bar{S}$, which we denote as S_{I_j}. The union of invariant sets is again an invariant set and we denote the union of all invariant sets $S_{I_j} \subset \bar{S}$ as $S_I = \cup_j S_{I_j}$, which we refer to as the maximal invariant set. In this section we present several important properties of invariant sets S_{I_j} of $\bar{S}$ and give a necessary condition for dead-beat controllability.

Lemma 5.1 *Every invariant set $S_{I_j} \subset \bar{S}$ contains an equilibrium or a periodic orbit.* □

Proof of Lemma 5.1: Consider an invariant set $S_{I_j} \subset \bar{S}$ and suppose that it does not contain either an equilibrium or a periodic orbit. Notice that card $S_{I_j} \leq$ card $\bar{S} < \infty$. Let card $S_{I_j} = L_j$.

Choose $x(0) \in S_{I_j}$. Since S_{I_j} is invariant, $x(0)$ is mapped to S_{I_j}. If $x(0)$ is mapped to $x(0)$ in the first step, then it is mapped to $x(0)$ in all future steps and therefore $x(0)$ is an equilibrium by definition. However, by assumption there are no equilibria and therefore $x(0)$ is mapped to some other point in S_{I_j}. Denote this point as $x(1)$. If $x(1)$ is mapped to $x(0)$ then we have a periodic orbit of period 2 and if it is mapped to $x(1)$ we have an equilibrium. By assumption, therefore, $x(1)$ must be mapped to some other point which we denote $x(2)$. Repeating this argument $L_j - 1$ times it follows that the point $x(L_j - 1)$ must be mapped to an element of S_{I_j} because of its invariance but in this case we have either an equilibrium or a periodic orbit contained in S_{I_j}. The contradiction completes the proof. Q.E.D.

The following two lemmas can be proved using very similar arguments.

Lemma 5.2 *Every initial state in an invariant set S_{I_j} is transferred to the equilibrium (periodic orbit of period p) which belongs to the same set in at most card $S_{I_j} - 1$ (card $S_{I_j} - p$) time steps.* □

Lemma 5.3 *Suppose that card $S_I{=}L$ and card $\bar{S}{=}N$. Then, any initial state in the set $\bar{S} - S_I$ is transferred to $\bar{S}^C$ in at most $N - L$ time steps.* □

Notice that card $S_{I_j} \leq$ card $\bar{S} \leq$ min$[m_i : i{=}1, 2, \ldots, n]$ where m_i are defined in (5.6). An immediate consequence of Lemma 5.3 is that if the invariant maximal set $S_I{=}\emptyset$, then any initial state in $\bar{S}$ is transferred to $\bar{S}^C$ in at most card $\bar{S}{=}N$ time steps. Also, from Lemmas 5.1 and 5.2 it follows that the invariant maximal set S_I of $\bar{S}$ contains finitely many invariant subsets $S_{I_j}, j{=}1, 2, \ldots, T$ which are such that each of them contains only one periodic orbit or one equilibrium.

The following invariant set plays an important role in dead-beat controllability of the system (5.3).

Definition 5.4 The trivial invariant set $S_T \subset \bar{S}$ is an invariant set which contains the origin as its only equilibrium and it does not contain any periodic orbits. □

From Lemma 5.2 it follows that any initial state that belongs to the trivial invariant set is transferred to the origin in finite time and it stays at the origin in all future time steps.

Lemma 5.4 *A necessary condition for the system (5.3) to be dead-beat controllable is that the invariant maximal set $S_I \subset \bar{S}$ is equal to the trivial invariant set S_T.* □

The proof of Lemma 5.4 follows trivially from Lemmas 5.1, 5.2 and 5.3 and Definition 5.2. It is, therefore, necessary for dead-beat controllability that there are no periodic points in the maximal invariant set and the origin is the only allowed equilibrium.

5.4 Odd Systems

In this section we consider odd systems (5.3). These systems have nice properties and their investigation is much simpler than that of even systems.

Lemma 5.5 *The odd system (5.3), is dead-beat controllable if and only if the invariant maximal set S_I is equal to the trivial invariant set S_T.* □

Proof of Lemma 5.5: Necessity is given in Lemma 5.4. Suppose that the maximal invariant set is equal to the trivial invariant set. Since the coefficient $a_n(x)$ is not identically equal to zero and since it is a polynomial, it can have only finitely many real roots. This means that for almost any initial state $x(0)$ the polynomial

$$a_n(x(0)\,)\,u(0)^n + \ldots + a_1(x(0)\,)\,u(0) + a_0(x(0)\,) \tag{5.10}$$

has the highest degree of $u(0)$ odd and therefore has at least one real root. In other words, the set S_0 (see Chapter 2 for definition of sets S_k) is almost the whole state space. Obviously, its compliment S_0^C contains the control independent set, but in general it is not equal to it. We now consider all initial states that are in the set $\mathcal{T}=\mathbb{R}-\{\bar{S}\cup S_0\}$. Consider the situation that $\mathcal{T}$ is not an empty set.

Not all $a_i(x)$, $i=1,2,\ldots,n$ vanish for a fixed $x\in\mathcal{T}$ and therefore we have for any, but fixed $x(0)\in\mathcal{T}$

$$a_\nu(x(0)\,)\,u(0)^\nu + \cdots + a_\eta(x(0)\,)\,u(0)^\eta = f(x(0)\,,u(0)\,)\,,\ \forall x(0)\in\mathcal{T} \tag{5.11}$$

where $0=\eta<\nu\leq n$, $\nu=\nu(x(0)\,)$, $\eta=\eta(x(0)\,)$ and $\nu,\eta\in\mathbb{N}$. Notice that η is necessarily equal to zero on the set $\mathcal{T}$. If $\eta(x(0)\,)$, the state $x(0)$ belongs to S_0, since by applying $u=0$ it is transferred to the origin.

We introduce the following sets:

$$\mathcal{T}_1 = \{x(0)\in\mathcal{T} : \nu(x(0)\,)\ \text{is odd}\} \tag{5.12}$$

$$\mathcal{T}_2 = \{x(0)\in\mathcal{T} : \nu(x(0)\,)\ \text{is even}\} \tag{5.13}$$

$$\mathcal{T}_{21} = \{x(0) \in \mathcal{T}_2 : a_{\nu(x(0))}(x(0)) > 0\} \tag{5.14}$$

$$\mathcal{T}_{22} = \{x(0) \in \mathcal{T}_2 : a_{\nu(x(0))}(x(0)) < 0\} \tag{5.15}$$

Case 1: If $x(0) \in \mathcal{T}_1$ there exists at least one real solution $u(0)$ of the equation (5.11) [169].

Case 2: If $x(0) \in \mathcal{T}_2$ there may or may not be a real solution to (5.11). If there is, we have one step controllability for $x(0)$. If there is no real solution of (5.11), we consider the equation

$$a_\nu(x(0))\, u(0)^\nu + \cdots + a_\eta(x(0))\, u(0)^\eta = K(x(0)),\ K(x(0)) \in S_0 \tag{5.16}$$

If there is a real solution to (5.16), then it is possible to map all the initial states of the set for which there is no one step zeroing into the set S_0. We can therefore map $x(0)$ to the origin in two steps. Since $a_\nu \neq 0$, it follows that it is either positive or negative. Therefore, the set $\mathcal{T}_2$ can be partitioned into $\mathcal{T}_{21}$ and $\mathcal{T}_{22}$.

All the polynomial functions $a_i(x(0))$ are bounded on the set and since $K(x(0)) \in S_0$, it is always possible to chose $K(x(0))$ such that the sign of $a_\nu(x(0))$ is opposite from the sign of $a_0(x(0)) - K(x(0))$. For example, if $x(0) \in \mathcal{T}_{21}$ we can find a large positive number $K(x(0)) \in S_0$ such that $a_0(x(0)) - K(x(0)) < 0$. Then it follows that the equation (5.16) has at least two real solutions $u(0)$; one is positive and another is negative [169, pg. 105]. Similarly, if $x(0) \in \mathcal{T}_{22}$ we can find a negative number $K(x(0)) \in S_0$ to which we can map $x(0)$. Therefore, it is possible to transfer every state $x(0)$ (for ν even and when there is no solution to (5.11)) to the set S_0 (in the first step) and then from S_0 to the origin (the second step). Moreover, observe that any $x(0) \in \mathcal{T}$ can be mapped to one of two states $K_1 \neq K_1(x(0))$ and $K_2 \neq K_2(x(0))$, where $K_1, K_2 \in S_0,\ K_1 < 0,\ K_2 > 0$ and $\max(\max_{x\in\mathcal{T}} |a_0(x)|, \min_{x\in S_0} |x|) \leq |K_i|,\ i=1,2$.

Since the maximal invariant set S_I is equal to the trivial invariant set S_T, all the initial states that belong to $\bar{S} - S_T$ are mapped either to $\mathcal{T}$ or to S_0 and hence can be mapped to the origin in a finite number of steps. Similarly, points in S_T are mapped to the origin in finite time. Q.E.D.

The proof of Lemma 5.5 is equivalent to IF THEN ELSE statements and it is possible to use the proof as a design of a feedback dead-beat controller for dead-beat controllable odd systems. The control law is typically discontinuous, except in some special cases (for example, when $a_n(x) \neq 0, \forall x \in \mathbb{R}$). However, it is continuous on intervals (subsets of state space) which depend on the coefficient polynomials $a_i(x)$. The ensuing control law is shown in Figure 5.1. Due to practical limitations, such as actuator saturations, this control law might not be possible to

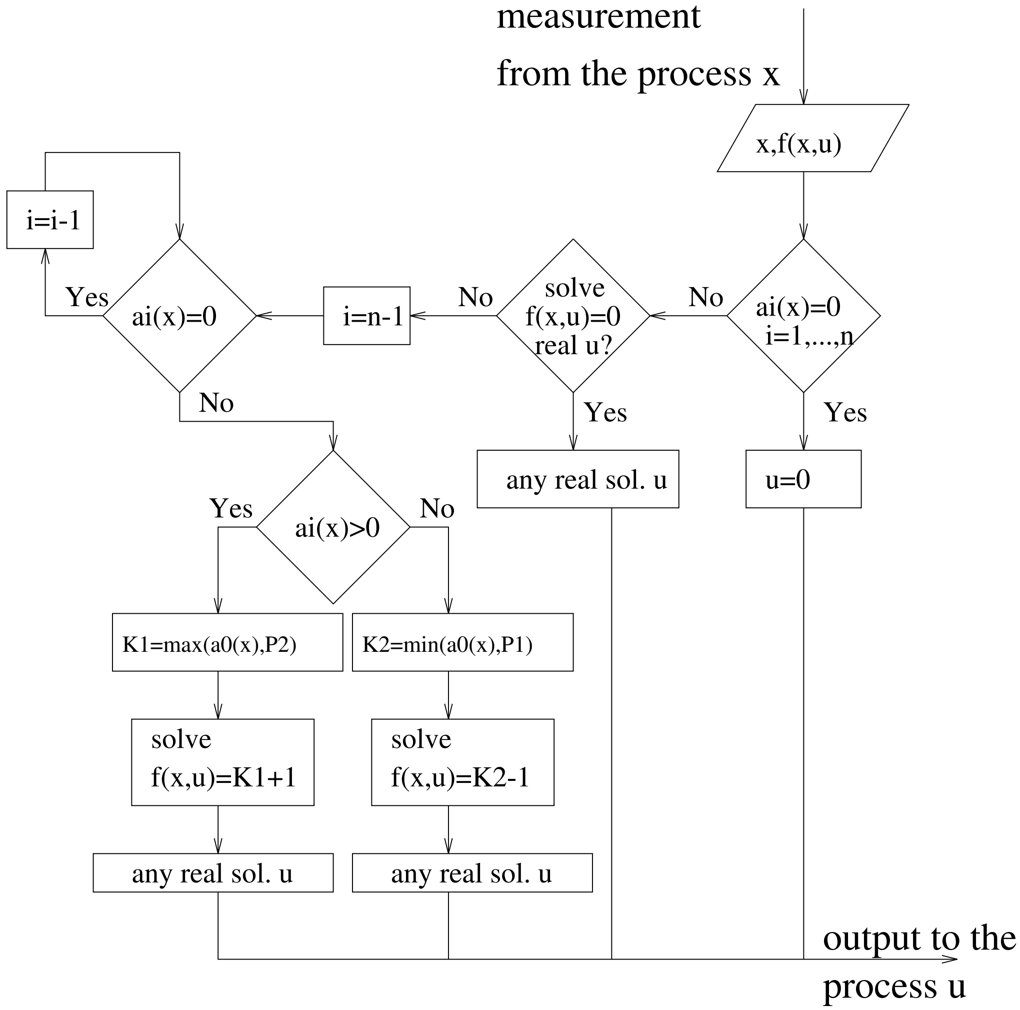

Figure 5.1: Dead-beat controller for odd systems

implement in its original form shown in Figure 5.1. Nevertheless, the algorithmic nature as the control strategy provides a template for the design of a feasible controller. In Lemma 5.5 we do not exclude the possibility of having very large magnitudes of the control signal and it may happen that its supremum as viewed over all states is infinite. It is possible to trade off the magnitude of the control signal and the minimum number of steps required to achieve dead-beat control. For example, for the control law of Figure 5.1 it can be shown that the value of the control signal goes to infinity in the neighbourhood of every point that belongs to the control independent set $\bar{S}$. If we modify our control law so that we apply only bounded control signals on those neighbourhoods, then it may be possible to achieve a non-minimum-time dead-beat control which yields smaller control signals and which is therefore feasible.

The control law shown in Figure 5.1 does not in general yield a good transient state response but it is time-optimal. The values of P_1 and P_2 in Figure 5.1 are pre-computed using Procedure 2, which is presented in the sequel, and the algorithm can be used for Classes 1 and 2 even systems (see the next section).

Corollary 5.1 Let $\bar{S}=\emptyset$. Consider the system (5.3). Let the system be odd. The system (5.3) is 1 or 2 steps dead-beat controllable. □

Proof of Corollary 5.1: Since the polynomial $a_n(x)$ can only have finitely many real zeros, the highest order of control signal may be even for finitely many initial states and odd for the complement of the state space. Thus only finitely many initial states may require two steps zeroing. Q.E.D.

Corollary 5.2 If we consider an odd system and $a_n(x)\neq 0$, $\forall x \in \mathbb{R}$, the system (5.3) is 1-step dead-beat controllable. □

Proof of Corollary 5.2: Since the highest order of control signal is odd on the whole state space, any initial state can be transferred to the origin in one step. Q.E.D.

Corollary 5.3 If the odd system (5.3) is dead-beat controllable then there exists a global stabilising dead-beat control law. □

Proof of Corollary 5.3: Since the set $\{x : a_n(x) = 0\}$ can not be dense in the neighbourhood of the origin, it follows that any initial state from a sufficiently small neighbourhood of the origin can be driven to the origin in the first step and therefore the system is stable. Q.E.D.

5.5 *Even Systems*

In this section we consider even systems (5.3). Since these systems are more difficult to deal with, we split the investigation into several parts. We first consider a class of even systems with properties not too dissimilar from the odd systems. The existence of "a neighbourhood of infinity" ($\mathcal{P}_1 \cup \mathcal{P}_2$) that is controllable to the origin in one control action plays a key role. The second class of even systems does not have similar properties to odd systems since there does not exist a "neighbourhood of infinity" that is controllable to the origin in one step. However, $\mathcal{P}_1$ or $\mathcal{P}_2$ can still be mapped to the origin. This enables us to use very similar methods to the ones used for

Case 1 in solving this case. For Case 3 systems neither interval $\mathcal{P}_1$ or $\mathcal{P}_2$ can be mapped to the origin in one step. This is the most difficult case to analyse. However, this situation is proved to be non generic.

Consider the equation:

$$f(x,u) = 0 \tag{5.17}$$

written in the following format:

$$a_0(x) = -a_1(x)\,u - \ldots - a_n(x)\,u^n \tag{5.18}$$

Define

$$-f_1(x,u) = -a_1(x)\,u - \ldots - a_n(x)\,u^n \tag{5.19}$$

Definition 5.5 The control value set $\mathcal{U}(x)$ at $x \in \mathbb{R}$ is

$$\mathcal{U}(x) = \{y : y = -f_1(x,u) \text{ and } y \in \mathbb{R},\ u \in \mathbb{R}\} \tag{5.20}$$

The control value domain is

$$\mathcal{U} = \bigcup_{x\in\mathbb{R}} (\{x\} \times \mathcal{U}(x)) \subseteq \mathbb{R}^2 \tag{5.21}$$

□

Obviously the control value set can only take on one of the following forms:

$$\begin{aligned}
\mathcal{U}(x) &= \{0\}, \text{ if } x \in \bar{S} \\
\mathcal{U}(x) &= \,]-\infty, L_1] \text{ or } [L_2, +\infty[, \text{ if } x \notin \bar{S} \text{ and the highest degree of } u \text{ in } -f_1(x,u) \text{ is even} \\
\mathcal{U}(x) &= \,]-\infty, +\infty[, \text{ if } x \notin \bar{S} \text{ and the highest degree of } u \text{ in } -f_1(x,u) \text{ is odd}
\end{aligned}$$

Figure 5.2 shows the introduced notation graphically, as well as the concept that is used in the proof of the main result. It can be seen that the set of points controllable to the origin in the first step (S_0) is obtained as the set of x for which the drift term belongs to the control value domain. If we want to find all real roots u of the equation $f(x,u) = K$ then it is necessary to translate the plot of the boundary of the control value domain over a distance K and determine the set of states for which we can a find real solution in the same way. This method may be used for the construction of sets S_k of states that can be transferred to the origin in k steps by taking $K_0 = 0$

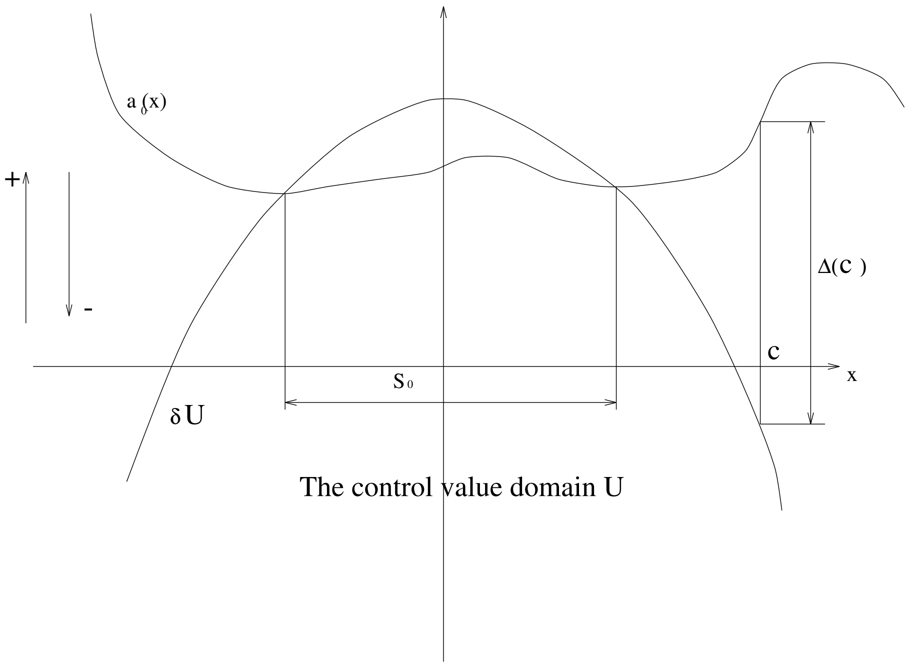

Figure 5.2: Graphical solution to the equation $f(x, u) = 0$

and $K_k \in S_{k-1}$. The arrows in Figure 5.2 indicate positive and negative directions in which it is possible to translate the boundary of the control value domain. Therefore, the mapping of a set of states to a state $K_1 \in S_0$ is equivalent to translating the plot of $\partial\mathcal{U}(x)$ over a distance K_1 and determining the set S_1 in the same manner. For this reason, we shall use terms "mapping from a set to a point $K(k)$" and "translating of the plot $\partial\mathcal{U}(x)$ over a distance $K(k)$" to describe the same thing.

5.5.1 Case 1

Lemma 5.6 *If the maximal invariant set S_I is equal to the trivial invariant set S_T and if there exist sets $\mathcal{P}_1$ and $\mathcal{P}_2$ of the form (5.9) such that*

$$\mathcal{P}_1 \cup \mathcal{P}_2 \subseteq S_0$$

then the even system (5.3) is dead-beat controllable. □

The proof of Lemma 5.6 is similar to the proof of Lemma 5.5.

Case 1 of even systems is very similar to odd systems. The only difference is that the existence of "the neighbourhood of infinity" $\mathcal{P}_1$ and $\mathcal{P}_2$ that is controllable to the origin in one step is not an issue for odd systems. Therefore, in Case 1 of even systems it is necessary to check the existence of suitable $\mathcal{P}_1$ and $\mathcal{P}_2$ that satisfy the conditions of Lemma 5.6. As a consequence of the similarity to odd systems, the controller shown in Figure 5.1 can be used for Case 1 even systems.

5.5.2 Case 2

Lemma 5.7 *If the maximal invariant set S_I is equal to the trivial invariant set S_T and one of the following conditions hold:*

1. $\exists\mathcal{P}_1$ *and* $\exists\mathcal{P}_2$ *such that* $\mathcal{P}_1 \subseteq S_0,\ \mathcal{P}_2 \not\subset S_0$ *and* $a_0(x) < \partial\mathcal{U}(x)\,,\ \forall x \in \mathcal{P}_2$

2. $\exists\mathcal{P}_1$ *and* $\exists\mathcal{P}_2$ *such that* $\mathcal{P}_2 \subseteq S_0,\ \mathcal{P}_1 \not\subset S_0$ *and* $a_0(x) > \partial\mathcal{U}(x)\,,\ \forall x \in \mathcal{P}_1$

the system (5.3) is dead-beat controllable. □

Proof of Lemma 5.7: Since $S_I{=}S_T$ we can concentrate on the points in $\bar{S}^C$. In order to map a state $x(0) \in S_0^C - \bar{S}$ to S_0 we need to translate the boundary of the control value domain in a desired direction over a value K which is such that $a_0(x(0)) \in \mathcal{U}(x(0)) + K$ (notice that f $\mathcal{U}(x(0)) =]-\infty, C]$ then $\mathcal{U}(x(0)) + K \triangleq]-\infty, C + K]$). Consider situation 1 of Lemma 5.7. Let $\mathcal{P}_2 \not\subset S_0$, then in order to map a state in $\mathcal{P}_2$ to $\mathcal{P}_1$, we need to translate $\partial\mathcal{U}$ over a distance K ($K < 0$, and $K \in \mathcal{P}_1$). This follows from the fact that a_0 is below $\mathcal{U}$, $\forall x \in \mathcal{P}_2$. Since we have at our disposal all $K \in \mathcal{P}_1$, we can do this in one step for any $x(0) \in \mathcal{P}_2$. Q.E.D.

Lemma 5.7 can be proved without using the control value domain and in a similar manner as Lemma 5.5. Indeed, notice that the condition $a_0(x) < \partial\mathcal{U}(x)\,,\ \forall x \in \mathcal{P}_2$ means that sign$a_0(x)$ =sign$a_n(x) = -1,\ \forall x \in \mathcal{P}_2$. Since $\mathcal{P}_1 \subseteq S_0$, we see that $\forall x(0) \in \mathcal{P}_2, \exists K(x(0)) \in \mathcal{P}_1$ such that sign$(a_0(x(0)) - K(x(0)))\neq$sign$a_n(x(0))$ and therefore $f(x(0), u) = K(x(0))$ has a real solution. However, the control value domain method is invaluable in proving the main result and the non genericity of Case 3 even systems.

5.5.3 Case 3

This last case contains a class of systems that is the most difficult to deal with. However, we prove that Case 3 systems are not generic. This case completes the classification of scalar polynomial systems and together with the previous two cases gives all the dead-beat controllable scalar polynomial systems.

Lemma 5.8 *If $S_I{=}S_T$ and there exists $k \in \mathbb{N}$ and S_k such that conditions of Lemmas 5.6 or 5.7 are satisfied when S_0 is replaced by S_k, then the even system (5.3) is dead-beat controllable.* □

Proof of Lemma 5.8: Suppose that $S_I{=}S_T$ (see Lemma 5.4). We can, therefore, find a control law which yields a Case 1 or 2 situation after k steps and the proof follows from the proof of Lemma 5.6 or 5.7 . Q.E.D.

5.6 Main Result

Lemmas 5.5, 5.6, 5.7 and 5.8 give a classification of dead-beat controllable scalar polynomial systems and in Theorem 5.1 this is explicitly stated. Before stating the main result, we need to define the distance between the control value domain and the drift term.

$$\Delta(x) = d(a_0(x), \mathcal{U}(x)) \tag{5.22}$$

$$d(a_0(x), \mathcal{U}(x)) = \min_{y \in \mathcal{U}(x)} |a_0(x) - y| \tag{5.23}$$

And its limits

$$\Delta_{+\infty} = \lim_{x \to +\infty} \Delta(x) \in \mathbb{R} \cup \{+\infty\} \tag{5.24}$$

$$\Delta_{-\infty} = \lim_{x \to -\infty} \Delta(x) \in \mathbb{R} \cup \{+\infty\} \tag{5.25}$$

The new variable $\Delta(x)$ represents the distance of the drift term from the control value domain for a given x. For instance, the distance of the drift term $\Delta(C)$ at the point C is shown in Figure 5.2. If the drift term belongs to the control value domain for a given x the distance $\Delta(x)$ is zero. We emphasize (this is shown in the sequel) that due to the underlying polynomial structure the limits in (5.24) and (5.25) always exist and they are equal to either a constant number or to $+\infty$.

Theorem 5.1 *Consider the polynomial scalar system (5.3) for which $S_I{=}S_T$. The system is dead-beat controllable if and only if one of the following conditions hold.*

1. $\exists \mathcal{P}_i,\ i{=}1,2$ *such that* $\mathcal{P}_1 \cup \mathcal{P}_2 \subseteq S_0$
2. $\exists \mathcal{P}_i,\ i{=}1,2$ *such that* $\mathcal{P}_1 \subset S_0,\ \mathcal{P}_2 \not\subset S_0$ *and* $a_0(x) < \partial\mathcal{U},\ \forall x \in \mathcal{P}_2$
3. $\exists \mathcal{P}_i,\ i{=}1,2$ *such that* $\mathcal{P}_2 \subset S_0,\ \mathcal{P}_1 \not\subset S_0$ *and* $a_0(x) > \partial\mathcal{U},\ \forall x \in \mathcal{P}_1$
4. $\Delta_{+\infty}{=}const. < +\infty$ *and/or* $\Delta_{-\infty}{=}const. < +\infty$ *and* $\exists k \in \mathbb{N}$ *and* S_k *such that one of the above given conditions are satisfied when* S_0 *is replaced by* S_k. □

We need to prove several lemmas before giving the proof of Theorem 5.1.

Lemma 5.9 *For every scalar polynomial system there exist $\mathcal{P}_1$ and $\mathcal{P}_2$ such that one of the following situations occur:*

1. $\mathcal{P}_1 \cup \mathcal{P}_2 \subseteq S_0$, *which is denoted as* $(+,+)$
2. $\mathcal{P}_1 \subseteq S_0,\ \mathcal{P}_2 \cap S_0 = \emptyset$, *which is denoted as* $(+,-)$
3. $\mathcal{P}_2 \subseteq S_0,\ \mathcal{P}_1 \cap S_0 = \emptyset$, *which is denoted as* $(-,+)$
4. $\mathcal{P}_1 \cap S_0 = \emptyset,\ \mathcal{P}_2 \cap S_0 = \emptyset$, *which is denoted as* $(-,-)$ $\square$

Proof of Lemma 5.9: We can use the Sturm theorem [87] in order to find the number of real roots of a single variable polynomial on the interval $[a,b]$, including $]-\infty,+\infty[$. Since we are dealing with the two variable polynomial:

$$f(x,u) = 0 \tag{5.26}$$

we can regard $x \in \mathbb{R}$ as a parameter and for any fixed value of x we can find the number of real roots u to (5.26). Using the algorithm, which is a slight modification of the division algorithm [87], we obtain the Sturm sequence:

$$\begin{aligned}
f_0(x,u) &= f(x,u) \\
f_1(x,u) &= \frac{\partial}{\partial u} f(x,u) \\
\cdots &\quad \cdots \\
f_{i+1}(x,u) &= q_i(x,u)\, f_i(x,u) - f_{i-1}(x,u)\,, \deg f_{i+1} < \deg f_i \\
\cdots &\quad \cdots \\
f_{s-1}(x,u) &= q_s(x,u)\, f_s(x,u)\,,\ (\text{that is } f_{s+1}(x,u) = 0)
\end{aligned} \tag{5.27}$$

The leading coefficient functions in the Sturm sequence are denoted as $l_i(x)\,,\ i=0,1,\ldots,s$. They are the functions in x that multiply control u with the highest exponent, in polynomials $f_i(x,u)\,,\ i=0,1,\ldots,s$. It turns out that $l_i(x)$ are rational functions. We denote numerators and denominators of $l_i(x)$ respectively as $\text{num} l_i(x)$ and $\text{den} l_i(x)\,,\ i=0,1,\ldots,s$ and introduce the following sets:

$$m_i = \{x \in \mathbb{R} : \text{num} l_i(x) = 0\}$$

$$D_i = \{x \in \mathbb{R} : \text{den} l_i(x) = 0\} \tag{5.28}$$

Denote:

$$\max_{x \in m_i,\ D_i} |x| = P$$

Therefore, the leading coefficient functions $l_i(x)$ do not change signs on intervals $]-\infty, -P-\varepsilon]$ and $[P+\varepsilon, +\infty[, \forall\varepsilon > 0$ and consequently there are constant numbers of real roots u to (5.26) on these intervals [87]. It is obvious that the conditions of Lemma 5.9 are satisfied when we define $P_1 = -P - \varepsilon$ and $P_2 = P + \varepsilon,\ \varepsilon > 0$ in (5.9). Q.E.D.

One can, therefore, classify all scalar polynomial systems into the four categories given in Lemma 5.9. Lemma 5.9 reflects a special property of scalar polynomial systems and it does not hold in general. For instance, if we consider $x(k+1) = \sin x(k) + u(k)^2$, it is obvious that it does not satisfy the statements of Lemma 5.9.

Lemma 5.10 *If* $\mathcal{P}_i \cap S_0 = \emptyset,\ i = 1, 2$ *then either*

$$a_0(x) > \partial\mathcal{U}(x),\ \forall x \in \mathcal{P}_i \quad or \quad a_0(x) < \partial\mathcal{U}(x),\ \forall x \in \mathcal{P}_i.$$

□

Proof of Lemma 5.10: Notice first that conditions of Lemma 5.10 can hold only for even systems since odd systems always belong to $(+,+)$ category in Lemma 5.9. Suppose that $\mathcal{P}_1 \cap S_0 = \emptyset$ and that there exist subsets R_1 and R_2 of $\mathcal{P}_1$ such that $a_0(x) > \partial\mathcal{U}(x),\ \forall x \in R_1$ and $a_0(x) < \partial\mathcal{U}(x),\ \forall x \in R_2$.

Suppose that $a_0(x) \not\equiv 0$ since $0 \in \mathcal{U}(x),\ \forall x \in \mathbb{R}$ and in this case lemma 5.10 can not be applied (we have $(+,+)$ even system). From the proof of Lemma 5.9 we see that on the set $\mathcal{P}_1$ the coefficient $a_n(x) \neq 0$ (see the Sturm sequence (5.27)). As a result, the control value domain restricted to $\mathcal{P}_1$ has one of the forms $]-\infty, P_1(x)] \times \mathcal{P}_1$ or $[P_2(x), +\infty[\times \mathcal{P}_1$, where functions $P_1(x) \geq 0$ and $P_2(x) \leq 0$ (this is not difficult to see from equations (5.19,5.20,5.21)). Without loss of generality assume that the control value domain has the form $]-\infty, P_1(x)] \times \mathcal{P}_1$ on the set $\mathcal{P}_1$. Consequently, $R_1 \not\subset S_0$ and $R_2 \subset S_0$. But since $R_1,\ R_2 \subset \mathcal{P}_1 \not\subset S_0$, we have a contradiction. Q.E.D.

Lemma 5.11 *If* $\mathcal{P}_1 \not\subset S_0$ *(*$\mathcal{P}_2 \not\subset S_0$*) then either* $\Delta_{-\infty} = const.$ *(*$\Delta_{+\infty} = const.$*) or* $\Delta_{-\infty} = \infty$ *(*$\Delta_{+\infty} = \infty$*).* □

Proof of Lemma 5.11: Denote $\partial U(x)=z$. It is not difficult to see that the boundary of the control value domain z must satisfy the following equations:

$$z-f(x,u)=0 \tag{5.29}$$

and

$$\frac{\partial}{\partial u}f(x,u)=0 \tag{5.30}$$

for all $x\in\mathbb{R}$. Since the conditions of Lemma 5.11 can be satisfied only for even systems ($n\geq 2$, see equation (5.3)), the equation (5.30) is a polynomial in x and u. Using the Gröbner basis method [37] with the lexicographic ordering $u\succ x\succ z$, it is possible to eliminate u from these equations and obtain a polynomial equation:

$$G(x,z)=0 \tag{5.31}$$

which must be satisfied for any real x (for more details on the Gröbner basis method see Appendix B). Notice that there exists in general more than one solution z to the equation (5.31) for a fixed x but just one solution corresponds to the boundary of the control value domain. Using the new notation we can write $\Delta^*(x)=z-a_0(x)$ and $\Delta=|\Delta^*(x)|$. Therefore, we have that:

$$G(x,\Delta^*(x)+a_0(x))=0 \tag{5.32}$$

So, for any fixed $x\in\mathbb{R}$, the distance of the drift term from the control value domain can be obtained as the absolute value of a real solution to the implicit polynomial equation (5.32). Observe that (5.32) defines an algebraic set (variety) in $\mathbb{R}^2$ which we denote by V_G. We consider now what happens with the roots to (5.32) when $x\to\pm\infty$. Suppose that the limit is neither a constant nor $+\infty$ or $-\infty$. In this case V_G would intersect a horizontal line $\Delta^*=X$ infinitely many times. We decompose the variety V_G into irreducible components $V_G=V_1\cup V_2\cup\ldots\cup V_l$ and since the horizontal line is also an irreducible variety, according to Bezout's theorem [20], it has finitely many intersections with V_i, $\forall i=1,2,\ldots,l$ unless there is a V_j which coincides with the horizontal line. This situation is permissible since then there exists a constant solution to (5.32) $\Delta^*=X$. Otherwise, all other roots tend to either constants or $\pm\infty$. Q.E.D.

Proof of Theorem 5.1: The sufficiency follows directly from Lemmas 5.5, 5.6, 5.7 and 5.8. Lemma 5.4 shows a necessary condition for the dead-beat controllability. We need to show that $\Delta_{+\infty}{=}const.$ and/or $\Delta_{-\infty}{=}const.$ is necessary for condition 4 of Theorem 5.1 and that all dead-beat controllable systems are given by conditions 1-4.

From Lemmas 5.9, 5.10 and 5.11, it follows that one can classify all scalar polynomial systems into the following three classes:

1. conditions of Lemma 5.9: $(\ +,+)$, $(\ +,-)$, $(-,+)$ and $(-,-)$

2. conditions of Lemma 5.10: $a_0 < \partial\mathcal{U}$ or $a_0 > \partial\mathcal{U}$ on $\mathcal{P}_i, i{=}1,2$ if $\mathcal{P}_i \not\subset S_0$

3. conditions of Lemma 5.11: $\Delta_{-\infty}{=}const.$ or ∞ and/or $\Delta_{\infty}{=}const.$ or ∞.

This classification is well defined in the sense that each scalar polynomial system belongs to only one class. Each class is defined by one condition from each group of the above given conditions, e.g. $(\ +,-)$, $a_0 > \partial\mathcal{U}$, $\forall x \in \mathcal{P}_2$ and $\Delta_{+\infty}{=}const.$ represent one class of scalar polynomial systems. Note that sometimes not all the conditions can be used since they may be contradictory. For instance, if the system belongs to the class $(\ +,+)$, the conditions given by Lemmas 5.10 and 5.11 are not well defined and can not be used in the classification. The classification yields a plethora of different cases and the proof is carried out on a case-by-case analysis. All cases that may be dead-beat controllable are summarised in Table 5.1 and the situations that are not listed always yields an uncontrollable system.

Table 5.1 should be read as follows. Column 2 indicates the major subdivision. Column 3 indicates that the system is either controllable (S) or possibly controllable (N). In the following columns, the conditions that have to be satisfied are indicated by "x".

Although complete analysis is a direct argument based on the preceding discussion, it is very long and is omitted. Note that for $(-,-)$ conditions $\Delta_{+\infty}$ and/or $\Delta_{-\infty}$ must be constant. Otherwise it will be impossible to have that $\mathcal{P}_1$ and/or $\mathcal{P}_2$ are subsets of S_k. Hence, if $(-,-)$ and $\Delta_{+\infty}{=}const.$, it may be possible to have $\mathcal{P}_2 \in S_k$ and vice versa. Consider, for instance, the 10th row of Table 5.1. Since $\Delta_{+\infty}{=}const.$, it may be possible to map $\mathcal{P}_2$ to S_k. Since $a_0 > \partial\mathcal{U}$, $\forall x \in \mathcal{P}_2$, it is also necessary that $\exists x \in S_0,\ x > 0$ so that it is possible to translate $\partial\mathcal{U}$ "upwards". If $\exists x > \Delta_{+\infty}$, it is possible to have $\mathcal{P}_2 \subset S_1$. If not, the process of translating the boundary of the control value domain for maximum values of $x \in S_i$ is continued. It may happen that we obtain a limiting set $\lim_{k\to\infty} S_k{=}\mathcal{L}$ and $\mathcal{P}_2 \not\subset \mathcal{L}$ and in this case the system is not

		Cond.	1	2	3	4	5	6	7
			S_T=S_I	$a_0 < \partial\mathcal{U}$ $\forall x \in \mathcal{P}_1$	$a_0 > \partial\mathcal{U}$ $\forall x \in \mathcal{P}_1$	$a_0 < \partial\mathcal{U}$ $\forall x \in \mathcal{P}_2$	$a_0 > \partial\mathcal{U}$ $\forall x \in \mathcal{P}_2$	$\Delta_{+\infty}$=const.	$\Delta_{-\infty}$=const.
1	(+,+)	S	$\times$						
2	(-,+)	S	$\times$		$\times$				
3	(+,-)	S	$\times$			$\times$			
4	(-,+)	N	$\times$	$\times$					$\times$
5	(+,-)	N	$\times$				$\times$	$\times$	
6	(-,-)	N	$\times$	$\times$		$\times$		$\times$	$\times$
7	(-,-)	N	$\times$	$\times$			$\times$	$\times$	$\times$
8	(-,-)	N	$\times$		$\times$	$\times$		$\times$	$\times$
9	(-,-)	N	$\times$		$\times$		$\times$	$\times$	$\times$
10	(-,-)	N	$\times$		$\times$		$\times$	$\times$	
11	(-,-)	N	$\times$		$\times$	$\times$		$\times$	
12	(-,-)	N	$\times$		$\times$	$\times$			$\times$
13	(-,-)	N	$\times$	$\times$		$\times$			$\times$

Table 5.1: All dead-beat controllable cases

dead-beat controllable. On the other hand, if $\exists k$ such that $\mathcal{P}_2 \subset S_k$ according to Lemma 5.7 the system is dead-beat controllable since $a_0 > \partial\mathcal{U}$, $\forall x \in \mathcal{P}_1$. All N-cases are included in condition 4 of Theorem 5.1 and it is obvious that $\Delta_{+\infty}$=$const.$ and/or $\Delta_{-\infty}$=$const.$. Q.E.D.

Corollary 5.4 If the condition 1 of Theorem 5.1 holds, every initial state of the system (5.3) is transferred to the origin in at most μ_1 time steps, where μ_1=max(card $\bar{S}$ − card S_T + 2, card S_T). □

Corollary 5.5 If the condition 2 or 3 of Theorem 5.1 hold, every initial state of the system (5.3) is transferred to the origin in at most μ_2 time steps, where μ_2=max(card $\bar{S}$ − card S_T + 3, card S_T). □

Corollary 5.6 If condition 4 of Theorem 5.1 is satisfied, then every initial state of the system (5.3) is transferred to the origin in at most μ_3 or μ_4 time steps, where μ_3=max(card $\bar{S}$ − card S_T + 2 + k, card S_T) and μ_4=max(card $\bar{S}$ − card S_T + 3 + k, card S_T), where k is the number of steps necessary to achieve one of the situations given by conditions 1, 2 or 3. □

5.7 *An Algebraic Test for Dead-Beat Controllability*

The conditions given in Theorem 5.1 are not easy to check in general. However, we present an algorithm that is used to check conditions 1, 2 and 3 of Theorem 5.1 and which is tractable for polynomial systems (5.3) whose degrees are not too high. The abundance of different cases forces us to use several different techniques and the algorithm that we obtained relies on the use of the

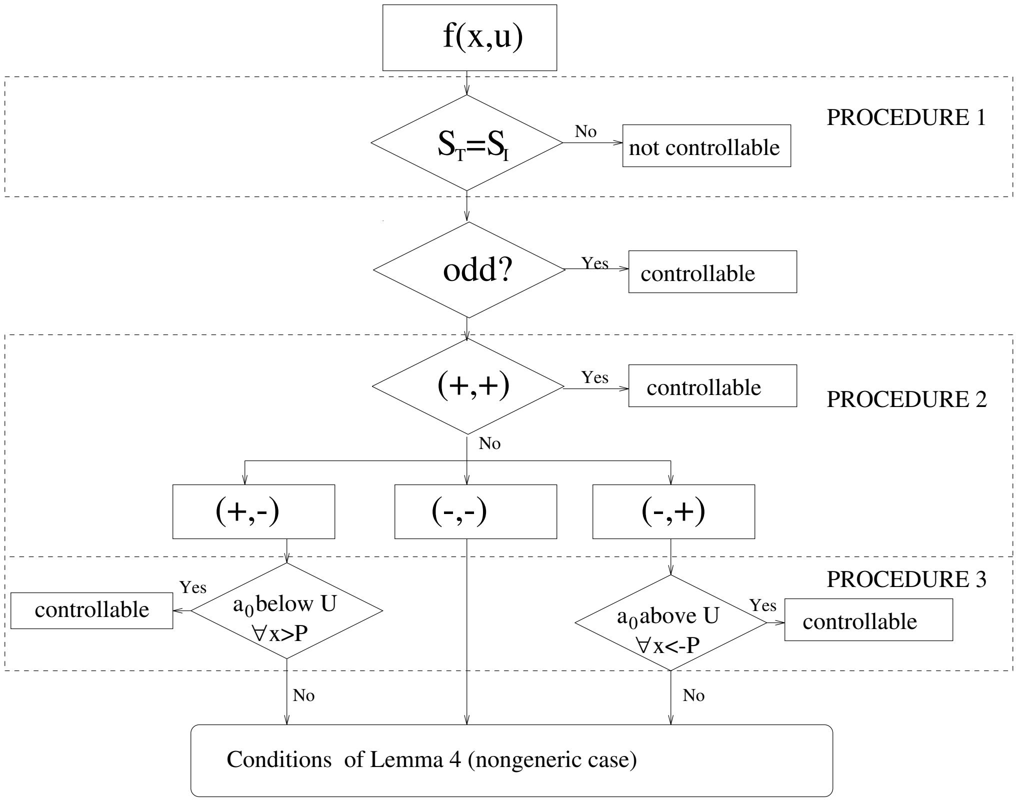

Figure 5.3: Algorithm for the dead-beat controllability test

Maple[1] software package (many of the computations are not easily done by hand) and Matlab[2]. Condition 4 of Theorem 5.1 is very hard to check and we still do not know a general method to check it. However, if n is 2 or 4 in (5.3) it is possible to check this condition as well. The classes of polynomial systems that we introduced (odd and Class 1, 2 and 3 of even systems) have different computational complexities. Odd systems are the easiest to deal with whereas Class 3 of even systems is the most complex.

We split the algorithm into three procedures. The algorithm is pictorially summarised in Figure 5.3.

Procedure 1: used to check the condition 1 of Theorem 5.1. We use Matlab for this procedure.

Step 1 Find all real roots of the polynomial $a_j(x)\,, j\in\{1,\dots,n\}$ in (5.6) which has the smallest degree in x. Denote this set as E_j=$\{x\in\mathbb{R}:a_j(x)=0\}$.

Step 2 Evaluate polynomials $a_i(x)\,,\ \forall i$=$1,2,\dots,n$ for all elements in E_j and find $E\subseteq E_j$ such that E=$\{x\in E_j : a_i(x)=0,\ \forall i$=$1,2,\dots,n\}$.

Step 3 Evaluate $a_0(x)$ for all $x\in E$ and find the set E^*=$\{y : y$=$a_0(x)\,,\ x\in E\}\cap E$. Determine all the cycles of the form $x(k+1)=a_0(x(k))\,,\forall k$=$1,\dots B_j, x(k)\in E^*$ and x_1=$a_0(x_{B_j})$, where

[1]Copyright (c) 1981-1992 by the University of Waterloo
[2]Copyright (c) 1984-94 The MathWorks, Inc.

the argument j counts the cycles. The only allowed cycle is when the origin is an equilibrium, that is $0 \in E^*$ and $a_0(0) = 0$.

Procedure 2: used to check whether the intervals $\mathcal{P}_1$ and/or $\mathcal{P}_2$ are subsets of S_0 (see the proof of Lemma 5.9). Although QEPCAD can be used to carry out Procedure 2 (see Chapter 3), we present another approach by using Maple.

Step 1 Compute the Sturm's sequence using Maple and regarding x as a parameter[3]. We obtain:

$$\begin{aligned} f_0(x,u) &= f(x,u) = l_0(x)\, u^n + \ldots \\ f_1(x,u) &= \partial/\partial u f(x,u) = l_1(x)\, u^{n-1} + \ldots \\ &\ldots \\ f_s(x,u) &= l_s(x) \end{aligned} \tag{5.33}$$

Where $l_i(x)$, $i = 0, 1, \ldots, n$ are rational functions in x and $l_0(x) = a_n(x)$, $l_1(x) = na_n(x)$. We can therefore write:

$$l_i(x) = \text{num}l_i(x) / \text{den}l_i(x), \ \forall i = 0, 1 \ldots, s \tag{5.34}$$

and $\text{num}l_i = \sum_{j=0}^{F_{i1}} c_{j1}^i x^j$, $\text{den}l_i = \sum_{j=0}^{F_{i2}} c_{j2}^i x^j$ are polynomials.

Step 2 Divide all coefficients of $\text{num}l_i$ by $c_{F_{i1}i}$, $i = 0, 1, \ldots, s$ and $\text{den}l_i$ by $c_{F_{i1}i}$, $i = 0, 1, \ldots, s$. Denote the new coefficients as $d_{jk}^i = c_{jk}^i / c_{F_{ik}i}$, $i = 0, 1, \ldots, s$, $k = 1, 2$, $j = 0, 1, \ldots, F_{ik}$. Use the formula for bound on roots [20]:

$$K = 1 + \sup_{ijk} |d_{jk}^i|, i = 0, 1, \ldots, s, \ k = 1, 2, \ j = 0, 1, \ldots, F_{ik} \tag{5.35}$$

and let $P = K + \varepsilon$, $\varepsilon > 0$. The sets (5.9) are defined as $\mathcal{P}_1 =]-\infty, -P]$ and $\mathcal{P}_2 = [P, +\infty[$.

Step 3 If $(-,+)$ or $(-,-)$ choose any $x^* \in \mathcal{P}_1$ and check whether there is a real solution to $f(x^*, u) = 0$. If there is, then $\mathcal{P}_1 \subset S_0$ and vice versa. Similarly, if $(+,-)$ or $(-,-)$, choose any $x^{**} \in \mathcal{P}_2$ and find whether there is a real solution to $f(x^{**}, u) = 0$. The existence of a real solution u guarantees that $\mathcal{P}_2 \subset S_0$ and vice versa.

Procedure 3: It is used to check whether $a_0(x)$ is "above" or "below" $\partial\mathcal{U}(x)$ on sets $\mathcal{P}_1$ and $\mathcal{P}_2$ if they are not subsets of S_0. Since $u = 0$ belongs to the control value domain, the abscissa in

[3]The command "rem" is used repeatedly in order to compute the Sturm sequence since the command "sturmseq" (Sturm sequence) does not accept parametric coefficients.

Figure 5.2 belongs to $\mathcal{U}$. As a result, if $\mathcal{P}_i \not\subset S_0,\ i{=}1,2$ then

1. if $a_0(x') > 0$ for any $x' \in \mathcal{P}_i,\ i{=}1,2$, then $a_0(x) > \partial\mathcal{U},\ \forall x \in \mathcal{P}_i$

2. if $a_0(x'') < 0$ for any $x'' \in \mathcal{P}_i,\ i{=}1,2$, then $a_0(x) < \partial\mathcal{U},\ \forall x \in \mathcal{P}_i$

Therefore, to check whether $a_0(x)$ is "above" or "below" the control value domain it is sufficient to evaluate a_0 for any $x \in \mathcal{P}_i$ and check its sign.

The algorithm to check conditions of Theorem 5.1 is given in Figure 5.3. The bottom box represents the hard problem for which we do not have a universal test although in some situations it is possible to check it (see Example 5.2). If condition 1, 2 or 3 of Theorem 5.1 is not satisfied it is sometimes possible to check whether $\Delta_{\pm\infty}{=}const.$. For example, if the system is given by:

$$x(k+1) = a_2(x(k))\,u(k)^2 + a_1(x(k))\,u(k) + a_0(x(k))$$

the distance is given by:

$$\Delta(x) = |a_0(x) - \frac{a_1^2(x)}{4a_2(x)}|$$

The limits of this function as $x \to \pm\infty$ are either $+\infty$ or a constant number. It follows that the situations when $\Delta_{+\infty}{=}const.$ and/or $\Delta_{-\infty}{=}const.$ are non generic. For $n \geq 6$ it is not possible to obtain $\Delta(x)$ in an explicit form.

5.8 Comparison with Some Known Results

It is interesting to compare the results that we presented in this chapter with some standard results for linear and classes of nonlinear systems. Of course, the comparison is restricted to scalar systems.

Linear systems: Consider the linear system:

$$x(k+1) = ax(k) + bu(k)\,,\ x, u \in \mathbb{R} \tag{5.36}$$

If $b{\neq}0$, the system is controllable and we need one step to zero any initial state [151]. Since the linear system belongs to the class of odd systems and $\bar{S}{=}\emptyset$, it follows from our Corollary 5.2 that the system (5.36) is one step controllable.

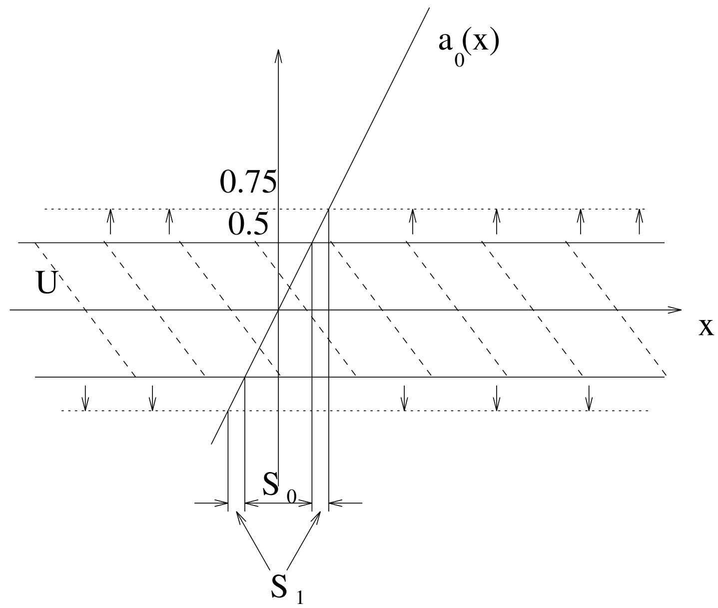

Figure 5.4: Example: a linear system with saturating controls

Bilinear systems: The bilinear system:

$$x(k+1) = ax(k) + x(k)\,bu(k)\,,\ x, u \in \mathbb{R} \tag{5.37}$$

is dead-beat controllable if and only if $b \neq 0$ [71]. We assumed $a_n(x) \not\equiv 0$ and therefore we should conclude from our results that the system (5.37) is dead-beat controllable. Indeed, the bilinear system (5.37) is an odd system and $\bar{S} = \{0\}$. Also, it is obvious that $S_I = S_T = \{0\}$ and from Lemma 5.5 it follows that (5.37) is dead-beat controllable.

Linear systems with bounded control signals: In [174], the linear systems with saturated controls of the form:

$$x(k+1) = ax(k) + bu(k)\,,\ x \in \mathbb{R},\ |u| \leq 1 \tag{5.38}$$

were considered. From [174], it follows that the system (5.38) is completely controllable if and only if $|a| \leq 1$. It should be pointed out that in [174] the definition of controllability is different from ours. Namely, we require a uniform bound on the number of steps necessary to transfer any initial state to the origin whereas in [174] there is no uniform bound. For instance, the system $x(k+1) = x(k) + u(k)^2 + u(k)$ is not dead-beat controllable in the sense of our definition whereas it is controllable according to [174]. Obviously, our notion of controllability is stronger. Nevertheless, we show by an example how it is possible to use the control value domain methodology in order to obtain the same answers.

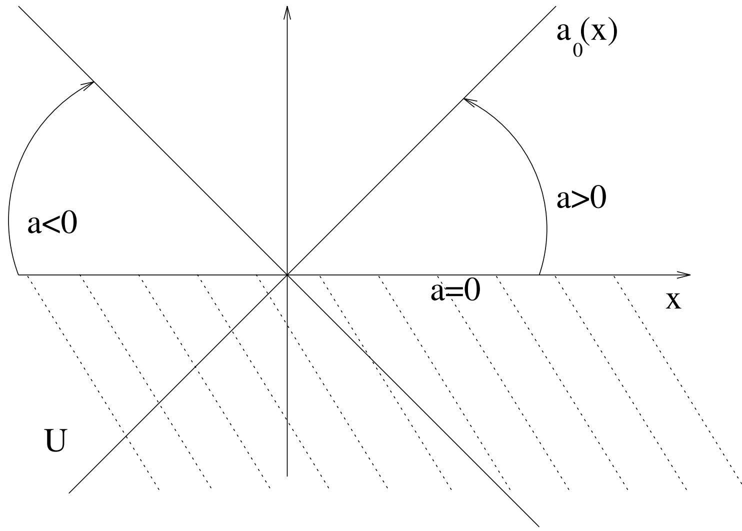

Figure 5.5: Linear scalar system with positive control signals

Consider the system

$$x(k+1) = 2x(k) + u(k)\,,\ |u| \leq 1$$

The control value domain is "a strip" in $\mathbb{R}^2$, $\mathcal{U}=\mathbb{R} \times \{u : |u| \leq 1\}$. Using our method we obtain (see Figure 5.4) that $S_0=\{x : |x| \leq 0.5\}$. Since it is possible to translate $\mathcal{U}$ in positive and negative directions for ± 0.5, we obtain $S_1=\{x : 0.5 < |x| \leq 0.75\}$. Continuing the same construction, it follows that there is a limiting set $\mathcal{L}=\{x : |x| < 1\}$ which is such that any initial state $x \in \mathcal{L}$ can be transferred to the origin in finite time whereas any $x \notin \mathcal{L}$ can not be zeroed. This always happens if $|a| > 1$ and the limiting set is $\mathcal{L}=\{x : |x| < |b/(a-1)|\}$, $|u| \leq 1$. On the other hand, if $|a| \leq 1$, any initial state can be transferred to the origin in finite time.

Linear systems with positive controls: Controllability of linear systems with positive controls was considered in [50]. The system has the form:

$$x(k+1) = ax(k) + bu(k)\,,\ x \in \mathbb{R},\ u \in [0, +\infty[\tag{5.39}$$

If we introduce a new control variable $v(k)^2 = u(k)\,,\ v(k) \in \mathbb{R}$, we obtain a scalar polynomial system. Necessary and sufficient condition for controllability of (5.39) is that $b \neq 0$ and $a < 0$. Note that in [50] complete controllability for any two states is considered, whereas we consider controllability to a point (the origin). Therefore, we can expect that conditions in [50] are stronger than ours. Indeed, if $a < 0$, the control value domain is $\mathcal{U}=\mathbb{R} \times]-\infty, 0]$ (see Figure 5.5) and from Lemma 5.7, point 2, it follows that the system is dead-beat controllable. Moreover, if a=0, the system (5.39) is dead-beat controllable but it is not completely controllable in the sense of [50]. If $a > 0$, however, the system (5.39) is not controllable in either sense.

In summary, our results recover some known results restricted to scalar systems. The methods are shown to be applicable to a much larger class of nonlinear scalar systems. We believe, for instance, that our method can be modified to encompass some classes of scalar analytic dynamics as well. Modifications that may be needed concern properties of the control value domain and its boundary, whereas the method does not need any changes since it reflects a property of *scalar systems*.

5.9 Local Dead-Beat Stabilisability

It is apparent that the concept of dead-beat controllability and the dead-beat control which is associated with it are not always satisfactory in terms of the system's behaviour. It was shown in Chapter 2 that a system can be dead-beat controllable but unstable. Since stability can not be ignored, we investigate conditions which guarantee dead-beat controllability and stability at the same time. Additionally, it is not always necessary to consider the system's behaviour in the whole state space and very often it is sufficient to restrict all the investigation to a neighbourhood of the origin. This motivates the next definition:

Definition 5.6 The system (5.3) is locally dead-beat stabilisable if there exists a neighbourhood $\mathcal{N} \subseteq \mathbb{R}$ of the origin and a function $g : x \to g(x), \forall x \in \mathcal{N}$ such that the system $x(k+1) = f(x(k), g(x(k)))$ is stable and any states from the neighbourhood is transferred to the origin in finite time. □

We use the standard definition of $\epsilon - \delta$ stability [154].

Theorem 5.2 *The system (5.3) is locally dead-beat stabilisable if* $a_0(0) \in \overset{\circ}{\mathcal{U}}(0)$. □

Proof of Theorem 5.2: The condition of Theorem 5.2 guarantees that there exists a neighbourhood of the origin $\mathcal{N}$, such that $x \notin \bar{S}$, $\forall x \in \mathcal{O}$; since we require that $a_0(0)$ belongs to the interior of the control value set $\mathcal{U}(0)$, it can not happen that $0 \in \bar{S}$ (otherwise the interior would be an empty set and the condition of the theorem would not be satisfied). Thus, because of properties of polynomials there exists a neighbourhood of the origin in which there are no equilibria or periodic orbits. Also, it follows that the control value set $\mathcal{U}(0)$ is either $]-\infty, +\infty[$, $[P, +\infty[$ or $]-\infty, P]$.

From the condition of Theorem 5.2 it follows that there is a real solution to (5.17) for x=0. Since $a_0(0)$ belongs to the interior of the control value set $\mathcal{U}(0)$ we can always find two values of control u_1 and u_2 for which the following inequality holds

$$-f_1(0, u_1) < a_0(0) < -f_1(0, u_2)\,.$$

Observe that $-f_1(x, u_i) \in \mathcal{U}, \forall x \in \mathbb{R}$ and i=1, 2 and since the functions $a_0(x)$ and $-f_1(x, u_i)$, i=1, 2 are polynomials it follows that there is a neighbourhood of the origin on which it holds:

$$-f_1(x, u_1) < a_0(x) < -f_1(x, u_2)\,.$$

Therefore, there exists a neighbourhood of the origin in which all the states can be transferred to the origin in one step. Thus, the system is dead-beat stabilisable. Q.E.D.

Observe that in some cases the condition of Theorem 5.2 sufficient for the existence of a stabilising dead-beat control in any neighbourhood of the origin $\mathcal{O} \subset \mathbb{R}$. The following corollary gives sufficient condition for the existence of global stabilising dead-beat control laws. We do not consider the continuity of control law and it may happen that the supremum of the control signal is infinite.

Corollary 5.7 *There is a global stabilising dead-beat control for the system (5.3) if the conditions of Theorems 5.1 and 5.2 are satisfied.* □

It follows directly from the proofs of Theorems 5.1 and 5.2.

5.10 Local Dead-Beat Stabilisability with a Bounded Control Signal

It may happen that the solutions to the dead-beat control problem yield very large control signals; since we want to drive every initial state to zero in the shortest time it is natural to expect large control. Consequently, the existence of a dead-beat control law does not guarantee that it can be implemented. We will consider the following problem:

Given the system (5.3), and the condition

$$u \in [-K, +K] = \mathcal{K} \tag{5.40}$$

find conditions for the existence of a local dead-beat control. We redefine the control value set and domain to include bounded controls:

$$\mathcal{U}(x)=\{y : y=-f(x,u)\ ,\ u\in\mathcal{K}\}$$

$$\mathcal{U}=\bigcup_{x\in\mathbb{R}}(\{x\}\times\mathcal{U}(x))\subset\mathbb{R}^2$$

Theorem 5.3 *The system (5.3) with the input constraint (5.40) is locally dead-beat stabilisable if* $a_0(0)\in\overset{\circ}{\mathcal{U}}(0)$. □

Proof of Theorem 5.3: The proof of Theorem 5.3 is the same as that of Theorem 5.2; the only difference is that the control value set at x=0 has the form $\mathcal{U}(0)=[P_1,P_2]\ ,\ P_i\in\mathbb{R},\ i$=1, 2. We shall denote the values of $u\in\mathcal{K}$ for which $-f_1(0,u)$ attains global minimum and maximum respectively as u_{min} and u_{max}. It is obvious that:

$$\begin{aligned}\min_u(-f_1(x,u))\le -f_1(x,u_{min})\le \max_u(-f_1(x,u))\\ \min_u(-f_1(x,u))\le -f_1(x,u_{max})\le \max_u(-f_1(x,u))\ ,\ \forall x\in\mathbb{R}\end{aligned} \tag{5.41}$$

Since there is a real solution to $f(x,u)=0,\ \forall a_0(x)\in\mathcal{U}$, it follows that the solution exists if

$$-f_1(x,u_{min})<a_0(x)<-f_1(x,u_{max}) \tag{5.42}$$

Since all the functions in (5.42) are polynomials and since (5.42) is satisfied for x=0, it follows that there exists a neighbourhood of the origin in which the last inequality holds. As a result, there is a local dead-beat control law in the neighbourhood of the origin. Q.E.D.

5.11 Examples

Example 5.1 Consider the system:

$$x(k+1)=x(k)\,u(k)^6+(x(k)+1)\,u(k)^3-2u(k)^2+3x(k)\,u(k)+2x(k)=f(x(k)\,,u(k))$$

Using the algorithm presented in this chapter, we determine if the system is dead-beat controllable.

Procedure 1 Since $a_2(x)=-2$, the control independent set is empty and the consequently S_I=S_T=$\emptyset$.

Procedure 2

Step 1 We compute the Sturm sequence of the polynomial $f(x(k), u(k))$, regarding the variable x as a parameter, and obtain:

$$\begin{aligned}
f_0(x,u) &= xu^6 + (x+1)u^3 - 2u^2 + 3xu + 2x \\
f_1(x,u) &= 6xu^5 + 3(x+1)u^2 - 4u + 3x \\
f_2(x,u) &= -(x+1)u^3/2 + 4u^2/3 - 5ux/2 - 2x \\
f_3(x,u) &= (1746x^3 - 27 - 1132x + 1494x^2 + 189x^4)u^2/9(x+1)^3 - 2(225x^4 - 434x^2 + 123x^3 - 6 - 18x)u/ \\
&\quad 3(x+1)^3 - x(9 - 485x + 387x^2 + 369x^3)/3(x+1)^3 \\
f_4(x,u) &= 6(12540x^2 + 616887x^6 + 97428x^3 + 88036x^4 + 118170x^5 - 8705x + 737505x^8 + 1063773x^7 \\
&\quad + 184194x^9 - 108)u/(1764x^3 - 27 - 1132x + 1494x^2 + 189x^4)^2 \\
f_5(x,u) &= (147311725113x^{15} + 3239903130156x^{14} + 24559958159352x^{13} + 75012346813320x^{12} \\
&\quad + 83571851597166x^{11} - 11817016513884x^{10} - 62904971745612x^9 + 1000927456776x^8 \\
&\quad + 19452507666921x^7 - 5541094157796x^6 - 2571086563956x^5 + 1560445934832x^4 \\
&\quad - 94944167336x^3 - 9376966188x^2 - 200550816x - 1259712)x/ \\
&\quad ((184194x^6 + 184923x^5 - 43578x^4 + 8658x^3 + 38007x^2 - 8381x - 108)^2(x+1)^3)
\end{aligned}$$

Step 2 All coefficients of numerators (denominators) of leading coefficient functions are divided by the coefficient of the leading term of the corresponding numerator (denominator) because we can then use the formula for a bound on roots instead of computing all real roots. For example, we divide all coefficients of the numerator of $f_5(x,u)$, which is itself the leading coefficient function $l_5(x)$, by $147, 311, 725, 113$. The "scaled" leading coefficient functions are given below:

$$l_0(x) = x,\ l_1(x) = 6x,\ l_2(x) = -0.5(x+1)$$

$$l_3(x) = 21(x^4 + 9.333x^3 + 7.9048x^2 - 5.9894x - 0.4429)/(x+1)^3$$

$$\begin{aligned}
l_4(x) = 5.1565(&x^9 + 4.004x^8 + 5.7753x^7 + 3.3491x^6 + 0.6416x^5 + 0.478x^4 + 0.5289x^3 + 0.0681x^2 \\
&-0.0473x - 0.0005)/(x^4 + 9.333x^3 + 7.9048x^2 - 5.9894x - 0.1429)^2
\end{aligned}$$

$$\begin{aligned}
l_5(x) = 0.3618(&x^{15} + 21.9935x^{14} + 166.721x^{13} + 509.2083x^{12} + 567.313x^{11} - 80.2178x^{10} - 427.0194x^9 + 67.9506x^8 \\
&+ 132.05x^7 - 37.6148x^6 - 17.4534x^5 + 10.5928x^4 - 0.6445x^3 - 0.0637x^2 - 0.0014x - 0.000008)/((x^6 \\
&+ 1.004x^5 - 0.2366x^4 + 0.047x^3 + 0.2063x^2 - 0.0455x - 0.0005)^2(x+1)^3)
\end{aligned}$$

It is now easy to compute the interval inside which all roots of numerators and denominators of the leading coefficient polynomials lie.

$$P = 1 + 567.3130 = 568.3130$$

Therefore, we can define $\mathcal{P}_1 =]-\infty, -570]$ and $\mathcal{P}_2 = [570, +\infty[$.

Step 3 We find all real roots of $f(-1000, u) = 0$. There are two real roots -1.2386 and -0.6088 and hence $\mathcal{P}_1 \subset S_0$. Real roots to $f(1000, u) = 0$ are -1.2396 and -0.6082. Consequently, $\mathcal{P}_2 \subset S_0$. The system is dead-beat controllable since $S_I = S_T$ and condition 1 of Theorem 5.1 holds. Note that the system belongs to the $(+, +)$ class of even systems. □

Example 5.2 Consider the system:

$$x(k+1) = 4(x(k)^2 + 1)\, u(k)^2 + x(k)^2\, u(k) + (Kx(k)^2 - 1/2)/16$$

where $K \in \mathbb{R}$. We shall consider several cases that may arise depending of the value of K. Using the introduced notation, we can write:

$$\Delta_{\pm\infty} = \lim_{x \to \pm\infty} |(K-1)\, x^2/16 + 1/32|$$

and

$$\Delta(x) = |a_0(x) - \max_u(-f_1(x, u))| = \left|\frac{(K-1)\, x^4 + (K - 1/2)\, x^2 - 1/2}{16(x^2+1)}\right|,$$

if

$$a_0(x) > \max_u(-f_1(x, u)).$$

Consider now the first case $K > 1$. Since $a_2(x) \neq 0, \forall x, K \in \mathbb{R}$ it follows that $\bar{S} = \emptyset$, that is the first condition of Theorem 5.1 is satisfied. The set for which there is a dead-beat control law at the first step is $S_0 = \{x : C_1 \leq x \leq C_2\}$ where C_1 and C_2 are real numbers which depend on K. It is obvious that the second condition of Theorem 5.1 is not satisfied and since $\Delta_{\pm\infty} = +\infty$, the third condition also does not hold. Thus, the system is not dead-beat controllable.

If $K = 1$ then $S_0 = \{x : -1 \leq x \leq 1\}$. Therefore, the second condition of Theorem 5.1 is not satisfied but the third condition of Theorem 5.1 is satisfied and the system is 2-step controllable (it is easy to check that all the initial states that do not belong to S_0 can be mapped to the point 1/32 in the first step and then to the origin in the second step).

The case $K < 1$ will be divided into two subcases depending on the minimum number of steps necessary for dead-beat control. First, it is obvious that the first two points of Theorem 5.1 are satisfied ($\Delta_{\pm\infty}$=0) and therefore there is dead-beat control. If $-0.5+\sqrt{2}<K<1$ then S_0=$\{x:-\infty<x\leq C_1\}\cup\{x:C_2\leq x\leq C_3\}\cup\{x:C_4\leq x<\infty\}$ and the system is 2-step controllable. For example, if K=0.99 the set S_0=$\{x:-\infty<x\leq -6.9251\}\cup\{x:-1.0211\leq x\leq 1.0211\}\cup\{x:6.9251\leq x<\infty\}$. In the second case we have $K\leq -0.5+\sqrt{2}$ and S_0=$\mathbb{R}$ and therefore the system is 1-step dead-beat controllable.

Consider now local dead-beat stabilisability of the system. It holds $\forall K\in\mathbb{R}$:

$$a_0(0)=-1/32\in\mathring{\mathcal{U}}(0)=\{y:-\infty<y\leq 0\}$$

and consequently the system is locally dead-beat stabilisable. Therefore, from Corollary 5.7 it follows that there exists a global stabilising dead-beat control law for $K\leq 1$.

If we assume that K=$1+\varepsilon$ where $\varepsilon\in\mathbb{R}$ is a parametric uncertainty we have that the system is not dead-beat controllable $\forall\varepsilon>0$. Hence, the concept of dead-beat controllability may not be robust to parametric uncertainties. □

Example 5.3 Consider the system $x(k+1)=x^2(k)+u^2(k)-2u(k)$. It is obvious that there is stabilising dead-beat control since $a(0)=0\in\mathring{\mathcal{U}}(0)=]-\infty,1[$. Also, $u(0)=1\pm\sqrt{1-x^2(k)}$ is the required dead-beat control law on the neighbourhood S_0=$\{x:-1\leq x\leq +1\}$. We are now interested in mapping all initial states from a larger set into the S_0. This is possible to achieve with the control of the form $u(0)=1\pm\sqrt{2-x^2(k)}$ which is defined on the set S_1=$\{x:-\sqrt{2}\leq x\leq +\sqrt{2}\}$, etc. Therefore, we have a sequence of nested intervals $S_0\subset S_1\subset\ldots$ on which there is stabilising dead-beat control. This sequence, however, has a limit set $\mathcal{L}$=$\{x:-(0.5+\sqrt{5}/2)<x<+(0.5+\sqrt{5}/2)\}$ from which all the initial states can be transferred to the origin in a finite number of steps. The set $\mathcal{L}$ is a proper subset of the state space. It is obvious, for instance, that there is no such control law which can transfer the system from initial state $x(0)=2\notin\mathcal{L}$ to the set S_0 and then to the origin. All trajectories that start from initial states which are not in the set $\mathcal{L}$ diverge to $+\infty$ regardless of the control law. □

Example 5.4 Consider the system:

$$x(k+1)=(x(k)+1)^2-2+(x(k)+1)(x(k)+2)(u(k)^2+u(k)).$$

Since $\bar{S}$=$\{-1,-2\}$ and $f(-1,\cdot)=-2$ and $f(-2,\cdot)=-1$, i.e. there is a periodic point of period

2, the system is not dead-beat controllable. However, since $a_0(0) = -1 \in \overset{\circ}{\mathcal{U}}(0) = \{y : -\infty < y \leq 1/2\}$ the system is locally dead-beat stabilisable. □

Example 5.5 Consider the system:

$$x(k+1) = 4(x(k)^2 + 1)\,(u(k)^2 + u(k)\,) + x(k)^2 + x(k) \text{ and } u(k) \in [-1, +1]$$

In this case we will use Theorem 5.3 in order to check the existence of dead-beat control. We can write for $x=0$

$$-f_1(0, u) = -4(u^2 + u)$$

Therefore, since $u_{max} = -1/2$ and $u_{min} = 1$ we have that:

$$a_0(0) = 0 \in \overset{\circ}{\mathcal{U}}(0) = \{y : -8 \leq y \leq 1\}$$

and there is local dead-beat control. □

Example 5.6 The following example shows that dead-beat control may be very sensitive to structural changes. Consider the system:

$$x(k+1) = \epsilon u(k)^4 + 0.02a(x(k)\,)\,u(k)^2 + a^2(x(k)\,)$$

where $a(x(k)\,) < 0, \forall x(k) \in \mathbb{R}$ is a polynomial and $\epsilon \geq 0$. The control signal that zeroes initial states in the first step is

$$u(0) = \pm\sqrt{\frac{-a(x(0)\,) - 2a(x(0)\,)\sqrt{10^{-4} - \epsilon}}{2\epsilon}}$$

It is obvious that if $\epsilon \leq 10^{-4}$ then there is global stabilising dead-beat control. On the other hand, if $\epsilon > 10^{-4}$ there is no dead-beat control. If ϵ is considered to be a structural uncertainty and if we assume that the smallest coefficient of $a(x)$ is $C = 10^3$, then their ratio is $C/\epsilon = 10^7$. □

5.12 *Case Study 2: a Heat Exchanger*

The model of a heat exchanger [75] can be identified as a two-input one-output scalar polynomial model. The system is given in Figure 5.6. The temperature of the inlet water is constant $T_0 = 16°C$

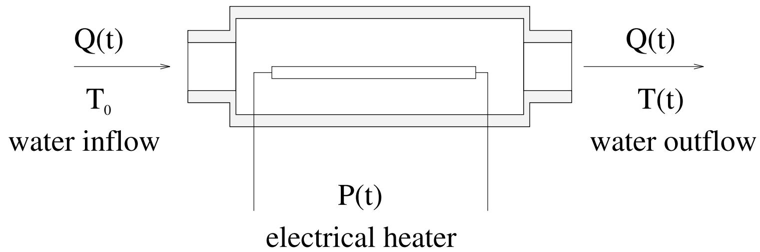

Figure 5.6: An electrically heated heat exchanger

during the identification. The temperature of the outlet water varies with the change of the heating power P and the water flow Q. The difference between the inlet and outlet temperatures $\Delta T = T(t) - T_0$ is the output of the system and P and Q are inputs to the system. The obtained polynomial NARMAX model is:

$$\Delta T(k) = 0.6612\Delta T(k-1) + 4.007P(k-1) - 0.0157341P(k-1)Q(k-1) + 0.006656\Delta T^2(k-1)$$

where k is the time index.

If we introduce the notation $x(k) = \Delta T(k)$, $u_1(k) = P(k)$, $u_2(k) = Q(k)$ we write

$$x(k+1) = 0.6612x(k) + (4.007 - 0.0157341u_2)\,u_1(k) + 0.006656x^2(k)$$

If we fix either of the control variables u_1 or u_2 and control the system with the other, we obtain a scalar polynomial system of the form considered in this chapter. Since the system is bilinear in control variables, we need to avoid the critical value $u_1 = 0$ if u_2 is chosen to be a control signal. Similarly, if we want to control the system with u_1 while u_2 is fixed, the value $u_2 = 254.66979$ should be avoided. Otherwise we have a loss of controllability.

If the critical value for u_1 (or u_2) is avoided, we obtain an odd polynomial system whose control independent set $\bar{S}$ is empty and therefore the system is dead-beat controllable (see Corollary 5.1). Moreover, it is dead-beat stabilisable according to Corollary 5.3.

5.13 Conclusion

We have presented results on dead-beat controllability and stabilisability of polynomial scalar systems. In principle, conditions for the existence of local dead-beat control with unbounded and bounded control signal are very easy to check. Many interesting phenomena that we illustrated by examples indicate difficulties that one may expect when dealing with higher order systems. The algebraic dead-beat controllability test suffers from computational inefficiency which is more an intrinsic feature of the polynomial structure of the systems than a deficiency of the method.

The fact that classes of systems for which dead-beat controllability is difficult to check are non generic is rather unexpected. It would be very interesting and important to prove whether similar results can be obtained for higher order polynomial systems, since this would lead to characterisation of classes of systems for which QEPCAD can be successfully used to test dead-beat controllability (see Chapter 3). The fact that we can generically achieve dead-beat control in at most three steps for scalar polynomial systems justifies the use of QEPCAD for scalar polynomial systems.

The dead-beat controllability test can be automated using Maple and Matlab (or QEPCAD as an alternative to Maple - see Chapter 3). In general, the dead-beat controller that we presented needs to be modified since control magnitudes may be too large, although it may perform well in certain situations. However, we regard it to be a first step towards the design of a feasible dead-beat controller. In some cases, poor robustness of global dead-beat control may be the main hindrance to its implementation. This is illustrated by examples.

Chapter 6

A Class of Odd Polynomial Systems

6.1 Introduction

This chapter is dedicated to the consideration of output dead-beat control for a class of nonlinear systems whose mathematical model can be described by a single I-O polynomial difference equation. An application of this class of systems can be found in [24] where a subsystem of a radiator and fan system is identified in this form. The class of systems that we consider in this chapter represents a subclass of polynomial systems that may be obtained using structure identification techniques of block oriented models [76]. Also, some classes of discrete-time bilinear systems may have this I-O representation [70]. Systems considered in this chapter are a subclass of systems considered in Chapter 4.

We have illustrated why the invariant sets are important for output dead-beat controllability in Chapter 4. However, we show here that the so called *strongly invariant sets* completely determine output dead-beat controllability of systems considered in this chapter. Different forms of invariant sets that we consider are closely related to the pioneering work [70] on controllability of a class of discrete-time bilinear systems. There, it was noticed that the loss of controllability might be caused by the existence of an invariant set which is called "a trajectory insensitive to control". However, the notions of strong invariance and invariance that we consider appear to be more general (if there exists a trajectory insensitive to control, it is in general a subset of the maximal invariant set, which we use).

We exploit the polynomial structure of this class of systems to a great extent and answer the question of existence of output dead-beat control. Necessary and sufficient conditions for output dead-beat controllability are obtained. We propose two different controllability tests. One test is based on a repeated use of the Gröbner basis test from Chapter 4. We emphasize that the test is finitely computable as opposed to tests presented in Chapters 3 and 4, where we might have a non terminating procedure.

Results of this and Chapters 3 and 4 show a trade-off between the generality of the considered systems and the feasibility of the controllability test. Indeed, in Chapter 3 we considered the most general class of systems in this thesis but the obtained controllability tests are computationally the most expensive (when compared to tests presented in Chapters 4 and 5). The output dead-beat controllability test described in Chapter 4 is less computationally expensive but the class of systems is smaller. However, the test still may not be finitely computable since QEPCAD needs to be used in general. The output dead-beat controllability test of this chapter is much simpler when compared to the tests of Chapters 3 and 4.

As an alternative to the Gröbner basis controllability test, we can use another test which requires a decomposition of a polynomial into irreducible polynomials and checking whether a set of polynomial divisions is satisfied or not. We present this alternative test only for second order systems but a generalisation to systems of arbitrary order is immediate.

6.2 Preliminaries

The class of systems that we consider in this chapter are given below:

$$y(k+1) = F(y(k), u(k-1), u(k-2), \ldots, u(k-n+1), u(k)) \tag{6.1}$$

where y and u are respectively the output and input of the system and $F[y, v_1, v_2, \ldots, v_{n-1}, u] \in \mathbb{Q}[y, v_1, v_2, \ldots, v_{n-1}, u]$. We introduce the state variables

$$x_1(k) = y(k),\ x_2(k) = u(k-1),\ \ldots,\ x_n(k) = u(k-n+1)$$

and obtain the state and output equations:

$$\begin{aligned}
x_1(k+1) &= F(x_1(k), x_2(k), \ldots, x_n(k), u(k)) \\
x_2(k+1) &= u(k) \\
x_3(k+1) &= x_2(k) \\
\vdots\ &\quad\ \vdots
\end{aligned}$$

$$\begin{aligned} x_n(k+1) &= x_{n-1}(k) \\ y(k) &= x_1(k) \end{aligned} \tag{6.2}$$

The polynomial F can be written as:

$$F(x_1, x_2, \ldots, x_n, u) = a_m(x_1, \ldots, x_n)\, u^m + \ldots + a_0(x_1, \ldots, x_n) \tag{6.3}$$

Some of the definitions and assumptions given below are already stated in Chapter 4 but we repeat them for completeness. If the highest coefficient polynomial a_m in (6.3) is not equal to zero for some $x(0) \in \mathbb{R}^n$, the set of one step reachable outputs from the initial state $x(0)$ is equal to $\mathbb{R}$. If for some $x(0) \in \mathbb{R}^n$ the coefficient a_m vanishes, the set of one step reachable outputs for the system (6.2) may be much smaller. Therefore, the only states from which it may not be possible to zero the output belong to the real variety V_C defined by

$$V_C = \{x \in \mathbb{R}^n : a_m(x) = 0\} \tag{6.4}$$

Definition 6.1 The variety V_C given by (6.4) is called the **critical variety**. □

The polynomials and varieties of special form are defined below:

Definition 6.2 Polynomials of special form are given by:

$$f_s = x_1 - \sum_{i_2 i_3 \ldots i_n} b_{i_2 i_3 \ldots i_n} x_2^{i_2} x_3^{i_3} \ldots x_n^{i_n},\ b_{i_2 i_3 \ldots i_n} \in \mathbb{Q}, \forall i_2, i_3, \ldots i_n$$

and varieties $V_s = V(f_s)$ are called varieties of special form. □

Definition 6.3 The number of varieties of special form that are contained in V_C is denoted by B. □

Notice that there may be only finitely many varieties of special form that are contained in the critical variety $V_C = V(a_m)$. Also, polynomials of special form and the varieties of special form are *irreducible*, since they can be parameterised by polynomials [37].

Definition 6.4 The variety $V(x_1)$ is denoted as V_O and is called the zero output variety. □

Since the highest exponent of u in the equation (6.3) is odd and the coefficient $a_m(x)$ is not identically equal to zero, the variety V_C has a dimension lower than n.

Assumption 6.1 m in (6.3) is an odd integer. □

Assumption 6.2 $\forall x \in V(a_m), \exists u \in \mathbb{R}, F(x,u)=0$. □

Definition 6.5 The p-step reachable set $V_r^p(x_0)$ from an initial state $x(0)$ is given by:

$$V_r^p(x(0))=\{\zeta : \zeta=f_{u(p-1)} \circ \ldots \circ f_{u(0)}(x(0)), u(i) \in \mathbb{R}, \forall i=0,\ldots,p-1\} \tag{6.5}$$

□

Consider the polynomial $f^r=f^r(x(n-1),x(0))$ defined as

$$\begin{aligned} f^r &= x_1(n-1)-F_{x_n(n-1)} \circ F_{x_{n-1}(n-1)} \circ \ldots F_{x_3(n-1)} \circ F_{x_2(n-1)}(x(0)) \\ &= x_1(n-1)-\sum_{i_1 i_2 \ldots i_n} B_{i_2 i_3 \ldots i_n}(x(0))\, x_2^{i_2}(n-1)\, x_3^{i_3}(n-1) \ldots x_n^{i_n}(n-1) \end{aligned} \tag{6.6}$$

The variety $V(f^r(x(n-1),x(0)))$ defines the set of states that can be reached from $x(0)$ in $n-1$ time steps, that is $V_r^{n-1}(x(0))=V(f^r(x(n-1),x(0)))$. The polynomial f^r has special form $\forall x(0) \in \mathbb{R}^n$ and we can conclude the following [37]:

1. $V_r^{n-1}(x(0))$ is an irreducible variety, $\forall x(0) \in \mathbb{R}^n$
2. $\dim V_r^{n-1}(x(0))=n-1$, $\forall x(0) \in V_C$
3. $f^r(x(n-1),x(0))$ is an irreducible polynomial in $x(n-1)$, $\forall x(0) \in \mathbb{R}^n$

Let V and W be varieties. We introduce notation:

$$V \rightsquigarrow W \tag{6.7}$$

to denote that $V_r^{n-1}(x)=W$, $\forall x \in V$. It should be emphasised that the equation (6.7) means that the one step reachable set from any initial state in V is *equal to* W. Therefore, by definition $x(0) \rightsquigarrow V_r(x(0))$.

Definition 6.6 A set $V_{I_j} \subseteq V_C$ is invariant if

$$\forall x \in V_{I_j},\ V_r^{n-1}(x) \subseteq V_{I_j} \tag{6.8}$$

The union of all invariant sets $V_I=\cup_j V_{I_j}$ is called the maximal invariant set. □

Definition 6.7 A subset W_{I_j} of the variety V_C is strongly invariant if it is invariant and $\forall x(0) \in W_{I_j}$ there exists an integer $t \geq 0, t{=}t(x(0))$ and a sequence of controls $\mathcal{U}_t$ which yields $x(t+1, x(0), \mathcal{U}_t){=}x(0)$. The union of all strongly invariant sets $W_I{=}\cup_j W_{I_j}$ is called the maximal strongly invariant set. □

Definition 6.8 The number of varieties of special form that are contained in the maximal strongly invariant set W_I of V_C is denoted by L. □

Because of Assumption 6.1, we can split the dead-beat control problem into two parts. First we find conditions which guarantee that it is possible to zero the output for any initial state. Once we have zeroed the output, Assumption 6.1 guarantees that there exists a control sequence which keeps the output at zero for all future time steps.

In other words, the sequence $\mathcal{U}$ that yields dead-beat behaviour may be split into two parts. $\{u(0),\ u(1), \ldots,\ u(t-1)\}$ is the part of the sequence $\mathcal{U}$ that transfers the output to zero and $\{u(t), \ldots\}$ the part which keeps the output at zero. This chapter is dedicated to the first part. The behaviour of the dynamics for the control that maintains the output at zero will be discussed in Chapter 11.

6.3 Output Dead-Beat Controllability

The special structure of the system (6.2) yields the particular structure of the reachable set $V_r^{n-1}(x(0))$ which we exploit to simplify the output dead-beat controllability test:

Lemma 6.1 *The maximal strongly invariant set $W_I \subseteq V_C$ can be decomposed into a finite union of the varieties of special form $W_I{=}V_{s1} \cup V_{s2} \cup \ldots \cup V_{sL}, L \leq B$.* □

Proof of Lemma 6.1: The proof is carried out in several steps. First, we prove that at least one variety V_s of special form belongs to the strongly invariant set. Then we show that if two points that belong to a variety of special form V_s have distinct $n-1$-step reachable sets, then the variety V_s can not be a subset of an invariant set W_I. By induction we prove that the union of varieties of special form is a subset of W_I. Finally, it is shown by contradiction that W_I is equal to the union of varieties of special form.

STEP 1 Consider any initial state $x(0) \in W_I$. From the invariance of W_I it follows that $V_r^{n-1}(x(0)) \subset W_I$. Denote $V_r^{n-1}(x(0))$ as V_{s1}.

STEP 2 V_{s1} is a subset of the strongly invariant set W_I. Notice that if at least one of the coefficients $B_{i_2 i_3 \ldots i_n}(x(0))$ in (6.6)is such that its image is an interval when viewed as a function on the variety V_{s1}, then that state $x(0)$ can not belong to an invariant set $W_I \subset V_C$. Indeed, this would imply that infinitely many distinct varieties of special form are contained in V_C, which can not be the case. Hence, because of invariance of W_I we have that states in V_{s1} are mapped to finitely many varieties of special form which are contained in V_C.

Suppose now that if $x(0) \in V_{s1}$, then either $V_r^{n-1}(x(0))=V_{s2}$ or $V_r^{n-1}(x(0))=V_{s3}$ where $V_{s2} \neq V_{s3}$. From the structure of (6.6) we see that $V_r^{n-1}(\tilde{x}(0)) \neq V_r^{n-1}(\hat{x}(0))$ if and only if there exists $i_1^* i_2^* \ldots i_n^*$ such that $B_{i_1^* i_2^* \ldots i_n^*}(\tilde{x}(0)) \neq B_{i_1^* i_2^* \ldots i_n^*}(\hat{x}(0))$. Assume that:

$$\forall x(0) \in V_{s1}, \quad B_{i_1^* i_2^* \ldots i_n^*}(x(0))=b^1 \text{ or } B_{i_1^* i_2^* \ldots i_n^*}(x(0))=b_2, \; b_1 \neq b_2$$

Consider now the polynomials $B_{i_1^* i_2^* \ldots i_n^*}(\zeta)-b^1$ and $B_{i_1^* i_2^* \ldots i_n^*}(\zeta)-b^2$ where $\zeta \in V_{s1}$. By construction, these polynomials are not identically equal to zero on V_{s1} but their product is:

$$(B_{i_1^* i_2^* \ldots i_n^*}(\zeta)-b^1)(B_{i_1^* i_2^* \ldots i_n^*}(\zeta)-b^2) \equiv 0, \; \forall \zeta \in V_{s1}$$

This, however, contradicts the irreducibility of V_{s1} [37, pg. 216]. By contradiction, we have that $B_{i_1 i_2 \ldots i_n}(x(0))=const., \; \forall i_1, i_2, \ldots, i_n, \; \forall x(0) \in V_{s1}$. So $V_r^{n-1}(x(0))=V_{sk}, \; \forall x(0) \in V_{s1}$ where $V_{sk} \subset V_C$ and we use the notation $V_{s1} \rightsquigarrow V_{sk}$.

STEP 3 Because of invariance of W_I, all initial states in V_{s1} are mapped to a variety of special form which is a subset of V_C. Note that V_C can contain only finitely many varieties of special form $V_{si}, i=1, 2, \ldots, B$. Thus, there exists $i=1, 2, \ldots, B$ such that V_{s1} is mapped to V_{si}. If $i=1$, then V_{s1} is a strongly invariant set. If not, assume that $i=2$. Because of invariance, there exists $i=1, 2, \ldots, B$ such that V_{s2} is mapped to V_{si}. If $i=1$ or 2 we have constructed a strongly invariant set $V_{s1} \cup V_{s2}$. If not, assume $i=3$, etc. Therefore, we have $V_{s1} \cup \ldots V_{sL} \subset W_I, \; L \leq B$.

STEP 4 Suppose that the strongly invariant set can be decomposed as $W_I=V_{s1} \cup \ldots V_{sL} \cup S$, where $S \not\subset \cup_i V_{si}$. Any point in S is mapped to one of $V_{si}, i=1, 2, \ldots, L$ because of invariance of W_I but the points of S can not be reached from V_{si}. If the set S were not empty, W_I would not be strongly invariant. Q.E.D.

Using arguments very similar to the proof of Lemma 6.1, we can prove the following three lemmas.

Lemma 6.2 *Every invariant set must contain a strongly invariant set.* □

Proof of Lemma 6.2: Suppose that $V_I \subseteq V_C$ is an invariant set and that it does not contain any strongly invariant subsets. If $x(0) \in V_I$ then because of invariance of V_I we have that $V_r^{n-1}(x(0)) \subset V_I$ and we can denote it as V_{s1}. Notice that there may be at most B varieties of special form contained in V_I. Using the property proved in Step 2 of the previous lemma, we have that $V_{s1} \rightsquigarrow V_{si},\ i{=}1,2,\ldots,B$. However, since we assumed that there are no strongly invariant sets in V_I, we must have that $i{\neq}1$. Therefore, $V_{s1} \rightsquigarrow V_{si},\ i{=}2,\ldots,B$, and we can assume $i{=}2$. Using the same argument we have that $V_{s2} \rightsquigarrow V_{si},\ i{=}3,\ldots,B$ and we can assume that $i{=}3$, etc. After $B-1$ steps we obtain that $V_{sB} \rightsquigarrow V_{si},\ i{=}1,2,\ldots,B$ because of invariance of V_I but this contradicts the assumption that there are no strongly invariant sets contained in V_I. The contradiction completes the proof. Q.E.D.

Lemma 6.3 *Every state in $V_C - V_I$ can be transferred to $\mathbb{R}^n - V_C$ in finite time.* □

Proof of Lemma 6.3: The Lemma follows trivially from the definition of the maximal invariant set V_I. Q.E.D.

Lemma 6.4 *Every state in $V_I - W_I$ is transferred to a strongly invariant set W_I in finite time.* □

Proof of Lemma 6.4: Assume that $W_I{\neq}\emptyset$. We assume that there exists a state $x(0) \in V_I - W_I$ such that $x(k, x(0), \mathcal{U}_{k-1}) \in V_I - W_I, \forall k, \forall \mathcal{U}_{k-1}$. In this case it follows that the set $V_I - W_I$ contains an invariant subset and from Lemma 6.2 it follows that $V_I - W_I$ contains a strongly invariant subset. Hence, we have that W_I is not the maximal strongly invariant set. The contradiction completes the proof. Q.E.D.

We can combine these Lemmas 6.1, 6.2, 6.3 and 6.4, to obtain the following result:

Theorem 6.1 *The odd polynomial system (6.2) is output dead-beat controllable if and only if either $W_I{=}\emptyset$ or every variety of special form contained in the maximal strongly invariant set W_I intersects the zero output variety V_O.* □

Proof of Theorem 6.1:

Necessity: Suppose that there exists a variety of special form V_s contained in the maximal strongly invariant set which is such that its intersection with V_O is empty. If the variety V_s is a strongly invariant set itself then there is no control sequence which transfers any initial state in V_s

to V_O. If V_s is a subset of a larger strongly invariant set W_I^* and $V_s \cap V_O = \emptyset$ then $W_I^* \cap V_O = \emptyset$ because of Assumption 6.1 and the same argument applies.

Sufficiency: We partition the whole state space $\mathbb{R}^n = (V_C - V_I) \cup (V_I - W_I) \cup W_I \cup (\mathbb{R}^n - V_C)$ and consider what happens on each of the subsets. If $x(0) \in \mathbb{R}^n - V_C$ we can zero the output in one step. If $x(0) \in V_C - V_I$, according to Lemma 6.3, it follows that the initial state can be transferred to $\mathbb{R}^n - V_C$ in finite time and consequently to V_O. Consider $x(0) \in V_I - W_I$. From Lemma 6.4 it follows that $x(0)$ is transferred to W_I in finite time. Since all irreducible components of W_I intersect V_O and because of Assumption 6.2 it follows that any state in V_I can be transferred to V_O in finite time. Because of Assumption 6.2 we conclude that the system is output dead-beat controllable. Q.E.D.

The following corollaries may help us to reduce computations even more.

Corollary 6.1 *If* $\dim V_C = \dim V(a_m) < n-1$ *the system is output dead-beat controllable.* □

Proof: Since $\dim V_r^{n-1}(x(0)) = n-1, \forall x(0) \in \mathbb{R}^n$, it follows that $V_r^{n-1}(x(0)) \not\subset V_C, \forall x(0) \in V_C$. Thus, we need at most n steps to map any initial state to V_O. Q.E.D.

It is possible to use the method based on the affine Hilbert polynomial (see the last chapter of [37]) in order to check the dimension of the variety V_C.

Corollary 6.2 *If V_C does not contain varieties of special form, that is a_m does not contain irreducible polynomials of special form, the system (6.2) is output dead-beat controllable.* □

Proof: From properties $V_r^{n-1}(x(0))$ it follows that $V_r^{n-1}(x(0))$ can not be a subset of $V_C, \forall x(0) \in V_C$. Q.E.D.

Corollary 6.3 *Suppose that there are B varieties of special form $V_{si} = V(f_{si})$ contained in V_C. The system (6.2) is output dead-beat controllable if $V_O \cap V_{si} \neq \emptyset$, $\forall i = 1, 2, \ldots, B$.* □

Comment 6.1 It is important to notice that Theorem 6.1 provides conditions for output controllability to the hyperplane $x_1 = 0$. If we want to check output controllability to some other point $x_1 = y^*$, $y^* \neq 0$ then all irreducible components (varieties) V_{si} of the maximal strongly invariant set W_I should intersect the hyperplane $x_1 = y^*$. □

Comment 6.2 Theorem 6.1 is very similar to the results for odd scalar polynomial systems that are analysed in Chapter 5. The periodic points and equilibria represent special forms of strongly invariant sets in the case of scalar polynomial systems. The target set in Chapter 5 is the origin whereas the target set for odd polynomial systems in this chapter is the hyperplane x_1=0. Moreover, in this chapter the dimension of the $n-1$ step reachable set is always $n-1$ whereas for scalar polynomial systems it can be 1 (n) or 0 ($n-1$). All differences between the main results in Chapters 5 and 6 come from these facts. □

Comment 6.3 The phenomena that are described in this chapter, that is invariant and strongly invariant sets, play an important role in the characterisation of other controllability properties. Indeed, we may have loss of state dead-beat or complete controllability if there exist some invariant sets. Moreover, invariant sets are important for controllability properties of even systems as well. The following example illustrates our claim. Consider the system:

$$y(k+1) = (y(k) - u^3(k-1))\, u^4(k) + u^3(k)$$

Obviously there is a strongly invariant set $W_I = \{(y, v) : y - v^3 = 0\}$. Notice that the output can be zeroed from the states that belong to the strongly invariant set. Moreover, the system is output and state dead-beat controllable. However, it is not completely controllable since any state that belongs to the strongly invariant set W_I can not be mapped outside the set. □

Comment 6.4 Even systems may exhibit other forms of invariance. Indeed, the system:

$$\begin{aligned} x_1(k+1) &= u(k) \\ x_2(k+1) &= x_2(k) - x_1^2(k) + u^2(k) \end{aligned}$$

has one invariant variety $V(x_2 - x_1^2)$ and two invariant semi-algebraic sets $I_1 = \{x \in \mathbb{R}^2 : x_2 < x_1^2\}$ and $I_2 = \{x \in \mathbb{R}^2 : x_2 > x_1^2\}$.

One way to check the existence of semialgebraic invariant sets is to use the methodology described in Chapter 3 which is based on the QEPCAD algorithm. The test for the existence of invariant semialgebraic sets is more computationally expensive and this is one of the main hindrances to a more complete investigation of controllability properties of even systems. □

6.4 *Output Dead-Beat Controllability Tests*

So far we have considered what happens geometrically, whereas an algebraic test is needed to check the conditions of Theorem 6.1. From Lemma 6.1 and the definition of strongly invariant sets, we can deduce the following method to check output dead-beat controllability of systems (6.2).

TEST:

1. Check Assumptions 6.1 and 6.2. Assumption 6.2 is checked using QEPCAD by considering the following decision problem

$$(\forall x)\,(\exists u)\,[h(x)=0 \wedge f_u \circ h=0]\,.$$

2. Decompose the polynomial $a_m \in \mathbb{Q}[x_1, \ldots, x_n]$ into irreducible polynomials (using eg. the command "factor" in Maple) and identify all polynomials that have special form. Denote this set as $\Sigma_1=\{f_{s1}, f_{s2}, \ldots, f_{sB}\}$.

3. (a) Check whether any of the varieties $V(f_{si})$, $i=1, 2, \ldots, B$ is invariant using the Gröbner basis method of Chapter 4. Denote the set of all polynomials f_{si} that yield invariant varieties as Σ_1^I. Obviously $\Sigma_1^I \subseteq \Sigma_1$. Define the set $\Sigma_2=\Sigma_1 - \Sigma_1^I$.

   (b) If $\Sigma_2\neq\emptyset$, find all products $f_{sj} \cdot f_{sk},\ f_{sj}, f_{sk} \in \Sigma_2$, and check the invariance of all varieties $V(f_{sj} \cdot f_{sk})$ using the Gröbner basis method. The set of all polynomials for which varieties $V(f_{sj} \cdot f_{sk})$ are invariant is denoted as Σ_2^I. Obviously, $\Sigma_2^I \subseteq \Sigma_2$. Define a new set $\Sigma_3=\Sigma_2 - \Sigma_2^I$.

   (c) Find the sets $\Sigma_j^I, j=3, \ldots B-1$ in the same way.

   (d) If $\Sigma_B\neq\emptyset$ find the product $f_{s1} \cdot \ldots \cdot f_{sB}$ and check the invariance of the variety $V(f_{s1} \cdot \ldots \cdot f_{sB})$ using the Gröbner basis method. If the variety is invariant then $\Sigma_B^I=\Sigma_1$. Otherwise, $\Sigma_B^I=\emptyset$. Define the set $\Sigma^I=\cup_{i=1}^B \Sigma_i^I$. The maximal strongly invariant set is then

$$W_I=V(\prod_{f_{si}\in\Sigma^I} f_{si})$$

4. Check whether $V_O \cap V(f_{si})\neq\emptyset, \forall f_{si} \in \Sigma^I$ using QEPCAD by considering the decision problems:

$$(\exists x)\,[h=0 \wedge f_{si}=0]\,,\quad \forall i \ \text{ for which } f_{si} \in \Sigma^I$$

   If this is true, the system is output dead-beat controllable and vice versa.

Comment 6.5 It is very important to notice that this output dead-beat controllability test stops after a finite number of operations. This was not the case with the systems considered in Chapters 3 and 4 since the chain $S_0 \subset S_1 \subset \ldots$ may not terminate. In general, we can not say a priori when the chain terminates and hence we can not say whether the controllability test stops after a finite number of operations or not. The structure of the class of systems (6.2), however, guarantees that the test described above stops in finite time. □

Comment 6.6 Notice that this test gives us the partition of the strongly invariant set into strongly invariant subsets. In other words, we find all strongly invariant sets. □

It is possible to derive another output dead-beat controllability test which uses polynomial divisions. We present the test only for second order systems, but it is straightforward to generalise the test to systems of arbitrary order.

Indeed, consider the system:

$$y(k+1) = F(y(k), u(k-1), u(k)) \tag{6.9}$$

where $F(y,v,u) = a_m(y,v)\,u^m + \ldots + a_0(y,v)\,, a_i \in \mathbb{Q}[y,v]$. We use the same definitions and assumptions as in the previous section. Then we can state the following test which can be used to check the existence of strongly invariant sets. In this case polynomials of special form are given by: $y - \sum_{i=0}^{m-1} b_i^p v^i, b_i \in \mathbb{Q}$.

Lemma 6.5 *Consider the system (6.9). The critical variety V_C (6.4) contains a strongly invariant subset if and only if:*

1. There exist polynomials $y - \sum_{i=0}^{m-1} b_i^p v^i,\ b_i^p \in \mathbb{Q},\ p=1,2,\ldots,T,\ T \le L \le B$ such that

$$a_m(y,v)\,|(y - \sum_{i=0}^{m-1} b_i^p v^i)\,, \forall p=1,2,\ldots,T$$

2.

$$a_i(y,v) \equiv b_i^{p+1}|(y - \sum_{i=0}^{m-1} b_i^p v^i)\,, \forall p=1,2,\ldots,T-1, \forall i=1,\ldots,m-1$$

and

$$a_i(y,v) \equiv b_i^1|(y - \sum_{i=0}^{m-1} b_i^T v^i)\,,\ \forall i=1,\ldots,m-1$$

Proof of Lemma 6.5: Suppose that $V_I \neq \emptyset$. Hence, $W_I \neq \emptyset$. It follows that there exist polynomials of special form f_{si} which divide a_m. We can actually write

$$a_m | f_{s1}^{n_1} \cdot \ldots f_{sB}^{n_B}$$

and

$$V_{sp} = \{x \in \mathbb{R}^2 : y - \sum_{i=0}^{m-1} b_i^p v^i\},\ \forall p = 1, \ldots, B.$$

Only L varieties of special form are contained in W_I and without loss of generality we may assume that the first L varieties $V_{si}, i=1,2,\ldots,L$ are contained in W_I. Consider an initial state $x(0) \in V_{s1}$. From the proof of Lemma 6.1 we see that there exists $p \in \{1,2,\ldots,B\}$ such that $a_i(y,v) = b_i^p, \forall i = 0,1,\ldots,m-1,\ \forall (y,v) \in V_{s1}$. Without loss of generality assume that $p=2$. Hence, if we regard $a_i(y,v)$ as functions on the variety V_{s1} we necessarily have that they are the same as the constant functions $b_i^2, \forall i$. We denote

$$I(V_{s1}) = \{f \in \mathbb{Q}[y,v] : f(y,v) = 0\ \ \forall (y,v) \in V_{s1}\}$$

It can be shown that $I(V_{s1})$ is an ideal [37, pg. 32]. Moreover, $I(V_{s1})$ is the radical ideal of $\langle f_{s1}^{n_1} \rangle$. In other words, $I(V_{s1}) = \langle f_{s1} \rangle = \langle y - \sum_{i=0}^{m-1} b_i^1 v^i \rangle$ [37, pp.175-179].

Finally, we have from [37, pg.215] that $a_i(y,v)$ and b_i^2 represent the same polynomial function on the variety V_{s1} if and only if

$$a_i(y,v) - b_i^2 \in I(V_{s1}) = \langle y - \sum_{i=0}^{m-1} b_1^p v^i \rangle$$

In other words, this is true if and only if

$$a_i(y,v) - b_i^2 = h_i(y,v)\,(y - \sum_{i=0}^{m-1} b_i^1 v^i)\,.$$

Hence, we have

$$a_i(y,v) \equiv b_i^2 | (y - \sum_{i=0}^{m-1} b_i^1 v^i)\,,\ \forall i = 1,2,\ldots,m-1.$$

Also, we know that necessarily:

$$a_m(y,v) \,|\, (y - \sum_{i=0}^{m-1} b_i^p v^i)\,,\ p = 1,2.$$

Using a similar argument as in the proof of Lemma 6.1 we obtain in general that

$$V_{s1} \rightsquigarrow V_{s2} \rightsquigarrow \ldots \rightsquigarrow V_{sB} \rightsquigarrow V_{s1} \rightsquigarrow ad\ infinitum$$

and hence we obtain the above formulas. Q.E.D.

Comment 6.7 The first step when checking whether the critical variety V_C has invariant subsets when using the second test is to find all polynomials of special form $y - \sum_i b_i v^i$ that divide the polynomial $a_m(y, v)$. This can be done using "factor" command in Maple. Having found all irreducible varieties of special form, we can write:

$$V_{sp}=\{(y,v)\in\mathbb{R}^2 : y - \sum_{i=1}^{m-1} b_i^p v^i=0\},\ p=1,\ldots,B.$$

The second step is to check whether

$$a_i(y,v) \equiv b_i^p | (y - \sum_{i=1}^{m-1} b_i^l v^i)\ ,\ \forall i=0,1,\ldots,m-1,\ \forall p,l=1,2,\ldots,B.$$

Then, the last step is to verify the conditions of Lemma 6.5. □

Comment 6.8 One can easily verify that the conditions under which the critical variety V_C may contain invariant subsets (for second order odd systems they are given in Lemma 6.5) are not generic. It follows that output dead-beat controllability is a generic property for odd systems considered in this chapter. □

6.5 Examples

Example 6.1 Consider the system:

$$\begin{aligned} y(k+1) &= (y^2(k) - 2y(k)\, u^2(k-1)\, u^2(k-2) - 3y(k) + u^4(k-1)\, u^4(k-2) \\ &\quad + 3u^2(k-1)\, u^2(k-2) + 2)\, u(k)^3 + u(k)^2\, u(k-1)^2 - y(k) \\ &\quad + u(k-1)^2\, u(k-2)^2 + 3 \end{aligned}$$

Introducing the state variables $x_1(k) = y(k)$, $x_2(k) = u(k-1)$ and $x_3(k) = u(k-2)$ we obtain the state space model:

$$
\begin{aligned}
x_1(k+1) &= (x_1^2(k) - 2x_1(k)\,x_2^2(k)\,x_3^2(k) - 3x_1(k) + x_2^4(k)\,x_3^4(k) + 3x_2^2(k)\,x_3^2(k) + 2) \\
&\quad u(k)^3 + u(k)^2\,x_2^2(k) - x_1(k) + x_2^2(k)\,x_3^2(k) + 3 \\
x_2(k+1) &= u(k) \\
x_3(k+1) &= x_2(k) \\
y(k) &= x_1(k)
\end{aligned}
$$

We are going to use the method based on the Gröbner basis algorithm for this example.

Step 1: Assumptions 6.1 and 6.2 are satisfied.

Step 2: Using the command "factor" in Maple for the polynomial $x_1^2 - 2x_1x_2^2x_3^2 - 3x_1 + x_2^4x_3^4 + 3x_2^2x_3^2 + 2$ we find that the only two polynomials of special form are $f_{s1} = x_1 - x_2^2x_3^2 - 1$ and $f_{s2} = x_1 - x_2^2x_3^2 - 2$. In other words, $V_{s1} = V(f_{s1}) \subset V_C$ and $V_{s2} = V(f_{s2}) \subset V_C$.

Step 3: We check whether the variety V_{s1} is invariant:

$$
\begin{aligned}
f_{s1} &= x_1 - x_2^2x_3^2 - 1 \\
f_{s1} \circ f_u(x) &= (x_1 - x_2^2x_3^2 - 1)\,(x_1 - x_2^2x_3^2 - 2)\,u^3 - x_1 + x_2^2x_3^2 + 2 \\
G_0 &= \{x_1 - x_2^2x_3^2 - 1\} \\
G_1 &= \text{Gbasis}[x_1 - x_2^2x_3^2 - 1, (x_1 - x_2^2x_3^2 - 1)\,(x_1 - x_2^2x_3^2 - 2)\,, -x_1 + x_2^2x_3^2 + 2] \\
&= \{1\}
\end{aligned}
$$

and since $G_2 = \langle 1 \rangle$ it follows that V_{s1} is not invariant. Similarly, we have for variety V_{s2}:

$$
\begin{aligned}
f_{s2} &= x_1 - x_2^2x_3^2 - 2 \\
f_{s2} \circ f_u(x) &= (x_1 - x_2^2x_3^2 - 1)\,(x_1 - x_2^2x_3^2 - 2)\,u^3 - x_1 + x_2^2x_3^2 + 1 \\
G_0 &= \{x_1 - x_2^2x_3^2 - 2\} \\
G_1 &= \text{Gbasis}[x_1 - x_2^2x_3^2 - 2, (x_1 - x_2^2x_3^2 - 1)\,(x_1 - x_2^2x_3^2 - 2)\,, -x_1 + x_2^2x_3^2 + 1] \\
&= \{1\}
\end{aligned}
$$

Therefore, V_{s2} is not invariant. Consider now the vari ety $V(f_{s1} \cdot f_{s2})$. We obtain:

$$\begin{aligned}
f_{s1} \cdot f_{s2} &= (x_1 - x_2^2 x_3^2 - 1)(x_1 - x_2^2 x_3^2 - 2) \\
(f_{s1} \cdot f_{s2}) \circ f_u(x) &= [(x_1 - x_2^2 x_3^2 - 1)(x_1 - x_2^2 x_3^2 - 2)u^3 - x_1 + x_2^2 x_3^2 + 2] \\
&\quad [(x_1 - x_2^2 x_3^2 - 1)(x_1 - x_2^2 x_3^2 - 2)u^3 - x_1 + x_2^2 x_3^2 + 1] \\
G_0 &= \{(x_1 - x_2^2 x_3^2 - 1)(x_1 - x_2^2 x_3^2 - 2)\} \\
G_1 &= \text{Gbasis}[(x_1 - x_2^2 x_3^2 - 1)^2 (x_1 - x_2^2 x_3^2 - 2)^2, (x_1 - x_2^2 x_3^2 - 1) \\
&\quad (x_1 - x_2^2 x_3^2 - 2)^2, (x_1 - x_2^2 x_3^2 - 1)^2 (x_1 - x_2^2 x_3^2 - 2), (x_1 - x_2^2 x_3^2 - 1) \\
&\quad (x_1 - x_2^2 x_3^2 - 2)] = \{(x_1 - x_2^2 x_3^2 - 1)(x_1 - x_2^2 x_3^2 - 2)\}
\end{aligned}$$

Since $G_0 = G_1$, we conclude that the variety $V(f_{s1} \cdot f_{s2})$ is invariant. It is not difficult to see that we actually have that $V_{s1} \rightsquigarrow V_{s2} \rightsquigarrow V_{s1} \rightsquigarrow \ldots$.

Step 4: We do not need to use QEPCAD in this case. Indeed, since the equations $x_2^2 x_3^2 = -K$, $K = 1, 2$ have no real solutions in x_2, x_3, we conclude that $V_{s1} \cap V_O = \emptyset$ and $V_{s2} \cap V_O = \emptyset$ and consequently the system is not output dead-beat controllable. □

Example 6.2 The system is described by the input-output recurrence equation:

$$y(k+1) = [y(k) - (u(k-1)^2 + 1)]\, u(k)^3 + u(k)^2 + 1$$

Therefore we can write

$$f(y, v, u) = [y - (v^2 + 1)]\, u^3 + u^2 + 1\ .$$

Assumption 6.1 is satisfied since for $y = 0$ we have

$$0 = -(v^2 + 1)\, u^3 + u^2 + 1\ .$$

This equation has a real solution u, $\forall v \in \mathbb{R}$. The critical variety V_C is given by:

$$V_C = \{(y, v) \in \mathbb{R}^2 : y - (v^2 + 1) = 0\}$$

If $(y(0), u(-1)) \notin V_C$ we can regulate the output to zero in one step. If $(y(0), u(-1)) \in V_C$ we have that $y(k+1) = u(k)^2 + 1, \forall k$, that is $(y(k), u(k-1)) \in V_C$, $\forall k$. So, it is possible to map any point that belongs to V_C into any other point in V_C, but it is impossible to map them to the origin. The critical variety V_C is invariant. In this case, it is clear that the system is not output dead-beat controllable. Therefore, we have the situation $V_C \rightsquigarrow V_C \rightsquigarrow V_C \rightsquigarrow \ldots$. □

Example 6.3 Consider the system:

$$\begin{aligned} y(k+1) &= (y(k) - u(k-1)^2 - 1)\,(y(k) + u(k-1)^2 + 1) \\ &\quad [(y(k)+2)\,u(k)^3 + u(k)^2 + 1] + u(k)^2 + 1 \end{aligned} \tag{6.10}$$

Assumption 6.1 is satisfied. The critical variety V_C is defined by:

$$V_C = \{(y,v) \in \mathbb{R}^2 : (y - v^2 - 1)\,(y + v^2 + 1)\,(y+2) = 0\}$$

In this case we may verify that the only strongly invariant set is given by:

$$W_I = \{(y,v) \in \mathbb{R}^2 : (y - v^2 - 1) = 0\} \subset V_C$$

Indeed, $W_I \rightsquigarrow W_I$, hence t in Definition 6.7 can be chosen to be 1. From equation (6.10) it is clear that $\forall (y,v) \in V_1$, where $V_1 = \{(y,v) \in \mathbb{R}^2 : (y + v^2 + 1) = 0\}$ (see Figure 6.1) we have $V_r(y,v) = W_I$. Therefore, any initial state in V_1 is transferred in one step to some point in W_I irrespective of the control that is applied. Thus, we can write:

$$V_1 \rightsquigarrow W_I \rightsquigarrow W_I \rightsquigarrow \ldots$$

Consider now initial states on the line $y(0) = -2$. The model of the system becomes:

$$y(1) = [(-3 - u(-1)^2)\,(-1 + u(-1)^2) + 1]\,(u(0)^2 + 1)\,.$$

Denote real solutions $u(-1)$ of the following equations:

$$[(-3 - u(-1)^2)\,(-1 + u(-1)^2) + 1] = -1$$

$$[(-3 - u(-1)^2)\,(-1 + u(-1)^2) + 1] = 1$$

as c_i and b_i ($i = 1, 2$), respectively. The set of one step reachable states from $(-2, c_1)$ and $(-2, c_2)$ is V_1 and from $(-2, b_1)$ and $(-2, b_2)$ is W_I. Notice also that $b_1 = 1,\ b_2 = -1$ and hence $(-2, b_1)$

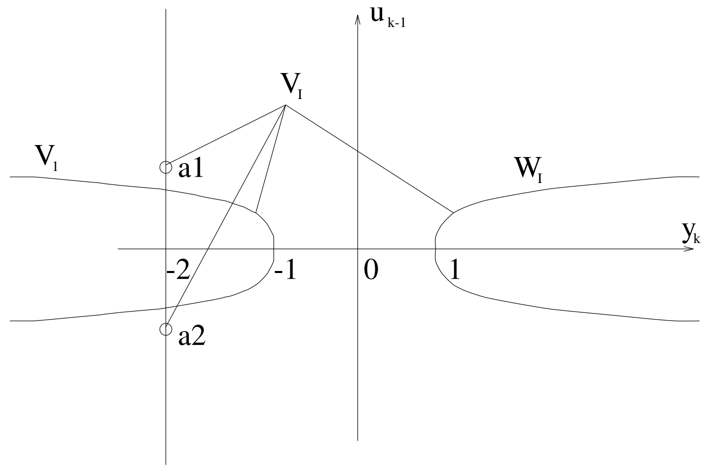

Figure 6.1: Invariant sets V_I and strongly invariant sets W_I

and $(-2, b_2)$ belong to V_1. Therefore, we can write:

$$(-2, a_i) \rightsquigarrow V_1 \rightsquigarrow W_I \rightsquigarrow W_I \rightsquigarrow \ldots,\ i=1,2$$

The maximal invariant set V_I is:

$$V_I=\{(y,v)\in\mathbb{R}^2 : (y-v^2-1)(y+v^2+1)=0\}\cup\{(-2,c_1),(-2,c_2)\}.$$

Sets V_I and W_I are shown in Figure 6.1. The set $V_C - V_I$ is not invariant and there exists a control $u(k)$ which can map any initial state from it to $\mathbb{R}^2 - V_C$ in one step. Observe that both V_I and W_I are real varieties, whereas $V_C - V_I$ is not. Also, initial states in V_I are transferred to W_I in one step and the initial states $(-2, c_i)$, $i=1,2$ are transferred to W_I in two steps. □

The following example shows a situation when the critical variety V_C does not contain invariant subsets.

Example 6.4 Consider the system:

$$y(k+1)=[y(k)-(u(k-1)^2+1)]\,u(k)^3+y(k)\,u(k)^2+1$$

Assumption 6.1 is satisfied and the critical variety V_C on which the highest order coefficient a_3 vanishes is given by:

$$V_C=\{(y,v)\in\mathbb{R}^2 : y-(v^2+1)=0\}$$

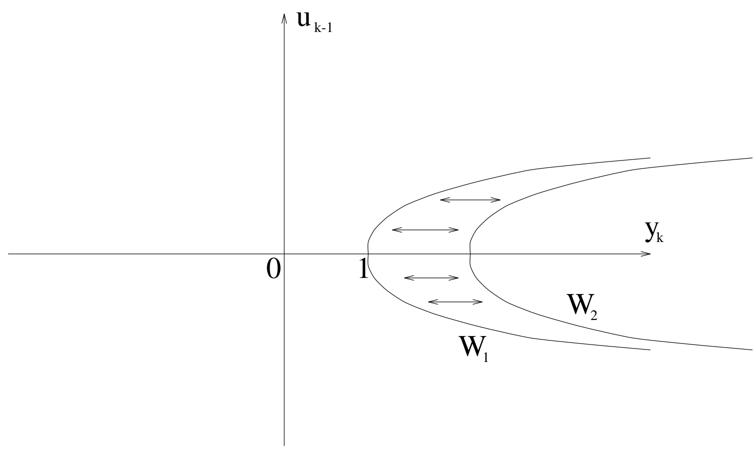

Figure 6.2: Periodic behaviour $V_{s1} \rightsquigarrow V_{s2} \rightsquigarrow V_{s1} \rightsquigarrow ad\ infinitum$

We see that the only variety of special form contained in V_C is V_C itself and it is defined by the polynomial $y-(v^2+1)$. Notice that

$$a_0{=}1,\ a_1{=}0,\ a_2{=}y$$

and therefore we have

$$a_0 \equiv 1|[y-(v^2+1)]\,;\ \ a_1 \equiv 0|[y-(v^2+1)]\,;\ \ a_2 \equiv 1 \not| [y-(v^2+1)]\,.$$

Since a_2 is not divisible by $[y-(v^2+1)]$ modulo 1, the system does not have the form given in Lemma 6.5 and therefore the critical variety V_C is not invariant. In this case we have three step output controllability. This can be easily verified by considering the first three iterations of the map f. After some calculations one obtains:

1. If $(y(0)\,,u(-1)\,) \notin V_C$, we can zero the output in one step.

2. If $(y(0)\,,u(-1)\,) \in V_C$, and $(y(0)\,,u(-1)\,) \neq (1,0)$ we have two step zeroing of the output.

3. If $(y(0)\,,u(-1)\,) = (1,0)$ the set of states reachable from this initial state is V_C. So, in the first step we have to map $(1,0)$ to some other state and then in the second step outside of the critical variety. Finally, we can zero this state in the third step.

Therefore, the system is 3-step output dead-beat controllable. □

Example 6.5 Consider the system:

$$y(k+1)=(y(k)-u(k-1)^2-1)\,(y(k)-u(k-1)^2-2)\,u(k)^3+u(k)^2-y(k)+u(k-1)^2+3$$

We can write:

$$a_3=(y-v^2-1)\,(y-v^2-2)\,,\ a_2=1,\ a_1=0,\ a_0=-y+v^2+3$$

and therefore

$$b_0^1=1, b_1^1=0, b_2^1=1,\ \ b_0^2=2,\ b_1^2=0,\ b_2^2=1.$$

It is easily checked that:

$$a_2\equiv 1|(y-v^2-1)\,;\ \ a_1\equiv 0|(y-v^2-1)\,;\ \ a_0\equiv 2|(y-v^2-1)$$

$$a_2\equiv 1|(y-v^2-2)\,;\ \ a_1\equiv 0|(y-v^2-2)\,;\ \ a_0\equiv 1|(y-v^2-2)$$

and therefore the critical variety V_C contains an invariant subset. The strongly invariant set of V_C is in this case the whole $V_C=\{(y,v)\in\mathbb{R}^2:(y-v^2-1)\,(y-v^2-2)=0\}$. Its irreducible components are $V_{s1}=\{(y,v)\in\mathbb{R}^2:(y-v^2-1)=0\}$ and $V_{s2}=\{(y,v)\in\mathbb{R}^2:(y-v^2-2)=0\}$ (see Figure 6.2). The irreducible varieties V_{s1} and V_{s2} do not intersect the line $y=0$ and therefore the system is not output dead-beat controllable.

A very interesting phenomenon can be illustrated by this example. We can see that the set of one step reachable states for any initial state in V_{s1} is V_{s2} and vice versa. Therefore, we can write:

$$V_{s1}\rightsquigarrow V_{s2}\rightsquigarrow V_{s1}\rightsquigarrow ad\ infinitum.$$

This means that if the initial state belongs to W_I, the trajectory periodically oscillates between V_{s1} and V_{s2}. Also, the set of one step reachable outputs from any initial state in V_{s1} is $[2,+\infty[$ and $\Pi V_r(y(0)\,,u(-1)\,)=[1,+\infty[,\ \forall(y(0)\,,u(-1)\,)\in V_{s2}$. For example, if the initial state belongs to V_{s1}, the set of reachable outputs changes periodically as:

$$y(k)\in[2,+\infty[, k=1,3,5,\ldots\ \text{ or }\ y(k)\in[1,+\infty[, k=2,4,\ldots.$$

□

6.6 Conclusions

We have presented necessary and sufficient conditions for the existence of output dead-beat controllability for a class of discrete-time systems described by a single input output polynomial equation. Two different output dead-beat controllability tests are presented. They are computationally less expensive than the tests of Chapters 3 and 4 and moreover they are finitely computable.

A number of interesting phenomena are observed. They shed more light on the properties of polynomial systems and contribute to a better understanding of output dead-beat controllability properties of general polynomial systems. Indeed, the interplay between the invariant sets and the strongly invariant sets gives a lot of insight into the output dead-beat problem. Moreover, the decomposition of the maximal strongly invariant set into varieties of special form helps us reduce computations considerably.

The results in this chapter illustrate the trade-off between the complexity of the controllability test and the generality of the considered class of systems. The tests that are presented in Chapter 3 do not give much insight into the underlying phenomena but the class of systems is much larger. On the other hand, in this and the previous chapters we characterised a number of new geometric and algebraic conditions which determine output dead-beat controllability. However, the considered class of systems is less general than the systems with rational coefficients considered in Chapter 3.

Chapter 7

Simple Hammerstein Systems

7.1 Introduction

This chapter is dedicated to dead-beat control of the so called simple Hammerstein systems. Identification techniques for block oriented models often yield models of simple Hammerstein form [76]. These systems can be represented by the block diagram of Figure 7.1. The system consists of a linear dynamical block W and a static nonlinearity $f(u)$. The nonlinearity is very often a polynomial and we consider the case when the highest degree of u in $f(u)$ is an even integer. If the polynomial $f(u)$ has an odd degree, then the overall system is dead-beat controllable if and

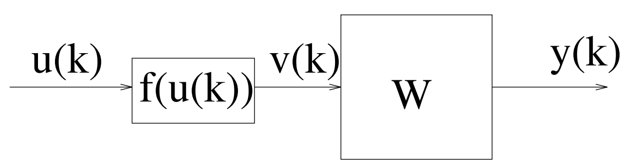

Figure 7.1: Block diagram of simple Hammerstein systems

only if the linear subsystem W is controllable from the input v. A minimum-time dead-beat controller for this class of simple Hammerstein systems immediately follows then from the design of the linear dead-beat controller [185].

If the polynomial $f(u)$ is of an even degree, the design is quite different from that for linear systems. Moreover, conditions for dead-beat controllability differ considerably from the known rank conditions for linear systems. The required analysis can be adopted from [50]. A minimum-time dead-beat control law is usually not unique for even input polynomials. Moreover, the obtained controllers are not continuous in general. The question of continuity of obtained control laws is important from a robustness point of view. We point out that this is still an open question.

The results of this chapter are directly applicable to SISO linear systems whose inputs are either positive or one side limited, that is they are not limited to polynomial input nonlinearities. Systems with positive controls are common in practice. For instance, systems such as biochemical reactors and some biological systems have one sided controls.

In the first part of this chapter we derive conditions for dead-beat controllability of linear systems with positive controls. The conditions are a straightforward consequence of results in [50]. Although obvious, the connection was not made explicit in the literature. In particular, we relax some of the conditions in [50], which consider complete (point-to-point) controllability, in order to obtain conditions pertaining to dead-beat controllability.

The second part of Chapter 7 contains the main results. We present a number of design methods for non-minimum and minimum-time dead-beat controllers for the above simple Hammerstein systems. First, we show that some simple design strategies can be used to design non-minimum-time dead-beat controllers for these systems. Although simple, the obtained control laws may not yield acceptable transient behaviour. The proposed non-minimum-time dead-beat controllers have constant structure on subsets (cones) of the state space. In particular, we apply on a cone the unconstrained time-optimal linear dead-beat controller and on the complement of the cone we apply a multi-rate feedback control law. We use the measured state at instant $k{=}0$ to determine *the control sequence* over a finite time horizon $k{=}0, \ldots, L$. In other words, we do not use the measurements of state at time instants $k{=}1, \ldots, L$. Hence, the control laws are such that the systems work in an open loop mode for a limited time for some initial states and hence we may expect that these do not perform well in the presence of disturbances

A family of minimum-time dead-beat controllers is designed. Parameters of the minimum-time dead-beat controllers can be easily changed to shape the transient response while preserving time optimality. The proposed time-optimal dead-beat controllers are nonlinear state feedback (on the whole state space) controllers. That is, control action at time k is a nonlinear function of the measured state at time k. We show by an example that the transient response of a time-optimal controller may be much better than that of the dead-beat controller without time optimality.

Similarity with the dead-beat controller of linear systems with bounded controls [174] is apparent. The linearity of systems equations allows us to compute the sets $S_k, k{=}0, 1, \ldots$ without resorting to QEPCAD. The sets S_k are in this case cones (not necessarily convex) and in the case of linear systems with bounded controls they are bounded convex subsets of state space.

Finally, we present a non-minimum-time output dead-beat controller for a class of simple

Hammerstein systems that are not necessarily state dead-beat controllable.

7.2 Notation and Definitions

For $\delta \in \mathbb{R}$, we write $\mathbb{R}_\delta^+ = [\delta, +\infty[$ and $\mathbb{R}_\delta^- =]-\infty, \delta]$. The class of nonlinear discrete-time systems that we consider can be written in the form:

$$\begin{aligned} x(k+1) &= Ax(k) + bf(u(k)); \; x(0); \; k=0,1,2,\ldots \\ y(k) &= cx(k) + df(u(k)) \end{aligned} \tag{7.1}$$

where $x(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}$ are respectively the state and the input of the system at time k. We consider the case $f(u) = \pm u^m + g_{m-1}u^{m-1} + \ldots + g_0$ and $m > 0$ is an even integer. In other words the image of f is $\mathrm{im}(f) = \mathbb{R}_\delta^i$, $i=+,-$ for some $\delta \in \mathbb{R}$. Obviously if we just introduce the transformation $f(u(k)) = v(k)$ we can regard the system (7.1) as a linear system with $v \in \mathbb{R}_\delta^i$, $i=+,-$ for some $\delta \in \mathbb{R}$.

Assumption 7.1 With reference to the system (7.1) the following holds:

$$\mathrm{rank}(\lambda I - A : b) = n, \; \forall \lambda \in \mathbb{C} - \{0\}.$$

□

In other words, we allow for uncontrollable modes that correspond to zero eigenvalues of matrix A. Assumption 7.1 corresponds to controllability to the origin for the linear system $x(k+1) = Ax(k) + bv(k)$, with $v \in \mathbb{R}$, clearly a necessary condition for dead-beat controllability of systems (7.1).

Assumption 7.2 With reference to the system (7.1), if $x(0) = 0$ then there exists $u(0) \in \mathbb{R}$ such that $x(1) = 0$. In other words, $\mathrm{im}(f) = \mathbb{R}_\delta^i$ with $\delta \le 0$ for $i=+$ and $\delta \ge 0$ for $i=-$. □

We use the following notation for a cone $C = \{x : x = \sum_i^r c_i v_i, \; v_i \in \mathbb{R}_0^+\}, c_i \in \mathbb{R}^{n\times 1}, i=1,\ldots,r$. It is obvious that a cone can be also defined by r inequalities $l_i x \ge 0, l_i \in \mathbb{R}^{1\times n}, i=1,2,\ldots,r$.

7.3 Dead-Beat Controllability

In this section we discuss some results from [50] and apply them to dead-beat controllability of simple Hammerstein systems (7.1). The following theorems play a crucial role for dead-beat controllability of simple Hammerstein systems.

Theorem 7.1 *[50] The system (7.1) with* $\mathrm{im} f(u) = \mathbb{R}_0^+, k = 0, 1, \ldots$ *is completely controllable on* $\mathbb{R}^n$ *in the sense of Definition 2.3 if and only if*

1. *rank*$[b\ Ab\ \ldots\ A^{n-1}b] = n$,

2. *the matrix* A *has no real positive or zero eigenvalues* □

Theorem 7.2 *[174] The system*

$$x(k+1) = Ax(k) + bu(k)\,,\ \text{with}\ u(k) \in [-1, 1]$$

is dead-beat controllable (see Definition 2.1) if and only if

1. *the unconstrained system (with* $u \in \mathbb{R}$*) is dead-beat controllable,*

2. A *has all its eigenvalues in the closed unit circle* $|\lambda| \leq 1$ □

The following theorem is a consequence of Theorem 7.1. Its proof is contained in the proof of the Theorem 7.1 in [50].

Theorem 7.3 *[50] If the second condition of Theorem 7.1 is satisfied, there exists a polynomial with positive coefficients* $c(\lambda) = \sum_{i=0}^{L} c_i \lambda^i, c_i \geq 0, \forall i = 0, 1, \ldots, L$ *such that* $c(A) = 0$. □

This leads to the following result.

Theorem 7.4 *The system (7.1) with* $\mathrm{im}(f) = \mathbb{R}_0^+$ *(or* $\mathbb{R}_0^-$*) is dead-beat controllable if and only if*

1. *Assumption 7.1 holds*

2. A *has no strictly positive real eigenvalues* □

Proof of Theorem 7.4:

Necessity: Necessity of the first point is obvious and the second point follows directly from [50]. Indeed, using the real canonical form for matrices A and b which was introduced in [50] and assuming that there exists a positive real root of A, which we denote λ_1 with algebraic multiplicity n_1 ($\lambda_1 > 0$), then there exists a transformation of coordinates which yields a subsystem of the original system (7.1) of the form:

$$z(k+1) = \begin{pmatrix} \lambda_1 & \delta_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_1 & \delta_2 & 0 & \ldots & 0 \\ & \ldots & & & \ldots & \\ 0 & 0 & 0 & 0 & \ldots & \lambda_1 \end{pmatrix} z(k) + \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_{n_1} \end{pmatrix} f(u(k)) \tag{7.2}$$

where δ_i=1 or 0 depending on the algebraic and geometric multiplicities of λ_1. Since Assumption 7.1 holds we necessarily have that $b_{n_1}\neq 0$. Suppose that $b_{n_1} > 0$. Notice that for any sequence of controls we have that $z_{n_1}(N) = \lambda_1^N z_{n_1}(0) + \sum_{i=0}^{N-1} \lambda_1^{N-1-i} b_{n_1} f(u(i))$. Therefore, if $z_{n_1}(0) > 0$ we have that $z_{n_1}(N, z_{n_1}(0), \mathcal{U}_{N-1}) > 0, \forall N, \forall \mathcal{U}_{N-1}$. It follows that the system is not dead-beat controllable.

Sufficiency: Assume now that the conditions of Theorem 7.4 are satisfied. We can write the system in Jordan canonical form:

$$x(k+1) = \begin{pmatrix} A_1 & 0 \\ 0 & A_0 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ b_0 \end{pmatrix} f(u(k))$$

where A_1 corresponds to the non zero eigenvalues and A_0 is the Jordan block corresponding to zero eigenvalues. There exists an integer r such that A_0^r=0. By assumption, (A_1, b_1) is controllable and A_1 does not have any positive or zero eigenvalues. From Theorem 7.1 it follows that for any initial condition $x(0)$ there exists a sequence $u(0), u(1), \ldots, u(T-1)$ such that $x(T) = (0 \;\; x^*)^T$, dim x^*=dim b_0. Applying henceforth $u(T+i) = 0$, i=0, 1, 2, ... we have $x(T+k) = 0, \forall k \geq r$. Notice that the pair (A_0, b_0) does not have to be controllable. Q.E.D.

We emphasize that there is no loss of generality if we assume that

- the pair (A, b) is controllable and
- A is full rank

In the sequel, we assume that *the pair (A, b) is in controllability canonical form.*

Comment 7.1 When the conditions of Theorem 7.1 are satisfied there exists a uniform bound on the dead-beat time for all initial states. In other words, there exists an integer $T \in \mathbb{N}$ such that $\forall x(0) \in \mathbb{R}^n$, $\exists \mathcal{U}=\{u(0), u(1), \ldots\}$ which yields $x(k)=0$, $\forall k \geq T$ (T is fixed for all initial states).

Notice, however, that under the conditions of Theorem 7.2, no such bound exists. Moreover, it is not difficult to see that if $\mathrm{im}(f)=\mathbb{R}_\delta^+$, $\delta < 0$ (in Theorem 7.1) it is possible to have real eigenvalues of A, $\lambda \in]0,1]$. In this case there is no uniform bound on the dead-beat time and, in principle, on a subspace of the state space we have the situation that the further the initial state from the origin, the longer the time required to zero that state.

The two types of dead-beat behaviour differ considerably. We will concentrate on the class of systems for which there is a **uniform bound on the dead-beat time**. If $\mathrm{im}(f)=\mathbb{R}_\delta^+$, $\delta < 0$ and $\exists\lambda(A) \in]0,1]$, it is possible to modify the design that we present to obtain a minimum-time dead-beat controller, but the design can only be carried out on a subset of the state space (no uniform bound on dead-beat time). □

7.4 State Dead-Beat Controllers

Most of the discussions presented in the next two subsections are on a rather elementary level but we use them to better illustrate the dead-beat controller designs.

7.4.1 Scalar Case

If we consider scalar Hammerstein systems $x(k+1)=ax(k)+bf(u(k)), a, b \in \mathbb{R}$, which are dead-beat controllable ($a \leq 0$), it is not difficult to see that the control law of the form: $u(k)$ is a real solution to $0=ax(k)+bf(u(k))$ if such a solution exists and $f(u(k))=0$; otherwise, it is a minimum-time dead-beat controller. The question is whether the same control law can be applied to higher order systems in order to obtain dead-beat behaviour. The question is motivated primarily by the simplicity of the controller.

7.4.2 Controller 1: Second Order Systems

The generalisation of the above control law to systems of higher order would be as follows. Find the minimum-time dead-beat controller for the unconstrained problem $x(k+1)=Ax(k)+bv(k)$. The controller is of the form $v(k)=Kx(k)$. Apply any real solution $u(k)$ to the equation

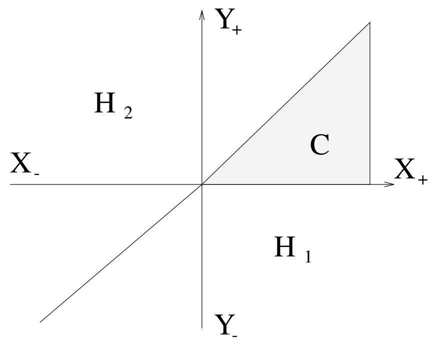

Figure 7.2: Sets H_1, H_2 and C

$Kx(k) = f(u(k))$ if such solution exists and $u(k) = 0$ otherwise. We prove below that this control law is indeed a non-minimum-time dead-beat control law for second order simple Hammerstein systems (7.1), but in general fails to be dead-beat for higher order systems. We emphasize that it suffices to consider the case when (A, b) is controllable and A has no zero or positive eigenvalues.

There is no loss of generality if we consider a second order system of the form:

$$x(k+1) = \begin{pmatrix} 0 & 1 \\ -\alpha & -\beta \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} f(u(k)), \; A = \begin{pmatrix} 0 & 1 \\ -\alpha & -\beta \end{pmatrix}, \; b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

It is obvious that the characteristic polynomial of the matrix A is $p^2 + \beta p + \alpha$. The minimum-time dead-beat controller for the unconstrained problem is given by $f(u(k)) = (\alpha \; \beta)\, x(k)$. We use the following notation:

$$H_1 = \{x : (\alpha \; \beta)\, x \geq 0\}; \; H_2 = \{x : (\alpha \; \beta)\, x < 0\}; \; C = \{x : (\alpha \; \beta)\, x \geq 0 \text{ and } (0 \; \alpha)\, x \geq 0\}$$

The introduced sets are shown in Figure 7.2.

Theorem 7.5 *A non-minimum-time dead-beat controller for second order Hammerstein systems (7.1) satisfying Assumptions 7.1 and 7.2 can be defined as:*

1. *apply any real solution $u(k)$ to the equation $Kx(k) = f(u(k))$ if there is such a solution*

2. *apply $u(k) = 0$ if there is no real solution to $Kx(k) = f(u(k))$*

The matrix gain K represents a minimum-time dead-beat controller for the unconstrained linear system $x(k+1) = Ax(k) + bv(k)$, $v \in \mathbb{R}$. □

In order to prove Theorem 7.5 we need several lemmas.

Lemma 7.1 *If the polynomial* $p^2 + \beta p + \alpha$ *has no positive real roots, the coefficients* α *and* β *satisfy the following relations:*

1. $\alpha \geq 0$ *and* $\beta \geq 0$*, or*

2. $\alpha > 0,\ \beta < 0$ *and* $\alpha > \beta^2/4$ □

Lemma 7.1 is proved by looking at the sign of the discriminant and using the Routh-Hurwitz criterion. The region to which the coefficients α and β must belong in order for the polynomial not to have positive real roots is shown in Figure 7.3.

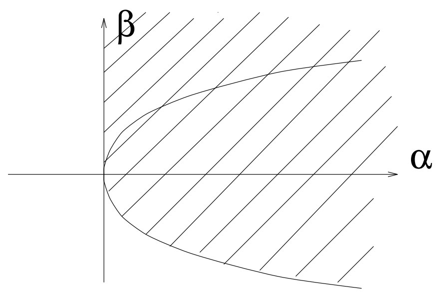

Figure 7.3: Domain for α and β for which A has no real positive eigenvalues

Lemma 7.2 *Consider the cone* $C \subseteq \mathbb{R}^2$ *defined by:*

$$C = \{x : (\alpha\ \beta)\, x \geq 0 \text{ and } (0\ \alpha)\, x \geq 0\}$$

The cone C *has a non empty interior in* $\mathbb{R}^2$*,* $\forall \alpha, \beta$ *satisfying the conditions of Lemma 7.1.* □

Proof of Lemma 7.2: The classification of all possible situations is given in Figure 7.4. Q.E.D.

Lemma 7.3 *Consider the system:*

$$x(k+1) = \begin{pmatrix} 0 & 1 \\ -\alpha & -\beta \end{pmatrix} x(k)\ , \ \textit{if}\ (\alpha\ \beta)\, x(k) < 0$$

$$x(k+1) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x(k)\ , \ \textit{if}\ (\alpha\ \beta)\, x(k) \geq 0$$

If α, β *satisfy the conditions of Lemma 7.1, then the following holds:*

1. $\forall x(0) \in C,\ x(k) = 0, \forall k \geq 2$

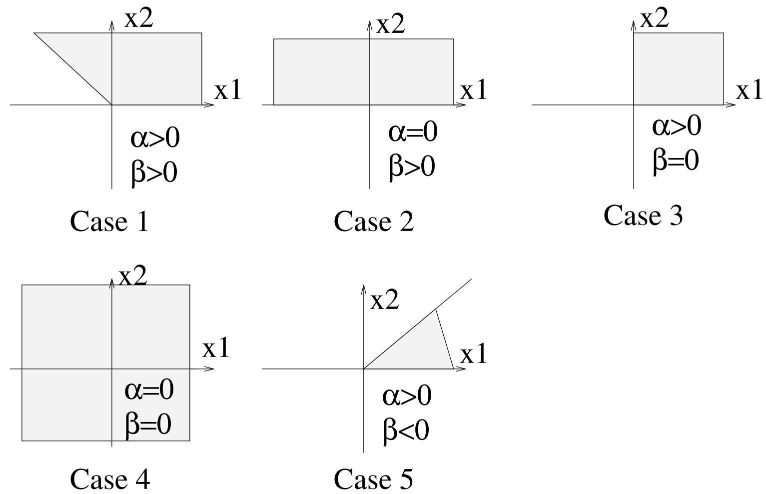

Figure 7.4: Cone classification

2. $\forall x(0) \in H_2, \exists L$ *such that* $x(L) \in C$ □

Proof of Lemma 7.3:

1. It follows trivially from the definition of the cone C and the fact that the time-optimal dead-beat controller for unconstrained linear system transfers any initial state of the unconstrained linear system to the origin in at most two time steps.

2. As the matrix A satisfies the condition of Theorem 7.1, it satisfies an equation $c(A)$ =0 where c is a monic polynomial with positive coefficients. Let the degree of c be L. It follows that A^L=$-\sum_{i=0}^{i=L-1} c_i A^i, c_i > 0, \forall i$=$0, \ldots, L-1$. Suppose that $x(k) \in H_2, \forall k$=$0, 1, 2, \ldots$. Then we have that $Kx(k) < 0, \forall k$. In other words, we have $Kx(0) < 0, KAx(0) < 0, \ldots$. Then, $KA^L x(0) < 0$ but since A^L can be expressed in terms of $-c_i A^i$, i=$0, 1, \ldots, L-1, c_i > 0$ we have a contradiction. Hence, $KA^L x(0) > 0$. We have shown that all initial states in H_2 are mapped to H_1 in finitely many steps. Consider Cases 1, 2, 3 and 5 of Lemma 7.2 (Case 4 is trivial). The cone C can be defined by inequalities $(\alpha\ \beta)\, x \geq 0$ and $x_2 \geq 0$. We know that for all initial states in H_2 there exists a time step L such that $x(L-1) \in H_2$ and $x(L) \in H_1$. Then using the fact that $x_2(k+1)$=$-Kx(k)$ we have by definition that $Kx(L) \geq 0$. Also, $x_2(L) > 0$ is automatically satisfied since $Kx(L-1) < 0$. Therefore, $x(L) \in C$. Q.E.D.

Theorem 7.5 does not hold for higher order systems. A counterexample (Example 7.2), which shows that the control strategy proposed in Theorem 7.5 does not realise dead-beat control, is presented in the next section.

7.4.3 Controller 2

If we base a control law again on the linear minimum-time dead-beat controller but instead of applying zero at the half space H_2 we apply $u{=}F(x)$, we obtain Controller 2. This controller is not time-optimal in general but it can be applied to any dead-beat controllable simple Hammerstein system. The main characteristics of this controller is a good design flexibility: we can modify the dead-beat time (lower bound is a property of the systems and can not be changed); for a given dead-beat time we can change the transient response. A drawback is that the controller consists of two different modes of operation. On a subset of the state space (C) the controller operates in a closed loop mode. On the complement of the set C the controller takes the measurement at time step k and then applies a *control sequence* which is computed on the basis of $x(k)$. In the meantime the system operates in an open loop mode. For a class of simple Hammerstein systems this control strategy yields a minimum-time controller which operates in closed loop mode on the whole state space.

The nilpotent matrix of dimension $n \times n$ is denoted as J.

$$J{=}\begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ & \ldots & & \ldots & \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix}$$

We point out that there is no loss of generality if concentrate only on completely controllable simple Hammerstein systems with a non singular matrix A.

Consider a completely controllable system (7.1) with the matrix A non singular. Suppose that the integer N represents the time such that any state $x(0) \in \mathbb{R}^n$ can be transferred to any $x^* \in \mathbb{R}^n$ in at most N steps. Such a number exists since the system is completely controllable.

We design now a minimum-time linear unconstrained $(v \in \mathbb{R})$ dead-beat controller $v(k){=}Kx(k)$. (A, b) is in controllable canonical form. Consider the cone:

$$C{=}\{x : Kx \geq 0, KJx \geq 0, \ldots, KJ^{n-1}x \geq 0\} \tag{7.3}$$

Notice that if the initial state is in the cone, it can be transferred to the origin in at most n time steps by applying the control action $u(k)$ which is equal to (any) real solution to the equation

$f(u(k))=Kx(k)$. Is is straightforward to show that C is actually equal to the set S_{n-1} (see Chapter 2 for the definitions of sets S_k). Hence, the design of a dead-beat controller can be formulated into the design of a controller which transfers any state outside of the cone to the cone in finite time. On the cone the linear controller $Kx(k)$ with an inversion yields minimum-time behaviour.

An important observation is that since A is not singular, the cone C has a non empty interior $\overset{\circ}{C}$ in $\mathbb{R}^n$. Indeed, non singularity of A implies that the matrix:

$$\begin{pmatrix} K \\ KJ \\ \dots \\ KJ^{n-1} \end{pmatrix}$$

has a full rank and the conclusion follows [153, 152]. Moreover, it is easy to show that the cone C is convex.

Since the system is completely controllable then for any $x^* \in \overset{\circ}{C}$ and for any positive λ there exists a sequence of controls $u(0), \dots, u(N-1)$ such that:

$$\lambda x^* = A^{N-1} b f(u(0)) + \dots + A b f(u(N-2)) + b f(u(N-1)) \tag{7.4}$$

Assume that x^* has been chosen (it is one of the design parameters). Since x^* is in the interior of the cone C, λx^* is also in the interior of the cone for all positive λ.

To complete the design of Controller 2 we need Corollary 6.4.1 in [153]:

Proposition 7.1 *Let C be a convex set in $\mathbb{R}^n$. Then $z \in \overset{\circ}{C}$ if and only if for every $y \in \mathbb{R}^n$, there exists some $\epsilon > 0$ such that $z + \epsilon y \in \overset{\circ}{C}$.* □

Using this result we can say that for any $x(0)$ in the complement of the cone C, there exists a positive λ such that

$$A^N x(0) + \lambda x^* \in \overset{\circ}{C}$$

In other words, there exists a sequence of controls satisfying (7.4) which yields $x(N) \in \overset{\circ}{C}$.

Given any $x(0)$ and $x^* \in \overset{\circ}{C}$, let us find the value of $\hat{\lambda}$ which yields $x(N) \in \overset{\circ}{C}$. Consider the set of inequalities

$$\begin{aligned} Kx(N) &> 0 \\ KJx(N) &> 0 \\ \ldots & \quad \ldots \\ KJ^{n-1}x(N) &> 0 \end{aligned} \tag{7.5}$$

If all of the inequalities are simultaneously satisfied the state $x(N)$ belongs to the interior of the cone C. Therefore, the inequalities:

$$\begin{aligned} KA^N x(0) + \lambda K x^* &> 0 \\ KJA^N x(0) + \lambda KJx^* &> 0 \\ \ldots & \quad \ldots \\ KJ^{n-1}A^N x(0) + \lambda KJ^{n-1}x^* &> 0 \end{aligned} \tag{7.6}$$

must be satisfied. Any $\hat{\lambda}$ satisfying

$$\hat{\lambda} > \max_i \left(-\frac{KJ^iA^N}{KJ^ix^*} \right)$$

guarantees that $x(N)$ belongs to the interior of the cone C. Hence, we can compute $\hat{\lambda}$ using

$$\hat{\lambda}=\max_i \left(-\frac{KJ^iA^N}{KJ^ix^*} \right) + \zeta, \; \zeta > 0 \tag{7.7}$$

Using (7.4) we can compute controls $u(i)$, $i=0,\ldots,N-1$ which transfer $x(0)$ to the interior of the cone C:

$$\hat{\lambda}x^*=A^{N-1}bf(u(0)) + \ldots + Abf(u(N-2)) + bf(u(N-1)) \tag{7.8}$$

Now we can formally state a result:

Theorem 7.6 *Consider Hammerstein systems for which (A, b) is a controllable pair, A non singular and Assumption 7.2 holds. The following controller yields dead-beat behaviour:*

if $x(k) \in C$ apply any real root u to $f(u)=K(x(k))$

otherwise apply a control sequence $u(0), \ldots, u(N-1)$ *which satisfies:*

$$\hat{\lambda}x^* = A^{N-1}bf(u(0)) + \ldots + Abf(u(N-2)) + bf(u(N-1))$$

where C *is defined by (7.3),* $x^* \in \mathring{C}$ *and* $\hat{\lambda}$ *computed using (7.7).* □

We present below two special situations in which there exists an integer L such that $A^L b \in \mathring{C}$. In the first case $L > 0$ and in the second $L=0$. It is interesting that if $L=0$, then a minimum-time dead-beat controller is obtained using this approach. Moreover, the obtained controller is closed loop (on the whole state space). This situation corresponds to the case when the characteristic polynomial of matrix A has all coefficients strictly positive.

Corollary 7.1 Consider a simple Hammerstein systems for which (A, b) is a controllable pair, A is full rank and Assumption 7.2 holds. If there exists an integer L such that $KJ^iA^Lb > 0, \forall i=0,1,\ldots,n-1$ then the control law:

$$u = \text{any real solution to} \begin{cases} f(u) = Kx, & \text{if} x \in C \\ f(u) = 0, & \text{if} x \in S \\ f(u) = \max_{i=0,1,\ldots,n-1} \frac{-KJ^iA^{L+1}x}{KJ^iA^Lb} + \zeta, \zeta \geq 0, & \text{otherwise} \end{cases}$$

where $S = \cup_i S_i, S_i = \{x : KA^ix \geq 0, KJA^ix \geq 0, \ldots, KJ^{n-1}A^ix \geq 0\}, i=0,1,\ldots,L$, is dead-beat and it transfers every initial state to the origin in at most $n+L+1$ time steps. □

It is important to emphasize that ζ may be a constant $\zeta \geq 0$ or a function $\zeta = \zeta(x(k), k) \geq 0, \forall x(k), \forall k$.

Proof of Corollary 7.1: Consider the following equations:

$$\begin{aligned} KA^{L+1}x + K(A^Lbf(u(0)) + A^{L-1}bf(u(1)) + \ldots + bf(u(L-1))) &\geq 0 \\ \ldots \quad \ldots &\ldots \\ KJ^{n-1}A^{L+1}x + KJ^{n-1}(A^Lbf(u(0)) + A^{L-1}bf(u(1)) + \ldots + bf(u(L-1))) &\geq 0 \end{aligned}$$

Since $KJ^iA^Lb > 0, \forall i=0,1,\ldots,n-1$, it follows that $\forall x(0) \notin C$ the control law

$$f(u(0)) = \max_{i=0,1,\ldots,n-1} \frac{-KJ^iA^{L+1}x}{KJ^iA^Lb} + \zeta, \zeta \geq 0$$

and $u(i) = 0, \forall i = 1, 2, \ldots, L-1$ transfers $x(0)$ to C in $L+1$ steps. Q.E.D.

We denote the characteristic polynomial of the matrix A as $p_c(A) = p^n + \sum_{i=0}^{i=n-1} a_i p^i, \ \ a_i \in \mathbb{R}, \forall i = 0, \ldots, n-1$.

Corollary 7.2 If the matrix A has a characteristic polynomial with all coefficients strictly positive, that is $a_i > 0, \ \forall i = 0, 1, \ldots, n-1$ then the controller:

$$u = \text{ any real solution to } \begin{cases} f(u) = Kx, & \text{if } x \in C \\ f(u) = \max_{i=0,1,\ldots,n-1} \frac{-KJ^i Ax}{a_i} + \zeta, \zeta \geq 0, & \text{if } x \notin C \end{cases}$$

is dead-beat and it transfers every initial state to the origin in at most $n+1$ time steps. □

Proof of Corollary 7.2: Notice that we have $KJ^i b = a_i, \forall i = 0, 1, \ldots, n-1$. Since $a_i > 0, \forall i = 0, 1, \ldots, n-1$, the proof follows easily. Q.E.D.

Comment 7.2 Observe that we have several design parameters which can be used to change the transient response when using Controller 2 given in Theorem 7.6. First, the state x^* can be arbitrarily chosen so that $x^* \in \overset{\circ}{C}$. Second, when computing the value $\hat{\lambda}$ using (7.7) we can choose a value for $\zeta > 0$. Furthermore, when solving the equation (7.8) we usually have more than a particular solution. Therefore, we need to specify a rule according to which we choose one solution. Finally, notice that if we multiply a polynomial with positive coefficients with another polynomial with positive coefficients, we obtain again a polynomial with positive coefficients. This implies that we may work with longer sequence of controls than N when the controller is in open loop mode. In other words, the dead-beat time can also be changed using this control method. □

7.4.4 Controller 3: General Case

The consideration of Controller 2 is motivated mainly by the simplicity of its design. We show now how it is possible to design a minimum-time dead-beat controller for general simple Hammerstein systems. We assume that $\mathrm{im} f(u) = \mathbb{R}_0^+$ but the generalisation to the situation when $\mathrm{im} f(u) = \mathbb{R}_\delta^+$ is straightforward. We again emphasize that we assume that the matrices A and b are in controllability canonical form and that the cone C can be easily shown to be equal to S_{n-1}. Hence, sets $S_k, k \geq n-1$ can be computed as set of points that can be mapped to the cone in one, two, etc. time steps. To emphasize the structure of these sets in the context of simple Hammerstein systems (cones) we use the notation $C_k, k \geq n-1$ for these sets.

We compute now the sets of states (cones C_i) that can be transferred to the origin in $1, 2, \ldots, L$ time steps. Since the cone C is actually equal to S_{n-1}, the set S_n=C_n can be computed as a set of states that can be transferred to the cone C in one step. Find the compositions

$$\begin{aligned} KAx + Kbf(u) &\geq 0 \\ KJAx + KJbf(u) &\geq 0 \\ &\ldots \\ KJ^{n-1}Ax + KJ^{n-1}bf(u) &\geq 0 \end{aligned} \tag{7.9}$$

We split the set of equations (7.9) into three groups according to the sign of KJ^iAb. The set of i for which KJ^iAb=0 is relabelled as $s_1, \ldots, s_{m_0}$. The same is done for the sets of indices i for which $KJ^iAb > 0$ and $KJ^iAb < 0$. They are denoted respectively as $t_1, \ldots, t_{m_+}$ and $p_1, \ldots, p_{m_-}$. It is obvious that the set $KJ^{s_i}Ax \geq 0, i$=$1, \ldots, m_0$ is a part of the set of equations that define C_n.

Moreover, we have that there exists a control u which transfers a state x from C_n to C if and only if the following inequalities are satisfied:

$$\begin{aligned} \min_{p_i} \frac{KJ^{p_i}Ax}{KJ^{p_i}b} \quad \geq f(u) \geq & \quad \max_{t_j} \frac{KJ^{t_j}Ax}{KJ^{t_j}b},\ \forall p_i, t_j, i{=}1, \ldots, m_-, j{=}1, \ldots, m_+ \\ f(u) \qquad \geq & \qquad 0 \end{aligned} \tag{7.10}$$

Using these inequalities we see that the following inequalities must be satisfied:

$$\begin{aligned} \left(\frac{KJ^{p_i}A}{KJ^{p_i}b} - \frac{KJ^{t_j}A}{KJ^{t_j}b}\right)x &\geq 0\ \forall p_i, t_j, i{=}1, \ldots, m_-, j{=}1, \ldots, m_+ \\ \frac{KJ^{p_i}A}{KJ^{p_i}b}x &\geq 0,\ \forall p_i, i{=}1, \ldots, m_- \end{aligned} \tag{7.11}$$

Now it is not difficult to see that the defining set for C_n is

$$\begin{aligned} KJ^{s_i}Ax &\geq 0,\ i{=}1, \ldots, m_0 \\ \left(\frac{KJ^{p_i}A}{KJ^{p_i}b} - \frac{KJ^{t_j}A}{KJ^{t_j}b}\right)x &\geq 0,\ \forall p_i, t_j, i{=}1, \ldots, m_-, j{=}1, \ldots, m_+ \\ \frac{KJ^{p_i}A}{KJ^{p_i}b}x &\geq 0,\ \forall p_i, i{=}1, \ldots, m_- \end{aligned} \tag{7.12}$$

If we denote the set of inequalities (7.12) as $l_i^n x \geq 0, i$=$1, 2, \ldots, m_n$, we can write that C_n=$\{x:$

$l_i^n x \geq 0, i{=}1,2,\ldots,m_n\}$. The set C_{n+1} is computed in a similar way where we start the same procedure from the following set of inequalities:

$$l_i^n Ax + l_i^n bf(u) \geq 0, i{=}1,2,\ldots,m_n$$

It is important to notice that there exists a uniform bound on the minimum number of steps necessary to transfer any initial state to the origin. This can be seen from the proof of Theorem 7.1. Consequently, there exists an integer L which is such that $\cup_{i=1}^{i=L+n+1} C_i{=}\mathbb{R}^n$. It only remains to compute the controls that transfer any state in C_{i+1} to $C_i, \forall i{=}1,2,\ldots,L+n+1$.

It is obvious that the control law $f(u){=}Kx$ maps C_{i+1} to $C_i, \forall i{=}0,1,\ldots,n-1$. We use the notation $C_i{=}\{x : l_j^i x \geq 0, j{=}1,2,\ldots,m_i\}, i{=}n,n+1,\ldots,L+n+1$. We also use the indices s_m^i, p_s^i and t_j^i to denote the indices f for which $l_f^i b$ is respectively equal, less than and greater than zero. Then the controls u that satisfy:

$$f(u){=}v_i(x)\,,\ \text{if}\ x \in C_{i+1},\ i{=}n,\ldots,L-1$$

and

$$f(u) \geq 0$$

where $v_i(x)$ can take values from the following interval

$$v_i(x) \in [\,\max(0, \min_{p_s^i} \frac{KJ^{p_s^i}Ax}{KJ^{p_s^i}b})\,, \min(0, \max_{t_j^i} \frac{KJ^{t_j^i}Ax}{KJ^{t_j^i}b})\,]\,,\ x \in C_{i+1}$$

transfer any state in C_{i+1} to C_i in one step. Hence, we designed a family of controllers and by specifying the law according to which we chose $v_i(x)$ we can shape the response of the system. One such rule for the choice of $v_i(x)$ might be: choose $v_i(x)$ such that $|u|$ has minimum value at each step. This control law is minimum-time and can be applied to any simple Hammerstein model. It is clear that the controller is much more complex than that presented in Theorem 7.1 and Corollary 7.2.

Comment 7.3 In this chapter we assumed that we have information about the full state vector for control purposes. This is not a restrictive assumption at all since we can design an observer for simple Hammerstein systems if the matrix pair (c, A) is observable. The observer then has the following form:

$$z(k+1)=Az(k)+bf(u(k))+g(cz(k)-y(k)+df(u(k))) \tag{7.13}$$

The error is then governed by

$$e(k+1)=z(k+1)-x(k+1)=(A+gc)e(k) \tag{7.14}$$

whose dynamics can be assigned arbitrarily. Notice that we have complete modularity between the controller-observer pair and if we design a dead-beat observer then the overall system will also have dead-beat behaviour. □

7.5 An Output Dead-Beat Controller

If instead of zeroing the state of the system (7.1) we wish to zero its output in finite time, we need an output dead-beat controller. Necessary and sufficient conditions for output dead-beat controllability of simple Hammerstein systems are not known. It is obvious though (see equation (7.1)) that output dead-beat controllability is an easy consequence of state dead-beat controllability. We discuss some conditions under which output dead-beat control can be achieved. These conditions allow for systems that are not state dead-beat controllable. An explicit construction of an output dead-beat controller is presented.

We still make use of Assumption 7.2 for the same reason as before.

Theorem 7.7 *Consider system (7.1) under Assumption 7.2. Let* $H=A-bcd^{-1}$*, assuming that* $d\neq 0$*. Define*

$$C_O=\{x : d^{-1}cH^i x \leq 0, i=0,1,\ldots,L-1\}$$

Suppose the following conditions are satisfied:

1. *The matrix* H *satisfies a polynomial equation*

$$H^L - \sum_{i=0}^{i=L-1} c_i H^i=0, \text{ where } c_i \geq 0, \forall i=0,\ldots,L-1$$

2. *There exists a number N such that:*

$$d^{-1}cH^iA^Nb < 0,\ \forall i{=}0,1,\ldots,L-1$$

Then system (7.1) is output dead-beat controllable. If H is a stable matrix (with all eigenvalues inside the closed unit disk), the system is output dead-beat controllable with stable zero dynamics. □

Comment 7.4 Notice that under the conditions of Theorem 7.7 the system (7.1) does not have to be state dead-beat controllable. □

Comment 7.5 Observe that $0 \in C_O$ is always satisfied and that if $\{0\}{=}C_O$ (the cone C_O is trivial), the system (7.1) must necessarily be state dead-beat controllable in order to have output dead-beat controllability. □

Proof of Theorem 7.7: Because of Condition 1 in Theorem 7.7, it is not difficult to see that the cone C_O is positively invariant. In other words, if an initial state is in the cone, it stays inside the cone when the control $f(u){=}{-}d^{-1}cx$ is applied to the system.

Consider the following inequalities:

$$d^{-1}(cA^{N+1}x(0) + cA^Nbf(u(0)) + cA^{N-1}bf(u(1)) + \ldots + cbf(u(N-1))) \le 0$$

$$d^{-1}(cHA^{N+1}x(0) + cHA^Nbf(u(0)) + cHA^{N-1}bf(u(1)) + \ldots + cHbf(u(N-1))) \le 0$$

$$\ldots$$

$$d^{-1}(cH^{L-1}A^{N+1}x(0) + cH^{L-1}A^Nbf(u(0)) + cH^{L-1}A^{N-1}bf(u(1)) + \ldots + cH^{L-1}bf(u(N-1))) \le 0$$

If Condition 2 of Theorem 7.7 is satisfied, we can transfer any state outside the cone C_O to the cone C_O by applying as control law

$$f(u(0)) = \max\left(\max_i \frac{-cH^iA^{N+1}x(0)}{cH^iA^Nb}, 0\right)$$

and $f(u(k)){=}0, \forall k{=}1,2,\ldots,N-1$. We have that $x(N) \in C_O$ and then we can apply $f(u(k)){=}{-}d^{-1}cx(k)$. Q.E.D.

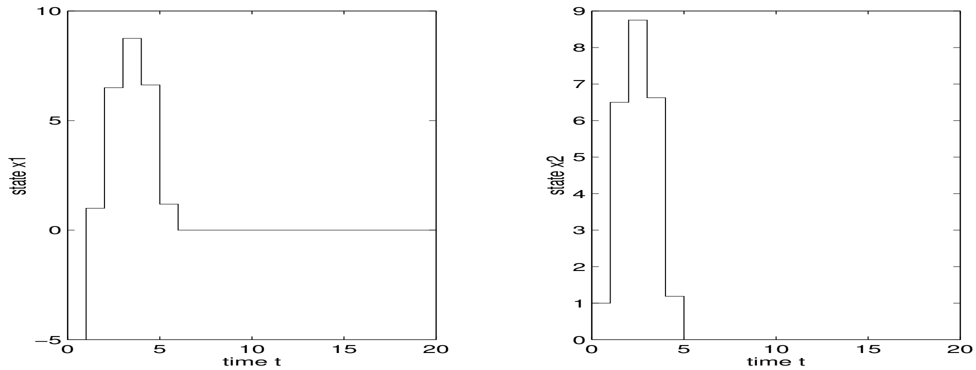

Figure 7.5: Controller 1 yields dead-beat behaviour for second order systems

From the proof of Theorem 7.7 it follows that the output dead-beat control law is:

$$u(k) = \text{any real root to} \begin{cases} f(u) = -d^{-1}cx, & \text{if } x \in C_O \\ f(u) = 0, & \text{if } x \in S \\ f(u) = \max(\, \max_i \frac{-cH^iA^{N+1}x}{cH^iA^Nb}, 0) + \zeta, & \text{if } x \in \mathbb{R}^n - (C_O \cup S)\,, \zeta \geq 0 \end{cases}$$

where

$$S = \cup_{i=1}^{i=N-1} \{x : d^{-1}cA^ix \leq 0, \ldots, d^{-1}cH^{L-1}A^ix \leq 0\}.$$

7.6 Examples

Example 7.1 Consider the system:

$$x(k+1) = \begin{pmatrix} 0 & 1 \\ -1 & 1.5 \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u^2(k)$$

The dead-beat control law for the unconstrained linear system is $u(k) = (1 \;\; -1.5)\, x(k)$. Controller 1 takes on the following form:

$$u^2(k) = (1 \;\; -1.5)\, x(k)\,, \text{ if } (1 \;\; -1.5)\, x(k) \geq 0$$

$$u(k) = 0, \text{ if } (1 \;\; -1.5)\, x(k) < 0$$

A simulation result for $x(0) = (-5 \;\; 1)^T$ is shown in Fig. 7.5.

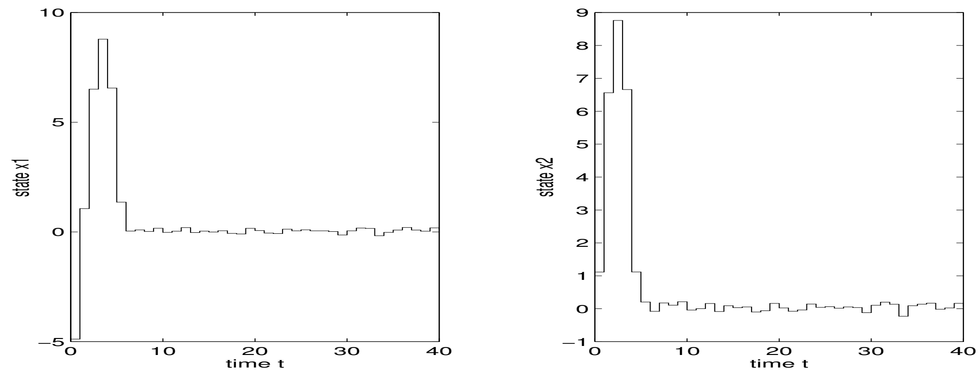

Figure 7.6: Controller 1 with measurement noise

In Fig. 7.6 the same system is simulated when the measurements are assumed to be affected by low level zero mean measurement noise. The system is simulated using Simulink (Matlab) and the source for noise is a random number generator (default with zero initial seed) which is fed into a linear gain of 0.1. The output of the linear gain is the generated noise which is added to the state measurements. Apparently, the performance has not deteriorated in the presence of noise. The standard deviations of state and control signals after the transient (for the period 7 to 40 time steps) with the noise are respectively $std(x_1)$ =0.095, $std(x_2)$ =0.1054 and $std(u)$ =0.0464. □

Example 7.2 Consider the system:

$$x(k+1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0.5 & 0.5 \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u^2(k) \tag{7.15}$$

The control law of Theorem 7.5 is implemented to this system and simulations show that there exist some states in the state space from which the closed loop system converges asymptotically to the origin. The results of simulations for the initial state $x(0) = (1\ \ -2\ 1)^T$ are shown in Figure 7.7.

We would like to show that the controller asymptotically stabilises the plant. Consider the following conditions:

1. $Kx \geq 0$

2. $KJx \geq 0$

3. $KJ^2x \geq 0$

We denote sets for which some of the above conditions hold as, e.g. S_{12}, which means that the first two conditions are satisfied whereas the third one is not. Obviously, the cone $C{=}S_{123}$. S denotes the set for which none of the conditions is satisfied. Also, the following relations are obvious:

$$S_1 \cup S_{12} \cup S_{123} \cup S_{13}{=}H_1$$

$$S_2 \cup S_{23} \cup S_3 \cup S{=}H_2$$

It is not difficult to see that all states from H_2 are mapped to H_1 in a finite number of steps. Moreover, since $x_3(k+1){=}{-}Kx(k)$ it follows that all states in H_2 are mapped to S_{123} of S_{13}. If a state is mapped to S_{123} in finite steps, we have dead-beat behaviour. If this is not the case we have that they are mapped to S_{13}. Simulations show that there exist some states for which S_{13} is mapped to H_1 and then H_1 to S_{13} etc. From simulations we can not say how many steps these states stay in H_1 before they are mapped back to H_2.

Suppose that there are states such that $x(0) \in S_{13}$ and $x(1), x(2), x(3) \in H_2$. This set is

$$\begin{aligned}&\{x : KJ^2x \geq 0, KAJx < 0, KA^2Jx < 0, KA^3x < 0\}\\&{=}\{x : x_3 \geq 0, 0.5x_2 + 0.75x_3 < 0, 0.75x_2 + 0.125x_3 < 0, -0.375x_2 + 0.9375x_3 < 0\}{=}\emptyset.\end{aligned}$$

Moreover, suppose that $x(0) \in S_{13}$ and $x(1), x(2) \in H_2$ and $x(3) \notin S_{123}$. This set is given by:

$$\begin{aligned}&\{x : KJ^2x \geq 0,\ KAJx < 0,\ KA^2Jx < 0,\ KJA^2Jx < 0\}\\&{=}\{x : x_3 \geq 0,\ 0.5x_2 + 0.75x_3 < 0,\ 0.75x_2 + 0.125x_3 < 0,\ -0.75x_2 + 0.875x_3 < 0\}{=}\emptyset.\end{aligned}$$

Therefore, all initial states in S_{13} that are mapped to H_2 in two consecutive steps are mapped to S_{123} in the third step. As a result, the only behaviour which is not dead-beat is defined by intermittent mapping between S_{13} and H_2 and the system evolves according to the equation:

$$x(k+2){=}AJx(k)$$

Since the matrix AJ has got $\{0, 0, 0.5\}$ eigenvalues, the system exhibits asymptotic behaviour on

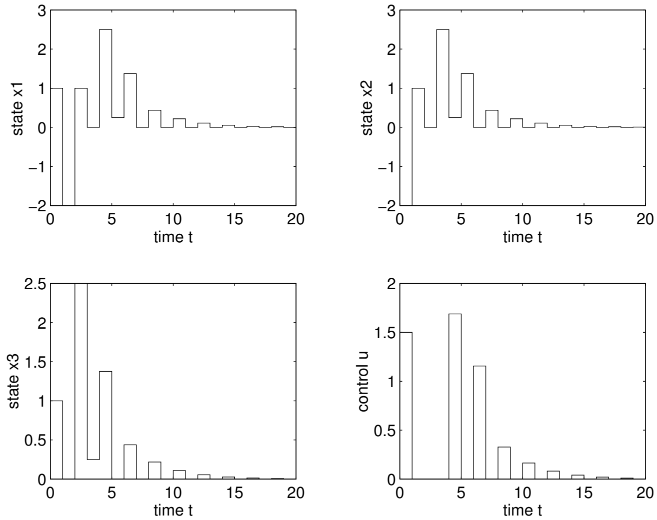

Figure 7.7: Controller 1 may yield asymptotic behaviour for a third order system

the cone defined as:

$$\{x : kx \geq 0,\ KJx < 0,\ KJ^2x \geq 0,\ KAJx \geq 0,\ KJAJx < 0, KJ^2AJ \geq 0\}$$

It is obvious now that the system is asymptotically stable. Controller 1, exhibits dead-beat behaviour for a subset of state space which consists of the cone C and sets that are mapped to the cone with the specified control law. This behaviour is very interesting and is a property of nonlinear systems which does not have a linear counterpart.

We show that Controllers 2 and 3 yield dead-beat behaviour for the system (7.15).

Controller 2:

The conditions of Theorem 7.1 are satisfied since

$$KA^2b{=}0.375,\quad KJA^2b{=}0.125,\quad KJ^2A^2b{=}0.75$$

Hence, $L{=}2$ and $KA^LJ^ib > 0, \forall i{=}0, 1, 2$. We now design the dead-beat controller for the system (7.15). First a minimum-time dead-beat controller is designed for the unconstrained linear system. The controller is given by $f(u(k)) = Kx(k) = (1\ -0.5\ -0.5)\,x(k)$. This controller is implemented on the cone $C_2{=}\{x : Kx \geq 0, KJx \geq 0, KJ^2x \geq 0\}$. The sets $C_i, i{=}3, 4, 5$ are computed. Sets C_3 and C_4 represent sets of states outside the cone $C{=}C_2$ that are transferred to the cone when $u{=}0$ is applied. The cone C_5 is such that any state in it can be transferred to $C_2 \cup C_3 \cup C_4$ in one step. The controller is given by:

$$\begin{aligned}
u^2 &= Kx = (1 \ -0.5 \ -0.5)\,x, \ \text{ if } x \in C \\
u^2 &= 0, \ \text{ if } x \in C_3 \cup C_4 \\
u^2 &= \max(k_1, k_2, k_3, 0) + 1, \ \text{ if } x \in C_5
\end{aligned}$$

where

$$\begin{aligned}
C_2 &= \{x : Kx \geq 0, KJx \geq 0, KJ^2x \geq 0\} \\
&= \{x : (1 \ -0.5 \ -0.5)\,x \geq 0, (0\ 1 \ -0.5)\,x \geq 0, (0\ 0\ 1)\,x \geq 0\} \\
C_3 &= \{x : KAx \geq 0, KJAx \geq 0, KJ^2Ax \geq 0\} \\
&= \{x : (0.5\ 0.75 \ -0.75)\,x \geq 0, (0.5 \ -0.25\ 0.75)\,x \geq 0, (-1\ 0.5\ 0.5)\,x \geq 0\} \\
C_4 &= \{x : KA^2x \geq 0, KJA^2x \geq 0, KJ^2A^2x \geq 0\} \\
&= \{x : (0.75\ 0.125\ 0.375)\,x \geq 0, (-0.75\ 0.875\ 0.125)\,x \geq 0, (-0.5 \ -0.75\ 0.75)\,x \geq 0\} \\
C_5 &= \mathbb{R}^3 - (C_2 \cup C_3 \cup C_4) \\
k_1 &= (1 \ -2.5 \ -0.8333)\,x; \ k_2 = (1\ 5.5 \ -7.5)\,x; \ k_3 = (1\ 0.1667\ 0.5)\,x
\end{aligned}$$

Controller 3:

We now design the general dead-beat controller for the system. The cone $C = C_2$ is the same as the one above and the same control law is applied on the cone. The sets $C_i, i = 3, 4, 5$ are computed. They are sets of states that can be transferred to the cone C in $1, 2, 3$ steps and therefore to the origin in $4, 5, 6$ steps. The design yields the general controller:

$$\begin{aligned}
u^2 &= Kx = (1 \ -0.5 \ -0.5)\,x, \ \text{ if } x \in C_2 \\
u^2 &= (\min(a,b) + c)\,/2, \ \text{ if } (\min(a,b) + c)\,/2 \geq 0, \ x \in C_3 \\
u^2 &= 0, \ \text{ if } (\min(a,b) + c)\,/2 < 0, \ x \in C_4 \\
u^2 &= (\min(d,e) + \max(f,g))\,/2, \ \text{ if } (\min(d,e) + \max(f,g))\,/2 \geq 0, \ x \in C_5 \\
u^2 &= 0, \ \text{ if } (\min(d,e) + \max(f,g))\,/2 < 0, \ x \in S_5 \\
u^2 &= \max_i(c_i) + 1, \ \text{ if } \max_i(c_i) \geq -1, \ x \in C_5 \\
u^2 &= 0, \ \text{ if } \max_i(c_i) < -1, \ x \in C_5
\end{aligned}$$

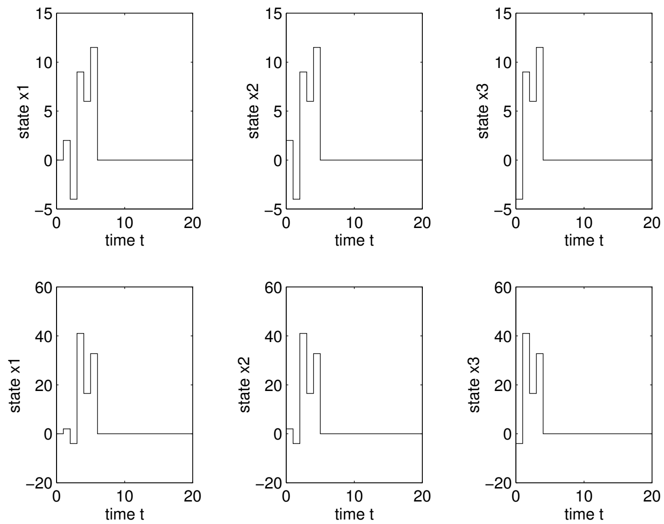

Figure 7.8: Controller 3 (top) and Controller 2 (bottom) for initial state $(0, 2, -4)$

where

$$
\begin{aligned}
a &= (1\ 1.5\ -1.5)\,x;\ b{=}(1\ -0.5\ 1.5)\,x;\ c{=}(1\ -0.5\ -0.5)\,x \\
d &= (1\ 0.1667\ 0.5)\,x;\ e{=}(1\ -0.5\ 1.5)\,x;\ f{=}(1\ -1.1667\ -0.1667)\,x \\
g &= (1\ -0.5\ -0.5)\,x;\ c_1{=}(1\ -2.5\ -0.8334)\,x;\ c_2{=}(1\ -1.1667\ -0.1667)\,x \\
c_3 &= (1\ -0.5\ -2.5)\,x;\ c_4{=}(1\ -0.5\ -1.1667)\,x;\ c_5{=}(1\ -0.5\ -0.9)\,x \\
c_6 &= (1\ -0.5\ -0.5)\,x \\
C_3 &= \{x : (1\ 1.5\ -1.5)\,x \geq 0, (1\ -0.5\ 1.5)\,x \geq 0, (0\ 0\ 1)\,x \geq 0, (0\ 2\ -1)\,x \geq 0\} \\
C_4 &= \{x : (1\ 0.1667\ 0.5)\,x \geq 0, (1\ -0.5\ 1.5)\,x \geq 0, (0\ 1.334\ 0.6667)\,x \geq 0, \\
&\quad (0\ 0.6667\ 1)\,x \geq 0, (0\ 0.6667\ 1.6667)\,x \geq 0, (0\ 0\ 1)\,x \geq 0\} \\
C_5 &= \mathbb{R}^3 - (C_2 \cup C_3 \cup C_4)
\end{aligned}
$$

From simulations it was observed that Controller 3 yields better transient performance. For instance, see Fig. 7.8. We emphasize that it is possible to shape the transient response while maintaining minimum-time dead-beat behaviour. □

The following example illustrates the deasign method for an output dead-beat controller. We note that the system is not state dead-beat cotrollable but the output dead-beat controller can still be designed.

Example 7.3 Consider the system:

$$x(k+1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0.2 \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u^2(k)$$

$$y(k) = \begin{pmatrix} -0.1 & 0 & -0.5 \end{pmatrix} x(k) + u^2(k)$$

It is straight forward to check that the conditions of Theorem 7.7 are satisfied, with L=3 and N=0. The design is very similar to the design of state dead-beat controllers given above. The output dead-beat controller is:

$$u^2(k) = \begin{pmatrix} 0.1 & 0 & 0.5 \end{pmatrix} x(k)\,, \text{ if } x \in C_O$$

$$u^2(k) = \max_i (a_i) + 1, \text{ if } x \in \mathbb{R}^3 - C_O$$

where

$$\begin{aligned} C_O \;\; = \;\; \{x : & \begin{pmatrix} 0.1 & 0 & 0.5 \end{pmatrix} x(k) \geq 0, \begin{pmatrix} 0.05 & 0.1 & 0.35 \end{pmatrix} x(k) \geq 0, \\ & \begin{pmatrix} 0.035 & 0.05 & 0.345 \end{pmatrix} x(k) \geq 0\} \end{aligned} \tag{7.16}$$

and

$$\begin{aligned} a_1 &= \begin{pmatrix} 0 & -0.2 & -0.2 \end{pmatrix} x(k)\,;\; a_2 = \begin{pmatrix} 0 & -0.1429 & -0.4857 \end{pmatrix} x(k) \\ a_3 &= \begin{pmatrix} 0 & -0.1014 & -0.3449 \end{pmatrix} x(k)\,;\; a_4 = 0 \end{aligned} \tag{7.17}$$

Simulation of the output dead-beat controller is given in Fig. 7.9. Notice that the system is not state dead-beat controllable. □

7.7 *Conclusion*

We have presented a number of state and output dead-beat controllers for the class of simple Hammerstein systems. The method that we propose consists of two steps. In the first step we find sets $C_0, C_1, \ldots$ from which the state/output can be zeroed in one, two, etc. time steps. The

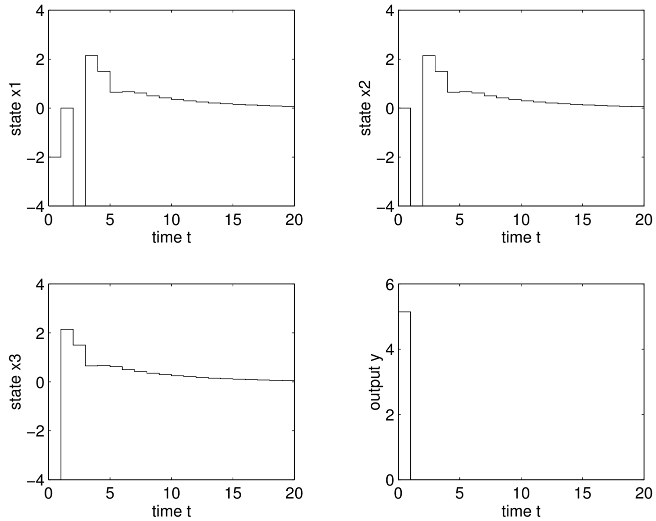

Figure 7.9: Output dead-beat controller for a system that is not state dead-beat controllable

second step is the design of a control law which maps C_{i+1} to $C_i, \forall i$. We showed how it is possible to obtain a family of time-optimal dead-beat controllers using this approach. We point out that because of the simplicity of the underlying structure of this class of systems the design of dead-beat controllers can be carried out without resorting to QEPCAD. Hence, the computational complexity of a dead-beat controller design is rather small when compared to the general algorithms proposed in Chapter 3.

It has been shown how the transient response can be modified, while preserving time optimality. An interesting open question is to design minimum-time dead-beat controllers which also minimise some quadratic cost. In this way, a more systematic analysis of an optimal choice of the design parameters could be carried out. In some cases, good behaviour of the designed controllers under low level measurement noise conditions is also observed.

Chapter 8

Generalised Hammerstein Systems

8.1 Introduction

Generalised Hammerstein systems may arise from identification techniques of the so called block oriented models [75, 76]. They represent a subclass of the class of input-output polynomial systems, very often referred to as NARMAX (nonlinear ARMAX) [75, 76]. Generalised Hammerstein systems can be regarded as a parallel connection of a simple Hammerstein system whose input nonlinearity is quadratic and a linear system, see Figure 8.1. The output connection may be more complicated than the one presented in the figure but this is not crucial for our developments.

Although the structure of this class of models is very simple, it turns out to be adequate to model the dynamics of some practically important plants. For example, in [104] the model of a cement mill is identified as a generalised Hammerstein system. In the same paper, the author claims that his method can be used to identify models for some other milling processes. Another application of generalised Hammerstein models (MI) can be found in [11] where the model for the cooling water circulation of a thermal power plant was identified in this form.

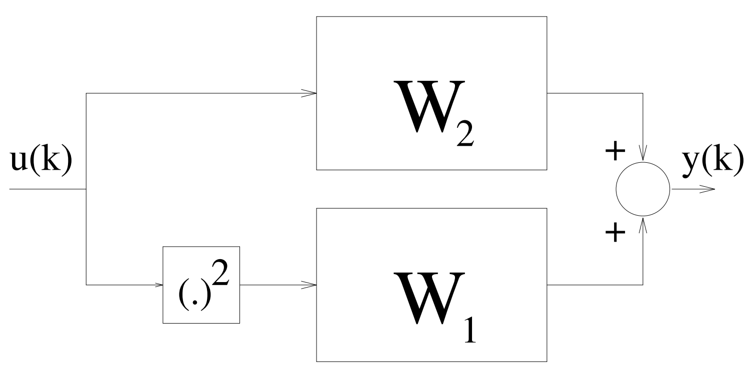

Figure 8.1: Block diagram of a generalised Hammerstein system

Controllability of this class of systems is a very important property and a checkable test similar to the well known rank condition for linear systems is an important tool in the control engineer's toolbox. However, the general class of NARMAX models is very large and it seems that it is necessary to constrain the structure of the general class of systems severely in order to obtain easy-to-check tests. Generalised Hammerstein systems offer such an opportunity since only linear algebra is needed to deal with the problem.

Dead-beat control and controllability of some classes of simple Hammerstein systems, which are characterised by a series of a static nonlinearity and a linear dynamic block, were investigated in Chapter 7. Simple Hammerstein systems have a structure which is very close to linear, which allows us to efficiently design non-minimum and minimum-time dead-beat controllers.

Here, we present necessary and sufficient conditions for dead-beat and complete controllability of generalised Hammerstein systems. A result on controllability of linear systems with positive controls is used in the proof [50]. The ensuing controllability test is very easy to use.

It is a well known fact that a parallel connection of two linear controllable systems may fail to be controllable [102, pg.156]. The main result of this chapter, however, states that the parallel connection of the linear and simple Hammerstein system is *always dead-beat controllable* if its subsystems are controllable. This result is somewhat unexpected.

Results of this chapter are important since they may be used to prove more general results on controllability of interconnected nonlinear systems (see Chapter 9). We also conjecture that the connection of a linear and a simple Hammerstein system with arbitrary input polynomial is *always dead-beat controllable* if the subsystems are dead-beat controllable.

8.2 Main Result

We consider generalised Hammerstein systems of the form [75, 76]:

$$
\begin{aligned}
\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \end{pmatrix} &= \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \begin{pmatrix} g_1 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ g_2 \end{pmatrix} u^2(k) \\
y(k) &= (c_1^T \; c_2^T) \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + d_0 + d_1 u(k) + d_2 u^2(k)
\end{aligned}
\tag{8.1}
$$

where $x(k) = (x_1(k) \quad x_2(k))^T \in \mathbb{R}^n$ is a state of the system at time k and $u(k) \in \mathbb{R}$ is the control at time k. We also have $F_1 \in \mathbb{R}^{n_1 \times n_1}$, $F_2 \in \mathbb{R}^{n_2 \times n_2}$, $g_1 \in \mathbb{R}^{n_1 \times 1}$, $g_2 \in \mathbb{R}^{n_2 \times 1}$, $x_1(k) \in \mathbb{R}^{n_1}$ and

$x_2(k) \in \mathbb{R}^{n_2}$.

Notice that the system (8.1) can be regarded as a parallel connection of systems

$$x_1(k+1) = F_1 x_1(k) + g_1 u(k) \tag{8.2}$$

and

$$x_2(k+1) = F_2 x_2(k) + g_2 u^2(k) \tag{8.3}$$

In Chapter 7 we used Theorems 7.1, 7.3 and 7.4 to characterise some properties of linear systems with positive controls. These theorems play a crucial role for dead-beat controllability of generalised Hammerstein systems.

Linear systems with positive controls (7.1) have an interesting property: *if the system (8.3) is completely controllable, there exists a uniform bound on the dead-beat time. That is, there exists a number H^* such that $H(x(0), x^*) \leq H^*$, $\forall x(0), x^* \in \mathbb{R}^n$.* The same holds for dead-beat controllability of (8.3). We will see that the class of generalised Hammerstein systems (8.1) has the same property.

We show below that the system (8.1) is dead-beat (completely) controllable if and only if its subsystems (8.2) and (8.3) are dead-beat controllable.

Theorem 8.1 *The system (8.1) is dead-beat controllable if and only if the following conditions are satisfied:*

1. *rank*$[I\lambda - F_1 : g_1] = n_1, \forall \lambda \neq 0$
2. *rank*$[I\lambda - F_2 : g_2] = n_2, \forall \lambda \neq 0$
3. *all real eigenvalues of the matrix F_2 are negative or zero* □

Comment 8.1 The first condition of Theorem 8.1 means that the subsystem (8.2) is dead-beat controllable. The second and third conditions of Theorem 8.1 represent the necessary and sufficient conditions for controllability of the subsystem (8.3) as established in Theorem 7.4. It is obvious that dead-beat controllability does not require zero modes to be controllable. □

Comment 8.2 Notice that if there are some zero eigenvalues of F_1 or F_2, we can find a non singular transformation T such that

$$\bar{F}_i = T^{-1} F_i T = \begin{pmatrix} D_{11}^i & 0 \\ 0 & D_{22}^i \end{pmatrix}, \ \bar{g}_i = T^{-1} g_i, \ \ i = 1, 2$$

and D_{22}^i is a nilpotent matrix. Assume that the degree of nilpotency of D_{22}^i is d_i. Consider the state at step $k+1 \geq d_i$:

$$x_i(k+1) = \bar{F}_i^k x_i(0) + \sum_{l=0}^{k-1} \bar{F}_i^{k-l-1} \bar{g}_i u(l), \ i = 1, 2$$

If we apply $u(l) = 0, l = k - d_i, k - d_i + 1, \ldots, k$, we have that $x_i(k+1) = (\hat{x}^T \quad 0)^T, \ i = 1, 2$ irrespective of the control sequence $u(l), \ l = 0, 1, \ldots, k - d_i - 1$. Thus, there is no loss of generality if we concentrate just on situations where

$$\text{rank}[F_i - I\lambda : g_i] = n_i, \forall \lambda \in \mathbb{R}, \ i = 1, 2$$

In other words, we assume that

1. $\text{rank}[g_1 : F_1 g_1 : \ldots : F_1^{n_1 - 1} g_1] = n_1$

2. $\text{rank}[g_2 : f_2 g_2 : \ldots : F_2^{n_2 - 1} g_2] = n_2$

3. F_2 has no zero or positive real eigenvalues □

Proof of Theorem 8.1:

Necessity: The necessity part of the proof is obvious since if either of the conditions is violated, either the subsystem (8.2) or (8.3) is not dead-beat controllable and hence (8.1) is also not dead-beat controllable.

Sufficiency: In order to prove sufficiency we will consider special sequences of controls which can transfer any initial state of (8.1) to the origin if the conditions of theorem are satisfied.

Since the last two of the conditions in Comment 8.2 guarantee that the subsystems (8.3) is completely controllable, it is possible to find a sequence of controls $\mathcal{U} = \{u(0), u(1), \ldots, u(P-1)\}$ which yields $x_2(P) = 0$ and $x_1(P) \in \mathbb{R}^{n_1}$. As a result, we assume without loss of generality that $x(0) = (x_1^T(0) \quad 0)^T$.

Since F_2 has no positive or zero eigenvalues (see Comment 8.2), according to Theorem 7.3

the matrix F_2 satisfies a polynomial equation with real positive coefficients:

$$C(F_2) = \sum_{i=0}^{i=N} c_i F_2^i = 0, \quad c_i \geq 0, \ \forall i = 0, 1, \ldots, N. \tag{8.4}$$

Consider now the following sequence of controls:

$$\begin{aligned}
u(0) &= \pm\sqrt{c_N}\, v(0) \\
u(1) &= \pm\sqrt{c_{N-1}}\, v(0) \\
u(2) &= \pm\sqrt{c_{N-2}}\, v(0) \\
&\cdots \\
u(N) &= \pm\sqrt{c_0}\, v(0) \\
u(N+1) &= \pm\sqrt{c_N}\, v(1) \\
&\cdots \\
u((N+1)\, n_1 - 1) &= \pm\sqrt{c_0}\, v(n_1 - 1)
\end{aligned} \tag{8.5}$$

It is obvious that because of (8.4) the state of the subsystem (8.3) $x_2(k)$ is zeroed every $N+1$ steps irrespective of the values $v(k) \in \mathbb{R}, \ k = 0, 1, \ldots, n_1 - 1$. That is, $\forall v(k) \in \mathbb{R}$ we have that $x_2(N+1) = x_2(2(N+1)) = \ldots = x_2(n_1(N+1)) = 0$.

Hence, we now consider if it is possible to zero the state of the subsystem (8.2) $x_1(n_1(N+1))$ by using $v(k), \ k = 0, 1, \ldots, n_1 - 1$ if we start from any initial state $x_1(0) \in \mathbb{R}^{n_1}$. It is important to emphasize that the sign of control $u(k)$ and the values $v(k)$ in (8.5) can be arbitrarily assigned and it is this additional degree of freedom that we are exploiting in the proof.

We have:

$$x_1((N+1)\, n_1) = \sum_{i=0}^{(N+1)n_1 - 1} F_1^{(N+1)n_1 - 1 - i} g_1 u(i) + F_1^{(N+1)n_1} x_1(0) \tag{8.6}$$

The control sequence (8.5) is now substituted in (8.6) and we want to specify the existence of appropriate signs and values $v(k)$, $k = 0, 1, \ldots, n_1 - 1$ such that:

$$\sum_{i=0}^{(N+1)n_1 - 1} F_1^{(N+1)n_1 - 1 - i} g_1 u(i) = -F_1^{(N+1)n_1} x_1(0) \tag{8.7}$$

We introduce the following vector functions:

$$\begin{aligned} L_0 &= \sum_{i=0}^{i=N} F_1^{N-i} g_1 \delta_{0,i} \\ L_1 &= F_1^{N+1} \sum_{i=0}^{i=N} F_1^{N-i} g_1 \delta_{1,i} \\ &\dots \\ L_{n_1-1} &= F_1^{(n_1-1)(N+1)-1} \sum_{i=0}^{i=N} F_1^{N-i} g_1 \delta_{n_1-1,i} \end{aligned} \tag{8.8}$$

where $\delta_{k,i}=\pm\sqrt{c_{N-i}}$, $\forall k=0,1,\dots,n_1-1, i=0,1,\dots,N$. We can rewrite the equation (8.7) as follows:

$$-F_1^{(N+1)n_1} x_1(0) = [L_0 : L_1 : \dots : L_{n_1-1}] \begin{pmatrix} v(0) \\ v(1) \\ \dots \\ v(n_1-1) \end{pmatrix} \tag{8.9}$$

If there exists a sequence of controls of the form (8.5) such that the matrix $[L_0 : L_1 : \dots : L_{n_1-1}]$ is non singular then the system (8.1) is dead-beat controllable.

Because of non singularity of F_2 there exists at least one $\delta_{k,i} > 0$. Non singularity of matrices F_1 and F_2 and controllability of the pair (F_1, g_1) causes the vectors L_k to have entries which are linear functions of $\delta_{k,i}$, $i=0,1,\dots,N$. As a result, the determinant of $[L_0 : L_1 : \dots : L_{n_1-1}]$ is a multi-linear function of $\delta_{k,i}$, which we denote as $p(\delta_{k,i})$.

For any scalar valued affine function $l(y)=ay+b$, $a,b\in\mathbb{R}$, $a\neq 0$ in a scalar variable y, we have that if $l(y)=0$ then $l(-y)\neq 0$. This observation is exploited to select $\delta_{k,i}$ such that $p(\delta_{k,i})\neq 0$.

Let us consider a multi-linear function with three $\delta_{k,i}\neq 0$, which we relabel as $\delta_1, \delta_2, \delta_3$. It can be written in the following form:

$$((K_1\delta_1 + L_1)\delta_2 + (K_2\delta_1 + L_2))\delta_3 + (K_3\delta_1 + L_3)\delta_2 + K_4\delta_1 + L_4 \tag{8.10}$$

If $K_1\neq 0$, we can render $K_1\delta_1 + L_1\neq 0$ by an appropriate choice of δ_1. Moreover, with this choice of δ_1 we can render $(K_1\delta_1 + L_1)\delta_2 + (K_2\delta_1 + L_2)$ non zero by choosing δ_2 and finally the whole expression can be made non zero by a choice of δ_3. If $K_1=0$ but if $L_1\neq 0$ we can do the same, etc. By induction, we show that there is no combination of $\delta_i=\pm\sqrt{c_i}$ which renders (8.10) non zero only if $K_i, L_i=0, i=1,2,3,4$ or F_2 is singular (that is, $\delta_i=0, i=1,2,3$). Since we assumed that $\delta_i \neq 0$, it follows that either F_1 is singular or the pair (F_1, g_1) is not controllable (e.g. $g_1=0$). Contradiction completes the proof. The argument can be carried out for a multilinear function in any number of variables $\delta_{i,k}$ and hence conditions of Theorem 8.1 are sufficient for dead-beat controllability. Q.E.D.

A similar method can be used to prove the following:

Theorem 8.2 *The system (8.1) is completely controllable if and only if the following holds:*

1. *rank*$[g_1 : F_1 g_1 : \ldots : F_1^{n_1-1} g_1] = n_1$
2. *rank*$[g_2 : F_2 g_2 : \ldots : F_2^{n_2-1} g_2] = n_2$
3. *all real roots of F_2 are negative.* □

It is important to notice that although we have used a control sequence of non minimal length in the proof, we did establish that there is a uniform bound on the number of steps necessary to perform dead-beat control.

8.3 Examples

Example 8.1 Consider the system (8.1) for which

$$F_1=\begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix},\ F_2=(-f_2),\ g_1=\begin{pmatrix} 0 \\ 1 \end{pmatrix},\ g_2=(1)$$

Notice that (F_1, g_1) and (F_2, g_2) are controllable pairs, We assume that $f_2 > 0$ and that the matrix F_1 is not singular. Hence, all conditions of Theorem 8.1 are satisfied. $F_2=(-f_2)$ satisfies the equation:

$$F_2 + f_2 I = 0$$

Consider the control sequence:

$$\begin{aligned} u(0) &= v(0) \\ u(1) &= \pm\sqrt{f_2}\, v(0) \\ u(2) &= v(1) \\ u(3) &= \pm\sqrt{f_2}\, v(1) \end{aligned}$$

Notice that we are not considering the most general sequence of controls, since we could in general have that $u(0)=\pm v(0)$, $u(2)=\pm v(1)$. When we apply the control sequence to the system we obtain the following equation:

$$x_1(4)=F_1^4x(0)+(F_1^3g_1\pm\sqrt{f_2}F_1^2g_1)\,v(0)+(F_1g_1\pm\sqrt{f_2}g_1)\,v(1)$$

If there exist a sequence of $\delta_i=\pm1$ for which the matrix:

$$[F_1^3g_1+\delta_1\sqrt{f_1}F_1^2g_1 : F_1g_1+\delta_2\sqrt{f_1}g_1] \tag{8.11}$$

is non singular, the system is completely controllable. Since $p^2+a_1p+a_0$ is the characteristic polynomial of F_1, upon applying the Cayley theorem we obtain that the determinant of the matrix (8.11) is:

$$(a_1^2-a_0-\delta_1ta_1)\,(-a_1+\delta_2t)-2a_0a_1+a_1^3-\delta_1t(a_1^2-a_0)$$

which is equal to zero for all possible choices $\delta_1=\pm1, \delta_2=\pm1$ if the following four equations are satisfied:

$$\begin{aligned}
(a_1^2-a_0-ta_1)\,(-a_1+t)-2a_0a_1+a_1^3-t(a_1^2-a_0) &= 0\\
(a_1^2-a_0+ta_1)\,(-a_1+t)-2a_0a_1+a_1^3+t(a_1^2-a_0) &= 0\\
(a_1^2-a_0+ta_1)\,(-a_1-t)-2a_0a_1+a_1^3+t(a_1^2-a_0) &= 0\\
(a_1^2-a_0-ta_1)\,(-a_1-t)-2a_0a_1+a_1^3-t(a_1^2-a_0) &= 0
\end{aligned} \tag{8.12}$$

where $t=\sqrt{f_2}$. Using the Gröbner basis method [37] (Maple software package) for polynomials (8.12) with the lexicographic ordering $a_1\succ a_0\succ t$, we obtain the Gröbner basis:

$$\{a_1a_0, a_1^2t, a_0t, a_1t^2\}$$

In other words the equations (8.12) are simultaneously satisfied if and only if

$$a_1a_0=0,\ a_1t^2=0,\ a_1^2t=0,\ a_0t=0 \tag{8.13}$$

and therefore at least one of the matrices F_1 or F_2 is singular. This contradicts the assumption that F_i are non singular. It is interesting that in this case we did not use the most general sequence of controls and still we could prove complete controllability. □

Example 8.2 In this example we show how the Gröbner basis method can be used to design a minimum-time dead-beat controller for systems where rankF_2=1. In general one needs to resort to quantifier elimination algorithms, such as QEPCAD. Consider the system:

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \begin{pmatrix} 1 & -2 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{pmatrix} + \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} u^2(k) \qquad (8.14)$$

The conditions of Theorem 8.1 are satisfied since:

$$\text{rank}[g_1 : F_1 g_1] = \text{rank} \begin{pmatrix} 3 & 5 \\ -1 & 4 \end{pmatrix} = 2$$

$$g_2 \neq 0, \text{ and } f_2 = -1 < 0$$

We compute first the set of states that can be transferred to the origin in one step. The entries of $x(0)$ are denoted as $x_1, x_2, \ldots, x_n$. If we compute the Gröbner basis [37] of $\langle x_1 - 2x_2 + 3u(0), x_1 - x_2 - u(0), -x_3 + u^2(0) \rangle$ with the lexicographic ordering $u(0) \succ x_1 \succ x_2 \succ x_3$ [37]. We obtain $\{4u(0) - x_2, 4x_1 - 5x_2, -8x_3 + x_2^2\}$. Using the elimination and extension theorems in [37] we obtain:

$$S_0 = \{x : 4x_1 - 5x_2 = 0 \text{ and } -8x_3 + x_2^2 = 0\}$$

Find $x(2)$ as a function of $x_i(0), u(j), i=1,2,3, j=0,1$ and let $x(2)=0$. The Gröbner basis of $\langle -x_1 + 5u(0) + 3u(1), -x_2 + 4u(0) - u(1), x_3 - 2u^2(0) + 2u^2(1) \rangle$ with the lex order $u(0) \succ u(1) \succ x_1 \succ x_2 \succ x_3$ is $\{17u(0) - x_1 - 3x_2, 5x_2 + 17u(1) - 4x_1, 289x_3 + 30x_1^2 - 92x_1x_2 + 32x_2^2\}$ and hence

$$S_1 = \{x : 289x_3 + 30x_1^2 - 92x_1x_2 + 32x2^2 = 0\}$$

Having found $x(3)$ and letting $x(3)=0$, we compute the Gröbner basis of $\langle -x_1 + 2x_2 - 3u(0) + 5u(1) + 3u(2), -x_1 + x_2 + u(0) + 4u(1) - u(2), -x_3 + 2u^2(0) - 2u^2(1) + 2u^2(2) \rangle$ with the lexicographic ordering $u(0) \succ u(1) \succ u(2) \succ x_1 \succ x_2 \succ x_3$:

$$\{17u(0) - 17u(2) - x_1 - 3x_2, 5x_2 + 17u(1) - 4x_1, -289x_3 + 1156u^2(2)$$

$$-30x_1^2 + 92x_1x_2 + 68x_1u(2) - 32x_2^2 + 204x_2u(2) \}$$

We see that the discriminant of the last polynomial in the basis must be positive, that is we find:

$$S_2 = \{x : 143344x_1^2 - 397664x_1x_2 + 189584x_2^2 + 1336336x_3 \geq 0\}$$

Take now one composition of the discriminant with the (8.14):

$$-96873x_1^2 + 378244x_1x_2 - 544246x_1u - 274204x_2^2 + 302412x_2u + 2092127u^2 - 668168x_3 \tag{8.15}$$

It is obvious that since the coefficient which multiplies u^2 is positive, we can render the equation (8.15) positive for any state in $\mathbb{R}^3$ and hence

$$S_3 = \mathbb{R}^3 - S_2$$

We have constructively proved that the system is dead-beat controllable in 4 steps, by computing the sets $S_k, k=0,1,2,3$. A minimum-time controller follows easily from the proof. $\square$

8.4 Conclusion

Necessary and sufficient conditions for dead-beat and complete controllability of generalised Hammerstein systems are presented. The conditions are very easy to check. The method based on QEPCAD which is described in Chapter 3 can be used to design dead-beat controllers for generalised Hammerstein systems. We think that similar results on dead-beat controllability can be obtained for a parallel connection of a linear system and a simple Hammerstein system with an arbitrary input nonlinearity. We have already proved that dead-beat controllability of the subsystems generically guarantees dead-beat controllability of the overall systems. The main difficulty is to generalise the periodic sequences which used in the proof for dead-beat controllability to deal with more general nonlinearities than the quadratic, which was considered in this chapter.

Chapter 9

Structured Polynomial Systems

9.1 Introduction

In this chapter we consider dead-beat controllability of several classes of interconnected systems. We exploit the way subsystems are interconnected in order to approach the global system's controllability properties. Indeed, interconnections may be such that they allow us to obtain a controllability test for the overall system by testing only some subsystems. This may result in a significant saving in complexity and computational cost for the controllability test. Such circumstances are identified in this chapter. We present three classes of systems but variations on the theme are endless. The results in this chapter should therefore not be viewed as comprehensive but rather be interpreted as examples of how to creatively exploit the topology of the interconnection of subsystems to approach the dead-beat control problem.

This *divide et impera* approach might suffer one drawback. Though we are able to decide controllability more precisely, we might lose the ability to design time-optimal dead-beat controllers. The presented tests do lead to dead-beat control algorithms in the now familiar way, but time optimality might be lost. The first class of triangular systems (Class 1) was already investigated in [160] and a dead-beat property was proved in the same paper. If we, however, combine this result with the result on dead-beat controllability of scalar polynomial systems that is presented in Chapter 5, the dead-beat controllability test is more explicit.

The second class of triangular systems (Class 2) is more interesting since the dead-beat controllability test naturally splits into a number of tests for lower dimensional systems which are readily defined.

Finally, we present a class of polynomial systems which may be viewed as odd systems in the context of state dead-beat controllability (Class 3). Ideas used for odd polynomial systems in Chapters 4, 5 and 6 carry through for this class of systems in order to decide on *state dead-beat controllability*. The generality of this class of systems is striking and unexpected. Indeed, we have found that a discrete-time version of strict feedback systems [109], a class of pure feedback systems [109], a class of NARMAX models [75, 76], a class of homogeneous bilinear systems [70, 48, 71] and a class of inhomogeneous bilinear systems [49] fall into this category. It is interesting that the two classes of bilinear systems are rare cases for which analytic controllability results have been found in the given references, which indicates that the given structure does simplify the controllability problem significantly.

9.2 Class 1

Let us consider state dead-beat controllability of systems:

$$x(k+1) = f(x(k), u(k)) \tag{9.1}$$

with the following triangular structure [160]:

$$\begin{aligned}
x_1(k+1) &= f_1(x_1(k), x_2(k)) \\
x_2(k+1) &= f_2(x_1(k), x_2(k), x_3(k)) \\
& \dots \\
x_{n-1}(k+1) &= f_{n-1}(x_1(k), x_2(k), \dots, x_n(k)) \\
x_n(k+1) &= f_n(x_1(k), x_2(k), \dots, x_n(k), u(k))
\end{aligned} \tag{9.2}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i=1,2,3,\dots,n$ are continuous functions which vanish at zero. The state and control are respectively denoted by $x=(x_1, x_2, \dots, x_n)^T$ and u.

Assumption 9.1 The system (9.2) satisfies the following:

1. For every $i=2,3,\dots,n$ and for each $x_1, x_2, \dots, x_i$, the map $f_i(x_1, \dots, x_i, \cdot)$ is a surjection.
2. For every $i=1,2,\dots,n-1$, $f_i(0,0,\dots,0,x_{i+1})=0 \Leftrightarrow x_{i+1}=0$ and $f_n(0,0,\dots,0,u)=0 \Leftrightarrow u=0$.

The class of systems (9.2) with Assumption 9.1 is very restrictive with respect to the general (polynomial) systems. However, it helps us to carry out a kind of "backstepping procedure" in the proof of dead-beat controllability [109]. Notice that the functions $f_i,\ i{=}2,\ldots,n$ do not have to be polynomial.

Theorem 9.1 *The triangular system (9.2) with Assumption 9.1 is completely dead-beat controllable if and only if the subsystem:*

$$x_1(k+1) = f_1(x_1(k), x_2(k)) \tag{9.3}$$

is completely dead-beat controllable when $x_2 \in \mathbb{R}$ is viewed as the control signal. □

Proof of Theorem 9.1

Necessity: It is clear that the scalar subsystem (9.3) should be dead-beat controllable in order to have dead-beat controllability for the overall system.

Sufficiency: Suppose that the subsystem (9.3) is state dead-beat controllable. This means that $\forall x_1(0) \in \mathbb{R}$, $\exists \{x_2(0), x_2(1), \ldots, x_2(m)\}$ which transfers $x_1(0)$ to the origin in $m+1$ time steps. Notice that for the first $n-1$ time steps the subsystem (9.3) can not be affected by the control $u(0)$. Hence, we can consider the subsystem (9.3) from the step $n-1$:

$$x_1(k+n) = f_1(x_1(k+n-1), x_2(k+n-1)) \tag{9.4}$$

We can also write:

$$\begin{aligned}
x_2(k+n-1) &= f_2(x_1(k+n-2), x_2(k+n-2), x_3(k+n-2)) \\
x_3(k+n-2) &= f_3(x_1(k+n-3), x_2(k+n-3), x_3(k+n-3), x_4(k+n-3)) \\
&\ \ \cdots \\
x_n(k+1) &= f_n(x_1(k), x_2(k), x_3(k), \ldots, x_n(k), u(k))
\end{aligned} \tag{9.5}$$

Notice also that $x_i(k+n-j) = \bar{X}_i(x_1(k), x_2(k), \ldots, x_j(k))$ and that the equations (9.4) and (9.5) hold for $k{=}0, 1, \ldots$. Because of the surjectivity assumption, given any sequence of $x_2(n-1), \ldots, x_2(m+n)$ which transfers the state $x_1(n-1)$ to the origin in $m+1$ time steps, we can compute a sequence of $x_3(n-2), \ldots, x_3(m+n-1)$, which realises $x_2(n-1), \ldots, x_2(m+n)$. By repeating the same argument, we can find a sequence of controls $u(0), u(1), \ldots, u(m)$ which realises the desired sequence $x_2(n-1), \ldots, x_2(m+n)$. Moreover, because of the triangular structure of the system we have that

$$\begin{aligned}
u(0) &= U_0(x_1(0), x_2(0), \ldots, x_n(0)) \\
u(1) &= U_1(x_1(0), x_2(0), \ldots, x_n(0), u(0)) \\
&\ldots
\end{aligned} \tag{9.6}$$

and hence we can arbitrary assign one control at a time for any initial state and any previously applied controls. Of course, the controls may not be unique. So the desired sequence of controls $u(0), \ldots, u(m)$ can be computed for any initial state $x(0) \in \mathbb{R}^n$ which yields $x_1(m+n+1) = 0$.

Consider now the equation

$$x_2(m+n+1) = f_2(x_1(m+n), x_2(m+n), x_3(m+n)) \tag{9.7}$$

Then because of the surjectivity assumption and using a similar argument we can show that there exists $u(m+1)$ which zeroes $x_2(m+n+1)$. In the same manner, we can find $u(m+2)$ which zeroes $x_3(m+n+1)$, etc. Therefore, for any initial state $x(0) \in \mathbb{R}^n$ it is possible to compute a finite sequence of controls $u(0), \ldots, u(m+n-1)$ which zeroes the state of the overall system in $m+n$ time steps. Then by simply applying $u(k) = 0, \forall k \geq m+n$ the state is kept at zero. Q.E.D.

A form of the above given theorem was first proved in [159]. However, if we combine it with Theorem 5.1 in Chapter 5, the dead-beat controllability test becomes more explicit. This motivates the following:

Assumption 9.2 The non-linearity in (9.3) is assumed to be polynomial, that is $f_1(x_1, x_2) \in \mathbb{Q}[x_1, x_2]$. □

Corollary 9.1 The system (9.2) with Assumptions 9.1 and 9.2 is state dead-beat controllable if and only if the scalar polynomial system (9.3) is state dead-beat controllable by means of x_2. □

In other words, the system (9.2) with Assumptions 9.1 and 9.2 is state dead-beat controllable if and only if conditions of Theorem 5.1 are satisfied for the scalar polynomial subsystem (9.3) when x_2 is viewed as a control signal.

Comment 9.1 By using the result of [159] we see that the overall system inherits the dead-beat controllability properties of the subsystem (9.3). In this way, when using QEPCAD we do not need to deal with the overall system but only with (9.3) while regarding $x_2(k)$ as the control input. Notice that the order of (9.3) subsystem n_1 may be much smaller than the order of the overall system n. This reduces computational time of the controllability TESTS 1 and 2 in Chapter 3. □

Comment 9.2 Class 1 systems have very simple structure as far as state dead-beat controllability is concerned. Nevertheless, it is very instructive to consider classes of block oriented models which fall into this category so that one can easily recognise when one can use the above described simplification when considering the dead-beat problem. Suppose that we have a linear dynamical system:

$$\begin{aligned} x_1(k+1) &= \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & & & \dots & \\ 0 & 0 & 0 & \dots & 1 \\ a_1 & a_2 & a_3 & \dots & a_n \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{pmatrix} u(k) \\ y(k) &= (1\ 0\ \dots\ 0)\, x(k) \end{aligned} \tag{9.8}$$

The z transfer function of this system has the form $W_p(z) = \frac{c}{a(z)}$, $c \neq 0$. It is trivial to show that this system is output controllable and we can easily obtain several classes of block oriented models for which one can use the results of this section. The systems are obtained as a series connection between the linear system W_p and nonlinear systems. Several examples are shown in Figure 9.1. In all of the examples the overall system is dead-beat controllable if the system 2 is dead-beat controllable (see Figure 9.1). Notice that in all examples we have explicit controllability tests for systems 2. We emphasize that one may obtain more complicated interconnected systems that fall into this category. □

9.2.1 Minimum-Time Dead-Beat Controller

We will keep the notation S_k to denote the set of states x_1 of the scalar subsystem (9.3) that can be transferred to the origin in at most $k+1$ time steps. Their defining expression are $S_k(x_1)$. On the other hand, we introduce the notation O_k to denote the set of states of the overall system (9.2) with the same property, i.e. states in these sets can be mapped to the origin in at most $k+1$ time

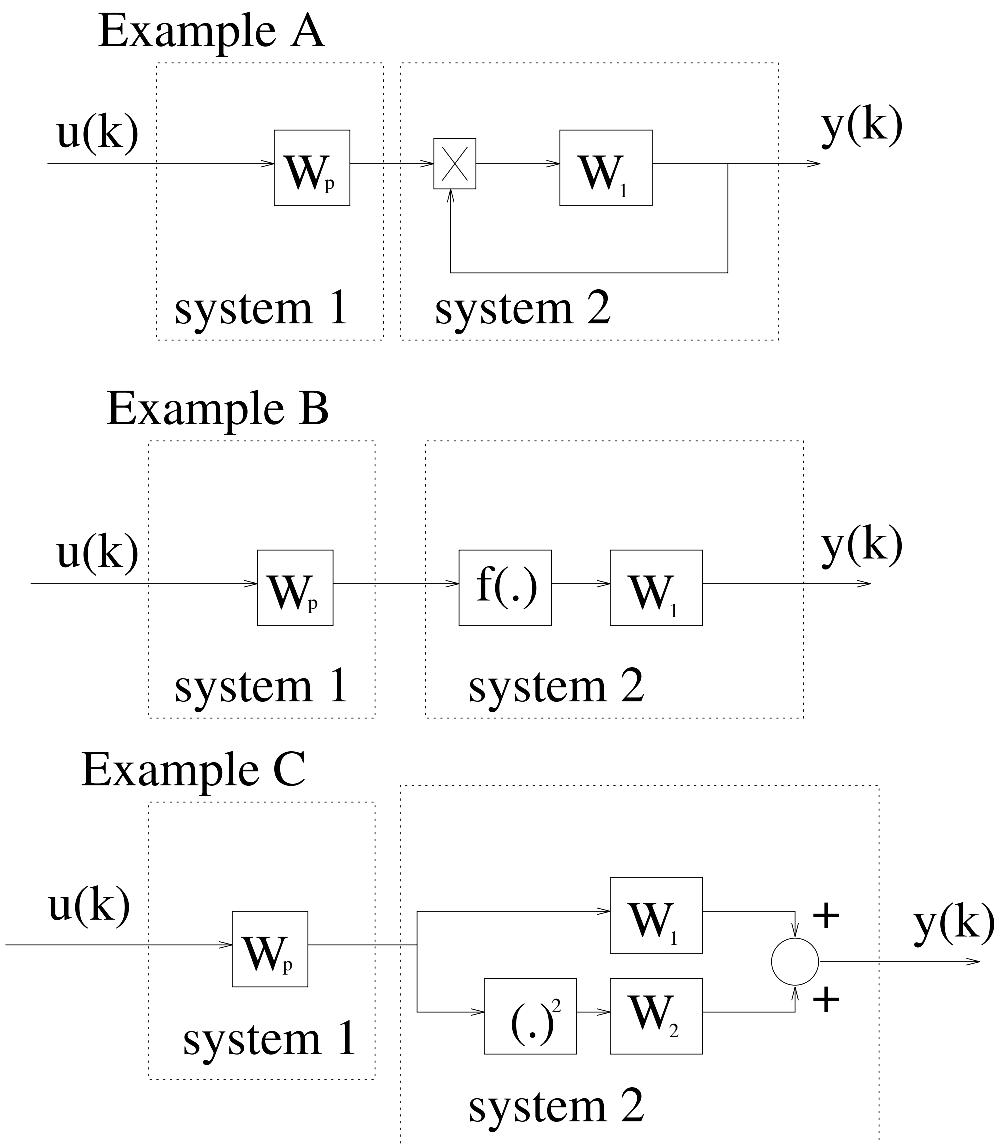

Figure 9.1: Examples of Class 1 block oriented models: series connection of the linear system W_p and a bilinear system (Example A); simple Wiener-Hammerstein system (Example B); series connection of the linear system W_p and a generalised Hammerstein system (Example C).

steps respectively. The defining expressions for sets O_k are denoted as $O_k(x)$.

In this section, we also use the notation $f_0{=}f(x,0)$ and $f_0^k(x) = f_0 \circ f_0 \circ \ldots \circ f_0(x)$. $f_0^0(x)$ is the identity mapping. Also, we write $f_i(x)$ although f_i depends only on $x_1, \ldots, x_{i+1}$. Notice that it is straightforward to compute the expressions $S_k(x_1)$ that define the sets S_k for the scalar polynomial subsystem using the methodology in Chapter 3 by regarding x_2 as a control signal.

The computation of $S_k(x_1)$ is the first step in the design of the time-optimal dead-beat controller for the overall system.

Notice that the following holds:

$$\begin{aligned}
O_0 &= \{x \in \mathbb{R}^n : f_1(x) = 0,\ f_2(x) = 0, \ldots,\ f_{n-1}(x) = 0\} \\
O_1 &= \{x \in \mathbb{R}^n : f_1 \circ f_0(x) = 0,\ f_2 \circ f_0(x) = 0, \ldots,\ f_{n-2} \circ f_0(x) = 0\} \\
O_2 &= \{x \in \mathbb{R}^n : f_1 \circ f_0^2(x) = 0,\ f_2 \circ f_0^2(x) = 0, \ldots,\ f_{n-1} \circ f_0^2(x) = 0\} \\
& \ldots \\
O_{n-2} &= \{x \in \mathbb{R}^n : f_1 \circ f_0^{n-2}(x) = 0\} \\
O_{n-1} &= \{x \in \mathbb{R}^n : S_0(f_1 \circ f_0^{n-2}(x))\} \\
O_n &= \{x \in \mathbb{R}^n : S_1(f_1 \circ f_0^{n-2}(x))\} \\
& \ldots \\
O_{n+m-2} &= \{x \in \mathbb{R}^n : S_{m-1}(f_1 \circ f_0^{n-2}(x))\}
\end{aligned}$$

Notice that the sets $O_{n-1}, \ldots, O_{n+m-2}$ are obtained directly from the expressions $S_k(x_1)$ by substituting $f_1 \circ f_0^{n-2}(x)$ instead of x_1. Hence, QEPCAD needs to be used only for the scalar problem to obtain $S_k(x_1)$ from which the expressions $O_k(x)$ follow easily.

Consider now the choice for control which yields time optimal behaviour. On the sets $O_k,\ k{=}0, 1, \ldots, n-2$ we need to apply control obtained as any real solution $u \in \mathbb{R}$ to the equation:

$$f_{n-k} \circ f_0^{k-1} \circ f_u(x) = 0 \tag{9.9}$$

which depends only on the control variable u and the measured state $x \in \mathbb{R}^n$. If the measured state x belongs to $O_k,\ k{=}n-1, \ldots, n+m-2$ then we can apply a control which is designed for the scalar subsystem with an appropriate inversion. We denote $S_k(f_1 \circ f_0^{n-2})$, $k{=}n-1, \ldots, m+n-2$ as $\tilde{S}_k(x)$ (recall the notation of Chapter 3) and write $\tilde{S}_k \circ f_u(x) \equiv \tilde{S}_k(f(x,u))$. On the sets O_{n-1}

we need to apply the control

$$u = \text{any real root to } f_1 \circ f_0^{n-2} \circ f_u(x) = 0$$

and on the sets $O_k, k=n, \ldots, n+m-2$ we apply the control

$$u = \text{any real root to } \tilde{S}_{k-1} \circ f_u(x)\,.$$

We emphasize that it is straightforward to modify the above presented time-optimal controller for cases when the subsystem (9.3) is not scalar, that is $x_1 \in \mathbb{R}^{n_1}, n_1 > 1$. The important point is that QEPCAD is used to compute a time-optimal controller for the subsystem, the dimension (n_1) of which may be much smaller than the dimension of the overall systems (n). Therefore, the topology of the interconnections of Class 1 systems leads to immense savings in computation time when designing a minimum-time dead-beat controller by means of QEPCAD.

9.2.2 Class 1: Examples

In the following examples we use the notation of Chapter 5 for the maximal invariant set S_I, control independent set $\bar{S}$ and trivial invariant set S_T.

Example 9.1 Consider the second order system:

$$\begin{aligned} x_1(k+1) &= x_1^2(k)\,(1 - x_1^2(k)\,) + x_2^2(k) \\ x_2(k+1) &= u(k) \end{aligned} \tag{9.10}$$

The scalar system

$$x_1(k+1) = x_1^2(k)\,(1 - x_1^2(k)\,) + x_2^2(k)$$

is two step dead-beat controllable when x_2 is viewed as control (see Example 2.2). So the whole system is dead-beat controllable. Indeed, by direct computation we obtain:

$$\begin{aligned} O_0 &= \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2(1 - x_1^2) + x_2^2 = 0\} \\ O_1 &= \{(x_1, x_2) \in \mathbb{R}^2 : |x_1^2(1 - x_1^2) + x_2^2| \geq 1\} \\ O_2 &= \{(x_1, x_2) \in \mathbb{R}^2 : |x_1^2(1 - x_1^2) + x_2^2| < 1\} \end{aligned} \tag{9.11}$$

It is interesting to compare the properties of the scalar system:

$$x(k+1) = x^2(k)\,(1 - x^2(k)\,) + u^2(k) \tag{9.12}$$

with the properties of the overall system. The scalar system (9.12) is dead-beat controllable (see Chapter 5). However, it is interesting that any minimum-time dead-beat controller destabilises the system. This is because on the neighbourhood of the origin $N =]-1, 1[$ there is no control that transfers an initial state $x(0) \in N$ to the origin in one step and, consequently, in order to transfer any initial state from the neighbourhood N to the origin, we need to exit the neighbourhood, from which it follows that any static discontinuous dead-beat controller yields an unstable system.

We can easily see that the same property is preserved by the overall system (9.10). Namely, in order to map any state from the neighbourhood $N_o = \{x : x_1^2 + x_2^2 < 0.5\}$ to the origin we need to exit this neighbourhood. In the examples that we give this always happens: not only does the overall system have the same controllability properties as the underlying scalar polynomial system, but the minimum-time control laws yield very similar qualitative behaviour for both systems. So if there is no stabilising dead-beat controller for the scalar subsystem, it is natural that the overall system would have the same property. This is not true in general and it is a direct consequence of the special triangular structure of Class 1 systems. □

Example 9.2 Consider the triangular multi-linear system given by:

$$\begin{aligned} x_1(k+1) &= (x_1(k) + 1)\, x_2(k) + x_1(k) \\ x_2(k+1) &= 3x_1(k) - x_1(k)\, x_2(k) - u(k) \end{aligned} \tag{9.13}$$

If $x_1(0) = -1$, the scalar subsystem is decoupled from the rest of the system and we have that $x_1(k) = -1,\ k = 1, 2, \ldots$. There is one invariant set and $S_I = \{-1\}$ (an equilibrium insensitive to control). Since $x_1(0) = -1$ can not be transferred to the origin, we have that $S_T = \emptyset$ and hence $S_I \neq S_T$. The scalar subsystem is not dead-beat controllable and neither is the overall system. This example illustrates the necessity of the condition $S_I = S_T$. □

9.3 Class 2

In this section we consider triangular systems given by:

$$\begin{aligned} x_1(k+1) &= f_1(x_1(k), u(k)) \\ x_2(k+1) &= f_2(x_1(k), x_2(k), u(k)) \\ x_3(k+1) &= f_3(x_1(k), x_2(k), x_3(k), u(k)) \\ \dots & \quad \dots \\ x_t(k+1) &= f_t(x_1(k), \dots, x_t(k), u(k)) \end{aligned} \tag{9.14}$$

where $x_i \in \mathbb{R}^{n_i}, u \in \mathbb{R}$ and $\sum n_i = n$. We have $f_i = (f_{i1}\ f_{i2}\ \dots\ f_{in_i})^T$ and the nonlinearities $f_{ij},\ i=1,\dots,n,\ j=1,\dots,n_i$ are polynomials in all their variables $f_{ij} \in \mathbb{R}[x_1, x_2, \dots, x_i, u]$. We also use the following assumption:

Assumption 9.3 The following is satisfied:

$$\forall u \in \mathbb{R},\ f_1(0,u)=0; \forall u \in \mathbb{R},\ f_2(0,0,u)=0; \dots;$$

$$\forall u \in \mathbb{R},\ f_{t-1}(0,0,\dots,0,u)=0; \exists u \in \mathbb{R},\ f_t(0,0,\dots,0,u)=0$$

$\square$

It is interesting that Class 1 systems can be regarded as a class of discrete-time systems for which the “backstepping” procedure [109] can be used whereas for Class 2 systems we can use a kind of “forwarding” procedure [158] for the dead-beat problem.

Theorem 9.2 *A Class 2 system (9.14) with Assumption 9.3 is state dead-beat controllable if each of the subsystems (viewed as systems from control u_i to state x_i) defined by:*

$$\begin{aligned} x_1(k+1) &= f_1(x_1(k), u_1(k)) \\ x_2(k+1) &= f_2(0, x_2(k), u_2(k)) \\ x_3(k+1) &= f_3(0, 0, x_3(k), u_3(k)) \\ \dots & \quad \dots \\ x_t(k+1) &= f_t(0, \dots, 0, x_t(k), u_t(k)) \end{aligned} \tag{9.15}$$

is state dead-beat controllable. □

Proof of Theorem 9.2 (sufficiency): Since the subsystem $x_1(k+1)=f_1(x_1(k),u(k))$ is state dead-beat controllable, for any $x_1(0)\in\mathbb{R}^{n_1}$ there exists a sequence of controls which transfers x_1 to zero in finite time. Because of Assumption 9.3 we can then keep $x_1(k)=0$ while applying any arbitrary control sequence in order to zero x_2. This is possible since $x_2(k+1)=f_2(0,x_2(k),u(k))$ is dead-beat controllable. The proof trivially follows. Q.E.D.

Notice that dead-beat controllability of the first subsystem (x_1,u_1), is also necessary for dead-beat controllability of the overall system. However, the remaining conditions are not necessary.

Comment 9.3 Assumption 9.3 is easily checked. One can easily identify a structure of state affine systems [163, 117] that belong to Class 2:

$$x(k+1)=\begin{pmatrix} A_{11}(u(k)) & 0 & \dots & 0 \\ A_{21}(u(k)) & A_{22}(u(k)) & \dots & 0 \\ \dots & & \dots & \\ A_{t1}(u(k)) & A_{t2}(u(k)) & \dots & A_{tt}(u(k)) \end{pmatrix} x(k)+\begin{pmatrix} 0 \\ 0 \\ \dots \\ g(u(k)) \end{pmatrix} \tag{9.16}$$

where $A_{ij}(u(k))$ are matrices whose entries are polynomial in $u(k)$ and $g(u(k))$ is a vector whose entries are polynomials in $u(k)$. Also, the following condition must be satisfied

$\exists u\in\mathbb{R}, g(u)=0$. □

Comment 9.4 Class 2 systems are very interesting because the state dead-beat controllability test of the system is obtained by combining a number of controllability tests of lower dimensional subsystems. In this way, simpler controllability tests, such as the one for scalar polynomial systems in Chapter 5, can be repeatedly used to check dead-beat controllability of higher dimensional systems. We again present some examples of block oriented models which belong to Class 2 systems. They are shown in Figure 9.2.

We emphasize that even if analytic controllability tests do not exist for the subsystems, the general controllability tests (TEST 1 and 2) of Chapter 3 are simplified since QEPCAD is used for a number of lower dimensional systems. □

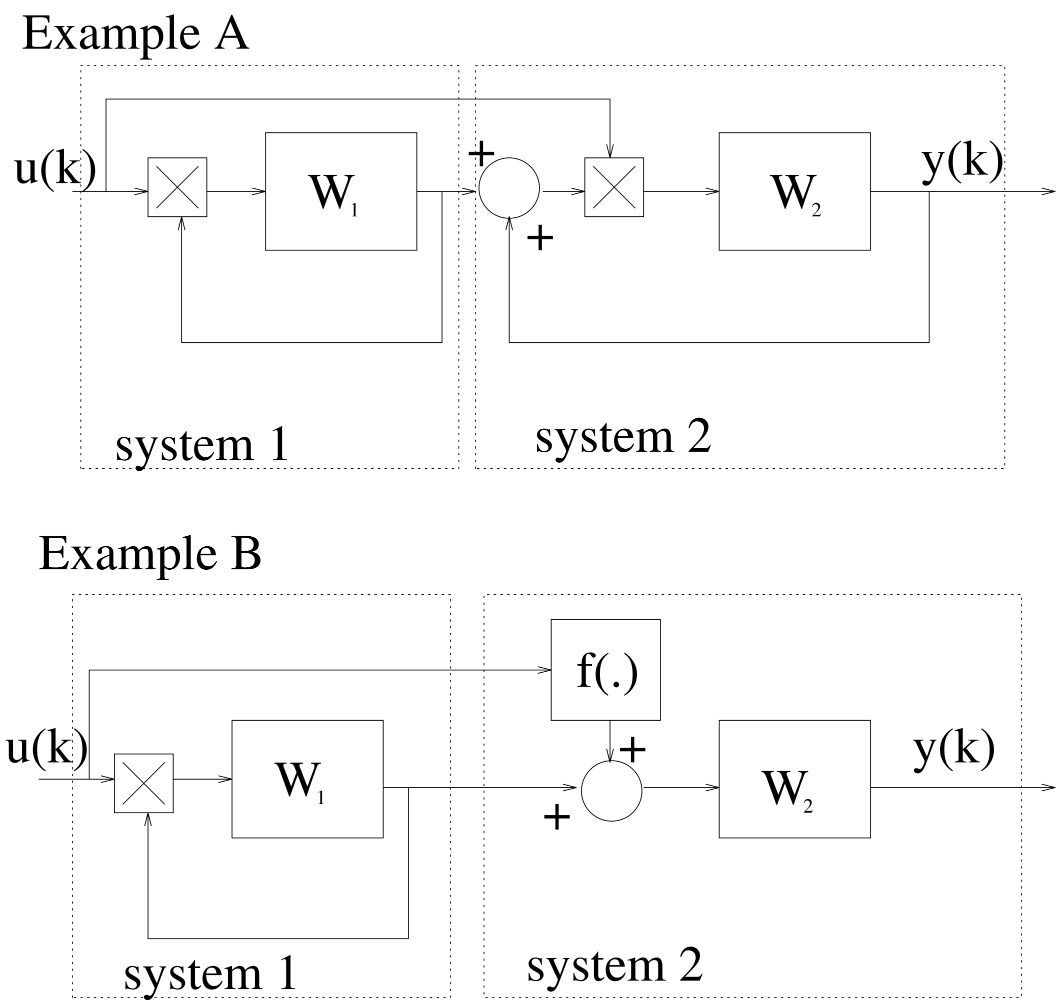

Figure 9.2: Examples of block oriented Class 2 systems: a series connection of two bilinear systems (Example A); a connection of a bilinear system and a simple Hammerstein system (Example B)

9.3.1 Class 2: Examples

Example 9.3 Consider the system:

$$\begin{aligned} x_1(k+1) &= -x_1^2(k) + x_1(k)\,u(k) + x_1^4(k)\,u^4(k) \\ x_2(k+1) &= x_1(k)\,x_2^2(k) + x_1^2(k) + x_2^3(k) - x_2 + u(k) - x_2^3(k)\,u^6(k) \end{aligned} \tag{9.17}$$

Assumption 9.3 is satisfied since for $x_1(0)=0$ we have that $x_1(1)=0, \forall u(0) \in \mathbb{R}$. Also, if $x_1(0)=0, x_2(0)=0$ by applying $u(0)=0$ we have that $x_2(1)=0$. First, we check state dead-beat controllability of the even subsystem:

$$x_1(k+1) = -x_1^2(k) + x_1(k)\,u(k) + x_1^4(k)\,u^4(k)$$

using the methodology of Chapter 5. The control independent set is $\bar{S}=\{0\}$. Also, the maximal invariant set $S_I=\{0\}$ and trivial invariant set $S_T=\{0\}$. Hence, $S_I=S_T$, which is a necessary condition for state dead-beat controllability. Consider the equation

$$-x_1^2 + x_1 u + x_1^4 u^4 = 0$$

It has at least two real solutions u for any $x_1 \neq 0$ because $-x_1^2$ and x_1^4 have opposite sign. Hence, the first subsystem is state dead-beat controllable according to Theorem 5.1.

Consider now the second subsystem (with $x_1(k) \equiv 0$):

$$x_2(k+1) = x_2^3(k) - x_2(k) + u(k) - x_2^3(k)\,u^6(k)$$

In this case $\bar{S}=S_I=\emptyset$. Also, since $x_2^3 - x_2$ and $-x_2^3$ have opposite signs for x_2 in intervals $]-\infty,-1[$ and $]1,+\infty[$ the equation $x_2^3 - x_2 + u - x_2^3 u^6 = 0$ has at least two real solutions u. Hence, we have that $]-\infty,-1[\cup]1,+\infty[\subset S_0$ and the subsystem is state dead-beat controllable using Theorem 5.1. We conclude that the system (9.17) is state dead-beat controllable. □

Example 9.4 Consider the system:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_1(k) + x_2(k) - 2x_1(k)\,u(k) + 3x_2 u(k) \end{aligned}$$

$$
\begin{aligned}
x_3(k+1) &= x_1(k) + x_2^2(k) + x_3(k) \\
x_4(k+1) &= x_1^3(k) - x_3(k) - 3x_4(k) + u^4(k)
\end{aligned}
\tag{9.18}
$$

Assumption 9.3 is satisfied. The controllability test can be carried out in two steps by considering state dead-beat controllability of a bilinear system and a simple Hammerstein system. State dead-beat controllability of the bilinear subsystem

$$
\begin{aligned}
x_1(k+1) &= x_2(k) \\
x_2(k+1) &= x_1(k) + x_2(k) - 2x_1(k)\,u(k) + 3x_2 u(k)
\end{aligned}
\tag{9.19}
$$

can be checked by using results of [127]. Using the periodic control sequence $u(2k)=0.5$ and $u(2k+1)=-0.63636363, \forall k$, we zero x_1 and x_2 in 3 time steps.

The second subsystem is given by (x_1, x_2=0):

$$
\begin{aligned}
x_3(k+1) &= x_3(k) \\
x_4(k+1) &= -x_3(k) - 3x_4(k) + u^4(k)
\end{aligned}
\tag{9.20}
$$

This simple Hammerstein system (or linear system with positive controls) is completely (and therefore state dead-beat) controllable by using results of [50] (see Chapter 7). With the notation:

$$
A=\begin{pmatrix} 0 & 1 \\ -1 & -3 \end{pmatrix}, \quad b=\begin{pmatrix} 0 \\ 1 \end{pmatrix}
$$

we have that det$[Ab : b] \neq 0$. The eigenvalues of A are -0.381 and -2.618. Since (A, b) is a controllable pair and A has no real positive eigenvalues, the system is completely controllable (and therefore dead-beat controllable). We conclude that the overall system is dead-beat controllable. Observe that there is a uniform bound on the dead-beat time for this system. Notice that using a straightforward method one can design a non-minimum-time dead-beat controller for the overall system. Time-optimal control law does not follow easily from this particular interconnected structure. □

9.4 Class 3

In this section we present a methodology which shows how one can use QEPCAD and the Gröbner basis method, together with some structural assumptions in order to obtain a computationally less expensive state dead-beat controllability test than the general tests presented in Chapter 3. Results of Chapters 3 and 4 are crucial to understand the material presented in this section. Moreover, there is a direct analogy between Class 3 systems, which we consider here, and odd systems considered in Chapter 4. Consequently, we use the same terminology to define objects which are analogous to those already introduced in Chapter 4. No confusion should arise since we present new equations that are used to define the critical variety V_C of this section. We also show how it is possible to use the Gröbner basis method (without resorting to QEPCAD) for some sub-classes of Class 3 systems to decide on state dead-beat controllability.

Consider a polynomial system:

$$x(k+1) = f(x(k), u(k)) \tag{9.21}$$

with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}$.

For a large class of polynomial systems, the set S_{n-2} is a variety defined by a polynomial $G(x) = 0$. Indeed, consider the set of equations $f_{u(n-2)} \circ \ldots \circ f_{u(0)}(x) = 0$ which are used to define the set S_{n-2}. We have n equations with $n-1$ controls (parameters) which very often can be eliminated (using QEPCAD or the Gröbner basis method) to obtain a single polynomial $G(x) \in \mathbb{R}[x]$ that defines the set S_{n-2} (with the notation of Chapter 3 we write $S_{n-2}(x) \equiv G(x) = 0$). This motivates the following assumption:

Assumption 9.4 *We assume that:*

1. $S_{n-2} = \{x : G(x) = 0\}$

2. $$G \circ f_u(x) = g_0(x) + g_1(x)\, u + \ldots + g_m(x)\, u^m \tag{9.22}$$

   and n is an odd integer. □

Notice that the set S_{n-1} is the whole state space except perhaps for the states that belong to the "critical variety" defined by $V_C = V(g_m)$, where the polynomial g_m is defined in (9.22). It is immediately clear that we can use the methodology presented in Chapter 4 to compute the maximal invariant subset of the critical variety ($V_I \subseteq V_C$) using the Gröbner basis method. We use the terminology and definitions as presented in Chapter 4. The only difference is that in this section we want to reach the origin $\{0\}$ from any initial state and not the zero output variety, which was introduced in Chapter 4. Also, we can redefine the trivial invariant set V_T of Chapter 4 as a subset of the maximal invariant set V_I from which we can reach the origin in finite time. V_T can be computed using QEPCAD in an obvious way using the methodology of Chapters 3 and 4.

The following results are immediately adopted from Chapter 4:

Theorem 9.3 *A polynomial system of the form (9.21) with Assumption 9.4 is state dead-beat controllable if and only if* $V_I{=}V_T$. □

Theorem 9.4 *A polynomial system of the form (9.21) with Assumption 9.4 is state dead-beat controllable if* $V_I{=}\emptyset$. □

Theorem 9.5 *Suppose* $V_I{\neq}\emptyset$. *A polynomial system of the form (9.21) with Assumption 9.4 is state dead-beat controllable only if the origin belongs to* V_I. □

The fact that we can identify a critical variety V_C which has a lower dimension than the order of the systems usually reduces the required computations. Indeed, since we have the inclusions $V_T \subseteq V_I \subseteq V_C$ and $\dim(V_C) \leq n-1$, we can use the method of equality constraints explained in [33] which helps in reducing computations of the set V_T by using QEPCAD.

The class of systems (9.21) satisfying Assumption 9.4 is not trivial. We emphasize that it is difficult to characterise nonlinearities $f_i(x,u)$ in (9.21) which satisfy Assumption 9.4. However, several subclasses of polynomial systems that generically satisfy the assumption are identified below.

Notice that the main issue here is the existence of a variety with the property that all states outside of it can be transferred to the origin. The states in the critical variety may be or may be not transferable to the origin. This implies that we may work with "much larger" critical variety which contains many "good" states as well. For instance, suppose that we have obtained using QEPCAD that at some step K the set $S_K{=}\mathbb{R}^n - \{x : x_1{=}0, x_2{=}0, \ldots, x_{n-1}{=}0, x_n > 0\}$. So the critical set is in this case a half line. However, nothing stops us from defining the critical variety $V_C{=}\{x : x_1{=}0, x_2{=}0, \ldots, x_{n-1}{=}0\}$, which obviously contains all "critical states" but also some "good" states. Then we can then apply the same methodology to compute the maximal invariant set of the critical variety.

In certain situations it may be straightforward or easier to compute a larger critical variety. We present below such an approach based on the Gröbner basis method which can be used for the class of polynomial systems in strict feedback form (see [109]) to obtain polynomials that define V_C.

9.4.1 Strict Feedback Polynomial Systems

The class of strict feedback polynomial systems that we consider is defined as:

$$\begin{aligned}
x_1(k+1) &= F_1(x_1(k)) + G_1(x_1(k))\, x_2(k) \\
x_2(k+1) &= F_2(x_1(k), x_2(k)) + G_2(x_1(k), x_2(k))\, x_3(k) \\
\dots &\quad \dots \\
x_n(k+1) &= F_n(x_1(k), \dots, x_n(k)) + G_n(x_1(k), \dots, x_n(k))\, u(k)
\end{aligned} \tag{9.23}$$

with $x_i \in \mathbb{R}, \forall i{=}1, \dots, n$ and $u \in \mathbb{R}$. We also have that $G_i, F_i \in \mathbb{Q}[x_1, \dots, x_i]$. Notice the difference between (9.24) and Class 1 systems: the functions $F_i(x_1, \dots, x_i) + G_i(x_1, \dots, x_i)\, x_{i+1}$ *may not be surjective in* $x_{i+1}, \forall x \in \mathbb{R}^n$. In other words, we allow for the possibility that the real varieties $V(G_i), i{=}1, 2, \dots, n$ are not empty.

We denote $x(k) = (x_1(k) \;\; x_2(k) \;\; \dots x_n(k))^T$. If we take n compositions of this map, starting from $x(0) \in \mathbb{R}^n$, we obtain

$$\begin{aligned}
x_1(n) &= c_1(x(0)) + d_1(x(0))\, u(0) \\
x_2(n) &= c_2(x(0), u(0)) + d_2(x(0), u(0))\, u_1(0) \\
\dots &\quad \dots \\
x_n(n) &= c_n(x(0), u(0), \dots, u(n-2)) + d_n(x(0), u(0), \dots, u(n-2))\, u(n-1)
\end{aligned} \tag{9.24}$$

where c_i, d_i are polynomials obtained using straightforward computations. Observe the triangular structure with respect to controls $u(i), i{=}0, 1, \dots, n-1$.

Let us now compute equations that define V_C. It is obvious that if $x(0)$ is such that $d_1(x(0)) \neq 0$ we can assign zero value to $x_1(n)$ by means of $u(0)$. Hence, any state that belongs to the real variety $V(d_1)$ may not be transferable to the origin in n steps. It is crucial to assume that $V(d_1)$ is not equal to $\Re^n$, that is $d_1(x) \not\equiv 0$. Otherwise, the critical variety V_C would be equal to $\mathbb{R}^n$, which is not a lower dimensional subset of the state space. In such cases, the method presented in this section can not be used to simplify considerations and one should resort to the direct use of QEPCAD as described in Chapter 3.

Consider now the first two equations in (9.25). Let us find the states for which the first equation can be made equal to zero whereas the second can not. In this case, we necessarily have that

$$\begin{aligned} c_1(x(0)) + d_1(x(0))\, u(0) &= 0, \\ d_2(x(0), u(0)) &= 0 \end{aligned}$$

If we apply the Gröbner basis method to eliminate $u(0)$ from the two above given equations using the lexicographic ordering $u(0) \succ x_1(0) \succ \ldots \succ x_n(0)$, we can almost always obtain a polynomial in $x(0)$ alone that defines the set of critical states[1]. That is we very often have that:

$$\text{Gbasis}[c_1(x(0)) + d_1(x(0))\, u(0), d_2(x(0), u(0))] =$$

$$\{l_{1,1}(x(0), u(0)), \ldots, l_{1,p_1-1}(x(0), u(0)), l_{1,p_1}(x(0))\}$$

Hence, if the initial state belongs to the variety $V(l_{1,p_1})$, we may not be able to zero simultaneously $x_1(n)$ and $x_2(n)$ in (9.25). We can continue in the same way to consider the first three equations in (9.25). After eliminating $u(0)$ and $u(1)$ from

$$\begin{aligned} c_1(x(0)) + d_1(x(0))\, u(0) &= 0, \\ c_2(x(0), u(0)) + d_2(x(0), u(0))\, u(1) &= 0, \\ c_3(x(0), u(0), u(1)) &= 0 \end{aligned}$$

we obtain a polynomial $l_{2,p_2}(x(0))$ which defines the set of states for which the first three equations may not be simultaneously rendered zero. In the same manner we can obtain the polynomials $l_{i,p_i}(x(0))$, i=$1, 2, \ldots, n-1$ of critical states for which the first $i+1$ equations may not be rendered zero simultaneously. The critical variety is then obtained as a union of varieties defined

[1] There are cases in which the elimination is simply not possible, that is the Gröbner basis does not contain polynomials in $x(0)$ only. However, these cases appear to be non generic (see Example 9.6).

by the obtained polynomials. That is, we have

$$V_C = V(d_1 \cdot l_{1,p_1} \cdot \ldots \cdot l_{n-1,p_{n-1}})$$

Now we can check the existence of maximal invariant sets using the Gröbner basis method of Chapter 4.

Comment 9.5 The same method may be used to compute the critical variety for the following classes of polynomial systems:

1. **Pure feedback polynomial systems investigated in [109]:**

$$\begin{aligned} x_1(k+1) &= F_0^1(x_1(k)) + F_1^1(x_1(k))\, x_2(k) + \ldots + F_{n_1}^1(x_1(k))\, x_2^{m_1}(k) \\ \ldots &\quad \ldots \\ x_n(k+1) &= F_0^n(x_1(k), \ldots, x_n(k)) + \ldots + F_{m_n}^n(x_1(k), \ldots, x_n(k))\, u^{m_n}(k) \end{aligned} \tag{9.25}$$

   where m_i are odd integers $\forall i = 1, 2, \ldots, n$. Also, $F_i^j \in \mathbb{R}[x], \forall i, j$.

2. **A sub-class of NARMAX models investigated in [75, 76]:**

$$y(k+1) = F(y(k), y(k-1), \ldots, y(k-s), u(k)) \tag{9.26}$$

   where

$$F(y_1, \ldots, y_{s+1}, u) = g_0(y_1, \ldots, y_{s+1}) + \ldots + g_m(y_1, \ldots, y_{s+1})\, u^m$$

   with $g_i \in \mathbb{R}[y_1, \ldots, y_{s+1}], i = 0, 1, \ldots, m$ and m is an odd integer.

3. **Homogeneous bilinear systems investigated in [48, 71, 70]:**

$$x(k+1) = (A + u(k)\, bc)\, x(k) \tag{9.27}$$

   where $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^{n \times 1}$, $c \in \mathbb{R}^{1 \times n}$ with (A, b) controllable and (c, A) observable pairs.

4. **Inhomogeneous bilinear systems investigated in [49]:**

$$x(k+1) = (A + u(k)\, bc)\, x(k) + du(k) \tag{9.28}$$

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^{n\times 1}$, $c \in \mathbb{R}^{1\times n}$ with (A, b) controllable, (c, A) observable and rank$[cb : d] = 1$.

We note that explicit controllability tests have been found in [49, 71, 70, 49] for the above classes of bilinear systems. The method that we propose is more tedious in these cases. However, for more general pure and strict feedback systems our method is to the best of our knowledge the only method to test state dead-beat controllability. It is interesting to observe that the bilinear systems fall into this category.

Several examples of block oriented models that normally belong to Class 3 systems are shown in Figure 9.3.

9.4.2 *Class 3: Examples*

Example 9.5 Consider the NARMAX model:

$$y(k+1) = y^2(k-2) + (y(k-2) - y(k))\,u(k) \tag{9.29}$$

We introduce the state variables $x_1(k) = y(k-2)$, $x_2(k) = y(k-1)$, $x_3(k) = y(k)$ and write the state space model:

$$\begin{aligned}
x_1(k+1) &= x_2(k) \\
x_2(k+1) &= x_3(k) \\
x_1(k+1) &= x_1^2(k) + (x_1(k) - x_3(k))\,u(k)
\end{aligned} \tag{9.30}$$

Consider the state $x(3)$ at time step 3:

$$\begin{aligned}
x_1(3) &= x_1^2(0) + (x_1(0) - x_3(0))\,u(0) \\
x_2(3) &= x_2^2(0) + [x_2(0) - x_1^2(0) - (x_1(0) - x_3(0))\,u(0)]\,u(1) \\
x_1(3) &= x_3^2(0) + [x_3(0) - [x_2(0) - (x_1^2(0) + (x_1(0) - x_3(0))\,u(0))]\,u(1)]\,u(2)
\end{aligned} \tag{9.31}$$

The first equation may not be possible to zero if the initial state belongs to the variety $V(x_1 - x_3)$.

Consider now for which states we can zero the first equation whereas the second one may not be possible to zero. For convenience we drop time indices for states, that is we write x_1 instead of

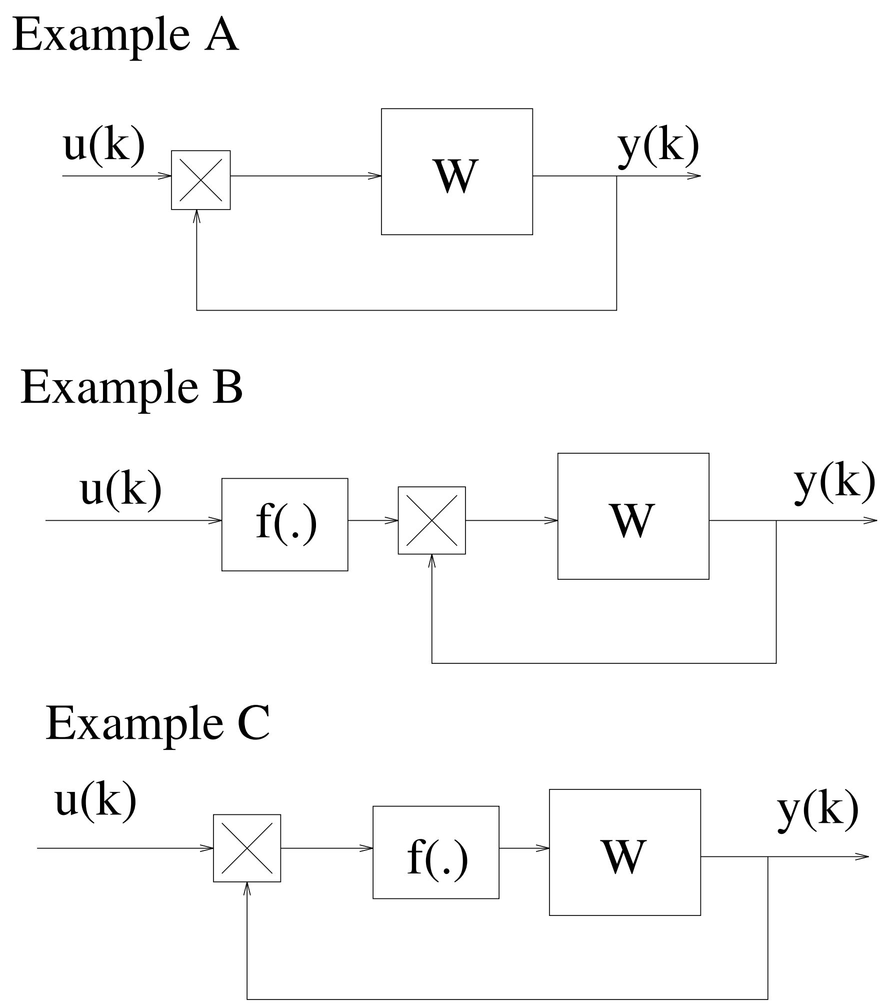

Figure 9.3: Examples of block oriented Class 3 systems: linear system with a multiplicative feedback (Example A); simple Hammerstein system with a multiplicative feedback (Examples B and C)

$x_1(0)$. Also, we use subscripts to denote time steps for controls. We compute the Gröbner basis

$$\text{Gbasis}[x_1^2 + (x_1 - x_3)\, u_0, x_2 - x_1^2 - (x_1 - x_3)\, u_0]$$

with lexicographic ordering $u_0 \succ x_1 \succ x_2 \succ x_3$. The obtained basis consists only one polynomial which does not depend on u_0, namely the polynomial x_2. Hence, the states that belong to the variety $V(x_2)$ are also critical.

Finally, we compute the following basis:

$$\text{Gbasis}[x_1^2 + (x_1 - x_3)\, u_0, x_2^2 + (x_2 - x_1^2 - (x_1 - x_3)\, u_0)\, u_1, x_3 - (x_2 - (x_1^2 + (x_1 - x_3)\, u_0))\, u_1]$$

with lexicographic ordering $u_1 \succ u_0 \succ x_1 \succ x_2 \succ x_3$. The only polynomial in the computed basis that does not depend on u_1 and u_0 is $x_3 + x_2^2$. Thus, for the states in the variety $V(x_3 + x_2^2)$ we may zero the first two equations but not necessarily the third one.

As a result, we obtain that the critical variety is given by $V_C = V(x_2(x_1 - x_3)(x_3 + x_2^2))$. By using the Gröbner basis method from Chapter 4 we obtain that the maximal invariant set is $V_I = \{(0,0,0), (1,1,1)\}$. By simple calculations one can verify that both of these states are invariant sets themselves and hence the state $(1,1,1)$ can not be transferred to the origin. The system is not state dead-beat controllable. □

Example 9.6 The following example is used to illustrate a situation when it is impossible to find a critical variety V_C of a lower dimension than the order of the system. Consider the bilinear system which was considered in [71]:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_1(k) + x_2(k)\, u(k) \end{aligned} \tag{9.32}$$

This system is completely controllable (according to the controllability definition given in [48, 71]) but it is not dead-beat controllable. Notice that the system is in strict feedback form. Consider the state at time step 2:

$$\begin{aligned} x_1(2) &= x_1(0) + x_2(0)\, u(0) \\ x_2(2) &= x_2(0) + (x_1(0) + x_2(0)\, u(0))\, u(1) \end{aligned} \tag{9.33}$$

From the first equation we obtain that some critical states belong to the variety $V(x_2)$. Let us compute the Gröbner basis:

$$\text{Gbasis}[x_1 + x_2 u_0, x_1 + x_2 u_0] = \{x_1 + x_2 u_0\}$$

Therefore, there is no polynomial in the basis that depends only on x. We may interpret this as if the polynomial that depends on x only is identically equal to zero. That is, the critical variety is in this case $V_C = V(x_2 \cdot 0) = V(0) = \mathbb{R}^2$. The system is such that the use of the Gröbner basis method does not help us in reducing the computations although its structure is in strict feedback form. □

9.5 Conclusion

We presented several classes of polynomial systems whose structure can be used to simplify dead-beat controllability tests considerably. The interconnected systems of this chapter allow one to use QEPCAD and the Gröbner basis method more efficiently and on certain occasions even analytical controllability tests could be derived. Moreover, non-minimum and/or minimum-time dead-beat controllers can easily be built upon such controllers for some lower dimensional subsystems of the interconnected system. We presented several classes of block oriented models for which the results of this chapter can be used. We emphasize that although the interconnections that we analysed may appear to be very simple, they allow one to build more complicated systems whose dead-beat properties can easily be checked.

Chapter 10

A Simulation Study: Biochemical Reactor

10.1 Introduction

The purpose of this chapter is to motivate the class of the dead-beat controllers that were presented in the previous chapters via an example. Dead-beat control of a biochemical reactor is investigated through a simulation study. We also want to motivate some questions that appear to be very important for possible further research. It is not our intention to present a complete analysis of a specific problem but rather to substantiate some of our claims concerning the usefulness of the control laws that we obtained and the importance of some open research topics.

Recall that in Chapter 1 we already mentioned that our dead-beat controllers need to be modified in general in order to be implemented. The main modification concerns the incorporation of actuator saturation into the dead-beat control laws. This question can be resolved in principle by using QEPCAD. However, it is interesting to investigate also some simpler schemes that may provide non-minimum-time dead-beat behaviour but may be easier to implement and use.

Another important issue that was raised in the introduction is the use of the Euler discretised models of sampled nonlinear systems for the dead-beat controller design. This approach was used by some authors in the context of adaptive control [44, 69]. It is very important for our work to investigate whether our dead-beat controllers designed for the Euler discretised models yield good behaviour when applied to the sampled system.

In the simulation study that is presented below we incorporate both of these above given issues into a control scheme. First, we approximate the sampled nonlinear model of a biochemical reactor using Euler discretisation. Then, we design a minimum-time dead-beat controller for the approximate model. The approximate model belongs to Class 1 systems considered in Chapter 9. The dead-beat controller is designed using the methodology described in Chapter 9. Finally, the designed controller is modified so that saturation of actuators is incorporated into the control law. We emphasize that no analytic analysis was carried out to arrive at this control scheme. Extensive simulations indicate that a well behaved closed loop has been obtained using this method.

10.2 The Simulation Study

We start from a continuous time plant model, which describes a biochemical reactor [44]. If we assume that the growth rate of the biomass is a linear function of the substrate concentration, that the influent substrate concentration is the control signal and that the influent flow rate is constant, we obtain a state space model of the bioreactor:

$$\begin{aligned} \dot{x}_1 &= \theta_1 x_1 x_2 - \theta_3 x_1 \\ \dot{x}_2 &= -\theta_2 x_1 x_2 - \theta_3 x_2 + u \end{aligned} \tag{10.1}$$

Here x_1 and x_2 are respectively the biomass concentration and the substrate concentration in the reactor. The constants θ_i, i=1, 2, 3 depend on yield coefficients and the influent flow rate and in our case they are θ_1=0.4, θ_2=0.16, θ_3=0.1. The influent substrate concentration u is a control signal and $u \in [0,\ 0.5]$. It is a physical constraint that all initial states are positive. It is not difficult to see that since the control is always positive, the states are also positive $\forall t \geq 0$.

Suppose that a digital control law is called for. The states in the continuous time model (10.1) are sampled using a sampler and zero order hold.

The sampled (discrete-time) model for system (10.1) typically leads to an infinite dimensional state space representation. However, if we restrict ourselves to a simple approximation scheme, using an Euler approximation, we obtain:

$$\begin{aligned} x_1(k+1) &= x_1(k) + T(\theta_1 x_1(k)\, x_2(k) - \theta_3 x_1(k)) \\ x_2(k+1) &= x_2(k) + T(-\theta_2 x_1(k)\, x_2(k) - \theta_3 x_2(k) + u(k)) \end{aligned} \tag{10.2}$$

where T is the sampling period. Notice that T is a design parameter. Let T=1. The approximate discrete-time model of (10.1) is very easy to obtain from the original model. We are interested in whether a dead-beat controller constructed for the approximate model provides an adequate control behaviour when applied to the continuous time system.

In the case of the biochemical reactor we do obtain for the Euler approximation a triangular system (9.2). Using Theorem 9.1, we conclude that the Euler discretised model of the systems is dead-beat controllable with $u \in \mathbb{R}$.

The system is Lipshitz since the right hand side of (10.1) is differentiable and hence the set of equilibria is given by $\{(x_1, x_2) : x_1(\theta_1 x_2 - \theta_3) = 0\}$ and $u \in \mathbb{R}$. We design a dead-beat controller which is designed to transfer any (allowed) initial state to the equilibrium $(x_1,\ x_2) = (4.37,\ 0.25)$ in minimum time. The stationary value of the control signal for the given equilibrium is $u^* = 0.2$. Assume that the state $x(k)$ is measured at time steps k. A minimum-time dead-beat controller for the approximate model with $\hat{u} \in \mathbb{R}$ is given by:

$$\hat{u}(k) = \begin{cases} 0.25 - x_2(k)\,(0.9 + 0.4x_2(k)\,) & \text{, if } 4.37 - x_1(k)\,(0.9 + 0.4x_2(k)\,) = 0 \\ 2.5(\frac{4.37}{x_1(k)(0.9+0.4x_2(k))} & \text{, otherwise} \\ -0.9 - 0.4x_2(k)\,(0.9 - 0.16x_1(k)\,)\,) & \end{cases} \tag{10.3}$$

The minimum-time dead-beat controller is very simple and easy to implement. It is static discontinuous state feedback controller. Because of the physical limitations, we also truncate the computed dead-beat controls to lie within the allowable range:

$$u(k) = \begin{cases} 0, & \text{if } \hat{u}(k) \\ \hat{u}(k)\,, & \text{if } 0 \leq \hat{u}(k) \leq u_{max} \\ u_{max}, & \text{if } \hat{u}(k) \geq u_{max} \end{cases} \tag{10.4}$$

where the maximum value u_{max}=0.5. We investigated via simulations the behaviour of the closed loop system where the modified controller for the Euler approximation (10.4) is applied to the sampled continuous time plant.

From simulations (Figure 10.1) we can conclude that a very good behaved closed loop system is obtained. The simulations reveal that a kind of quasi dead-beat behaviour is obtained. In the neighbourhood of the equilibrium $(4.37, 0.25)$ all the trajectories converge very rapidly (faster than a quadratic convergence) to the equilibrium. In Figure 10.1, however, this looks as a dead-beat behaviour. The rate of convergence is very interesting and it would be very important if we could

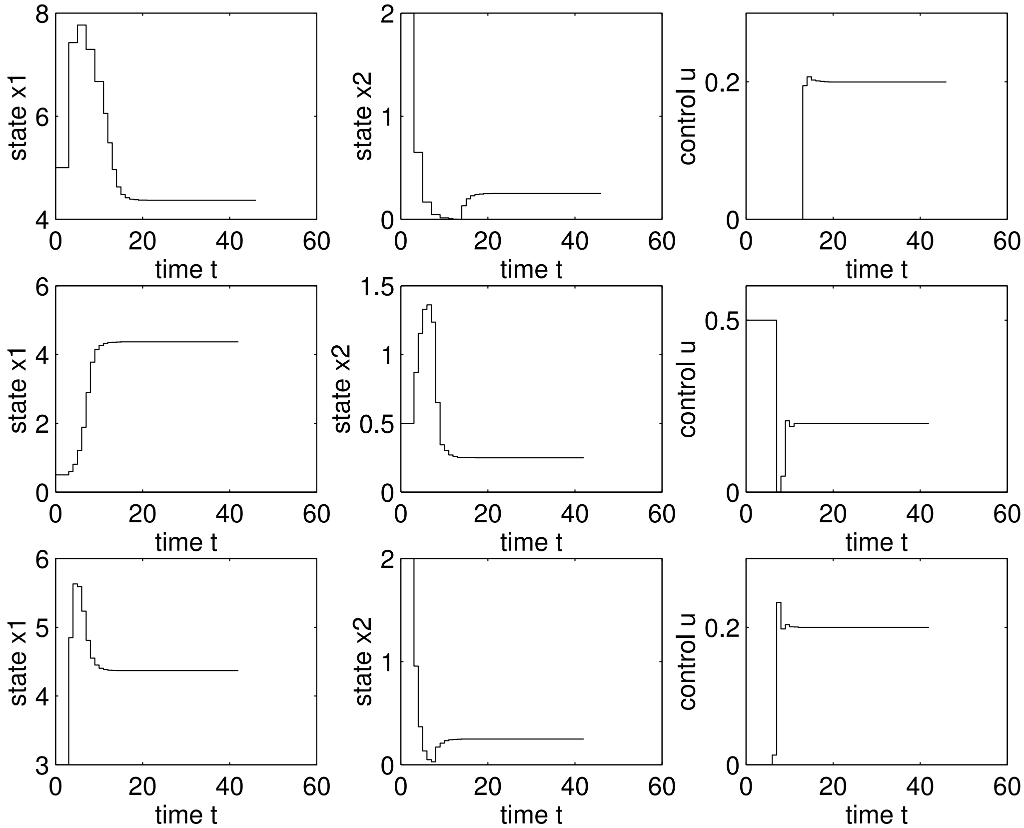

Figure 10.1: Simulation results for the biochemical reactor

prove that using this scheme we can achieve similar results for classes of nonlinear systems.

The rule of thumb that is used to design this controller yields very good results. It is then apparent that for classes of nonlinear systems it might be possible to implement dead-beat controllers designed on the basis of an Euler approximation of the sampled model in order to obtain good control strategies for the sampled non-truncated system. We believe that this question is very important from an applications point of view and should be pursued in future research.

10.3 Conclusion

The simulation study of the biochemical reactor indicates that it is possible to modify controllers that are obtained in this thesis in order to obtain good control strategies. More importantly, it seems that there exists a motivation for considering minimum-time dead-beat controllers for the Euler approximate models of a sampled nonlinear systems since their implementation may yield satisfactory results. We believe that this study motivates the work in this thesis and in particular the controllers that we propose.

Part II

Minimum Phase Polynomial Systems and Stable Zero Dynamics

Chapter 11

Minimum Phase Polynomial Systems

11.1 Introduction

The notions of zero output constrained dynamics and minimum phase systems play an important role in output dead-beat control [66, 15], I-O linearisation of nonlinear systems [123, 124] and a number of related problems [130, 86].

This chapter consists of three parts. In the first part, we illustrate through examples the phenomenon of non-uniqueness of I-O linearising control laws. This provides us with motivation for introducing definitions of minimum phase systems that deal with both local and global aspects. In the second part of the chapter, we show how QEPCAD can be used to test these minimum phase properties for polynomial systems. Finally, in the last part of this chapter we derive more explicit conditions for stability of one dimensional implicit zero dynamics, which can be checked without resorting to QEPCAD. We emphasize that QEPCAD based stability tests are applicable to much larger classes of explicit and implicit polynomial systems than the ones considered here. In this sense, the results that we present are by no means comprehensive or the most general but are rather an illustration of how QEPCAD may be used for related problems.

Several papers have dealt with the minimum phase property of nonlinear discrete-time systems [123, 124]. The basic ingredients in arriving at zero dynamics are the concept of relative degree and I-O linearisation through a state space coordinate transformation and a feedback transformation. These notions are normally introduced in some neighbourhood of a point of interest in state space $\times$ input space ($X \times U$). The main analytical tool used in the construction of the appropriate transformation is the implicit function theorem.

This approach may introduce unnecessary limitations in control design, say for stability purpose. Indeed, we are normally interested in a point in state space X, not necessarily in a point in state $\times$ input space ($X \times U$). Allowing ourselves the freedom to work in a not necessarily small set of the input space may offer the potential of better control action. This point will be illustrated with some examples. Furthermore, even when a point in $X \times U$ space is of interest, but when the conditions of the implicit function theorem are not satisfied, it remains possible to consider an I-O linearising control action. Of course, we will need to contend with the possibility of choosing from amongst many alternative and co-existing control laws. This point is also illustrated with an example.

In other words, we may have a choice over several different control laws which bring the nonlinear systems into the canonical form which is used to define zero dynamics. Each of the control laws I-O linearises the system while yielding different equations for zero dynamics. The importance of the choice of an appropriate control law was noted already in [15] in the context of output dead-beat control of recursive nonlinear systems. These results, however, did not make any connection with the results [123, 124] and the two approaches seemed unrelated. We attempt below to provide a unified theory and more importantly we propose tests which can be used to check different minimum phase properties of polynomial systems. Recently, the notion of bijective relative degree has been introduced by [159]. This notion corresponds to situations when there is a unique control law which input-output linearises the system and hence zero dynamics are uniquely defined. This notion is too restrictive for the implicit polynomial zero dynamics considered in this chapter.

11.2 Motivation

In this section, we present some results from the literature. We point out the phenomenon of non-uniqueness of I-O linearising control laws which may give rise to zero dynamics with different stability properties. Two examples serve as motivation for the definitions of minimum phase systems that are introduced in the next section.

Consider the systems of the form [123]:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= h(x(k)) \end{aligned} \tag{11.1}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}$ and $y(k) \in \mathbb{R}$ are respectively the state, the input and the output of the system at time k. The functions f and h are assumed to be analytic [123]. Denote $f(x,0) = f_0$.

Definition 11.1 *[123]* The relative degree r of the system (11.1) is an integer with the property[1]:

$$\begin{aligned} \frac{\partial h \circ f_0^i \circ f(x,u)}{\partial u} &\equiv 0, i=0,1,\ldots,r-1 \\ \frac{\partial h \circ f_0^r \circ f(x,u)}{\partial u} &\neq 0 \text{ almost everywhere in } \mathbb{R}^{n+1} \end{aligned} \tag{11.2}$$

□

Assume that there exists an equilibrium x^* and a control u^* such that

$$f(x^*,u^*) = x^*, \; h(x^*) = 0. \tag{11.3}$$

According to [123, 124] if the following conditions are satisfied:

C1 The system has relative degree $r \leq n$.

C2 $\frac{\partial (h \circ f_0^r)(x,u)}{\partial u}|_{(x^*,u^*)} \neq 0$.

C3 $0 \in \text{im } (h \circ f^i)(x,\cdot), \; \forall x$

then there exists a non-singular transformation of coordinates $z = Z(x)$ and a non-singular state feedback $u(k) = U(x(k), v(k))$ which I-O linearise the system around the equilibrium (x^*, u^*):

$$\begin{aligned} z_1(k+1) &= Az_1(k) + bv(k) \\ z_2(k+1) &= F(z_1(k), z_2(k), v(k)) \\ y(k) &= cz_1(k) \end{aligned} \tag{11.4}$$

The (explicit) zero dynamics are then defined as ($z_1 \equiv 0, \; v \equiv 0$)

$$z_2(k+1) = F(0, z_2(k), 0) \tag{11.5}$$

and the system (11.1) is defined to be minimum phase if its zero dynamics (11.5) are stable [123, 124].

[1]There are slight variations of the definition of relative degree in the literature, see for instance [130, 159]. However, none of them seems to be general enough to deal with the problems presented in this section.

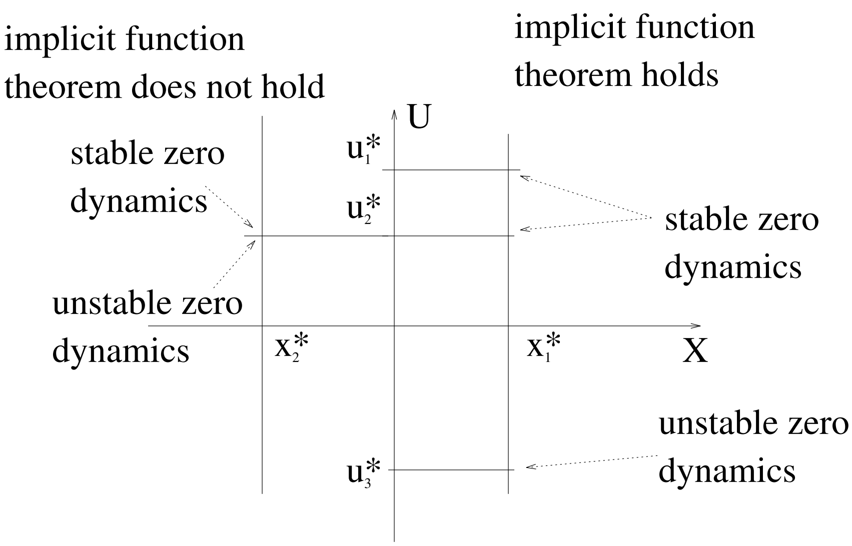

Figure 11.1: Non-uniqueness of I-O linearising control laws

We emphasize that in order to use the implicit function theorem as in [123, 124] we have to keep track of the equilibrium point, given by (11.3), around which we are working. In general, however, the pairs of controls and states, which satisfy the above equations, may be non-unique, that is, we may have (x_i^*, u_i^*), $i{=}1, \ldots, p$ such that (11.3) holds. If in addition we have that the condition **C2** is satisfied for all (x_i^*, u_i^*), we can use the implicit function theorem for any of the points (x_i^*, u_i^*). Hence, we may have non-uniqueness of control laws that locally I-O linearise the system.

More importantly, it may happen that for a single state x_j^* we may have several controls $u_{k_j}^*, k_j \in \{1, \ldots, p\}$ that satisfy the equation (11.3) and the condition **C2**. In other words, for a single x_j^* we may have several continuous control laws that I-O linearise the system around the equilibrium x_j^*. In general, these control laws give rise to *different* equations that define zero dynamics (11.5). Some of them may yield stable zero dynamics and some may yield unstable zero dynamics. This situation is depicted in Figure 11.1.

Finally, if the relative degree of the system is r but for some (or all) of the points (x_i^*, u_i^*) we have

$$\frac{\partial (h \circ f^r)(x,u)}{\partial u}\Big|_{(x_i^*, u_i^*)}{=}0$$

it is not possible to use the methodology of [123, 124].

We conclude that the stability of zero dynamics depends in general not only on the equilibrium (x^*, u^*) but also on the control law that we are going to use. We observe that in such circumstances it is no longer clear what is meant by "the system is minimum phase". The obvious control choice that we may have needs to be reflected into the definition of the minimum phase property.

The following examples serve to illustrate these observations more concretely.

Example 11.1 Consider the system:

$$\begin{aligned} x_1(k+1) &= 0.7x_2(k) + u(k) + u^2(k) \\ x_2(k+1) &= x_1(k) + u^3(k) - u(k) \\ y(k) &= x_1(k) \end{aligned} \tag{11.6}$$

Suppose that the equilibrium $x^*{=}(0\ 0)^T$ is the point around which we wish to I-O linearise the system. Then, the controls $u_1^*{=}0$ and $u_2^*{=}-1$ are both good candidates for the application of the implicit function theorem. Hence, there are two equilibria in $X \times U$ around which it is possible to I-O linearise the system: $(x^*, 0)$ and $(x^*, -1)$. Suppose that both controls are well within the actuator range.

It is not difficult to check that all conditions of [123, 124] are satisfied (for both equilibria) and hence there are two *different* control laws that I-O linearise the system around the origin $x^*{=}0$.

The control laws:

$$u(k) = -0.5 \pm \sqrt{0.25 - 0.7x_2(k) + v(k)}$$

both I-O linearise the system, where $v(k)$ is the new external control input. The system is then transformed into

$$\begin{aligned} x_1(k+1) &= v(k) \\ x_2(k+1) &= x_1(k) + (-0.5 \pm \sqrt{0.25 - 0.7x_2(k) + v(k)})^3 \\ &\quad -(-0.5 \pm \sqrt{0.25 - 0.7x_2(k) + v(k)}) \\ y(k) &= x_1(k) \end{aligned} \tag{11.7}$$

Zero dynamics are defined for $v(k){=}0$ and $x_1(k){=}0$ and they are given by:

$$x_2(k+1) = (-0.5 \pm \sqrt{0.25 - 0.7x_2(k)})^3 - (-0.5 \pm \sqrt{0.25 - 0.7x_2(k)}) \tag{11.8}$$

The zero dynamics are well defined on the neighbourhood of the origin $x_2 \leq 0.3571$.

Suppose that we choose the control law $u(k) = -0.5 - \sqrt{0.25 - 0.7x_2(k) + v(k)}$ (in this case we work around the point $(x^*, -1)$). The corresponding zero dynamics are given by

$$x_2(k+1) = (-0.5 - \sqrt{0.25 - 0.7x_2(k)})^3 - (-0.5 - \sqrt{0.25 - 0.7x_2(k)})$$

It is easy to show that these zero dynamics are unstable and we conclude that the system is not minimum phase.

If we now use the control law $u(k) = -0.5 + \sqrt{0.25 - 0.7x_2(k) + v(k)}$ (we work around the point $(x^*, 0)$), the zero dynamics become:

$$x_2(k+1) = (-0.5 + \sqrt{0.25 - 0.7x_2(k)})^3 - (-0.5 + \sqrt{0.25 - 0.7x_2(k)})$$

These zero dynamics are stable.

We emphasize that this obvious choice that we have at our disposal is not clearly incorporated into the definition of minimum phase systems found in the literature. Moreover, it does not seem to be natural to work around points (x^*, u^*) since we are usually interested in x^* provided u^* is in the actuator range. This is a drawback of the definition of minimum phase systems which relies on the method of the implicit function theorem. Indeed, the definition should be applicable to any nonlinear system irrespective of the method that we use to analyse the minimum phase property.

□

Example 11.2 If the condition **C2** does not hold for a system, we can no longer appeal to the implicit function theorem arguments to define/construct zero dynamics. Yet, such a notion may still have a meaning. Consider the following system:

$$\begin{aligned} x_1(k+1) &= (u(k) + x_1(k) + 0.5x_2(k))(u(k) + x_1(k) - 3x_2(k)) \\ x_2(k+1) &= u(k) \\ y(k) &= x_1(k) \end{aligned} \tag{11.9}$$

Notice that the point $(x^*, u^*) = (0, 0)$ does not satisfy the implicit function theorem condition given by **C2**. The system, however, has relative degree $r = 1$ in the sense of [123, 124]. Observe that the partial derivative in the condition **C2** vanishes at the particular point (x^*, u^*). This situation has not been analysed in the literature to the best of our knowledge.

We show below that an I-O linearising feedback law still exists. The systems is already in the form which does not require a change of coordinates and only the state feedback is needed to linearise the first state equation. Consider the following continuous feedback laws:

$$\begin{aligned} u(k) &= -x_1(k) + 1.25x_2(k) + [0.25(x_1(k) - 2.5x_2(k))^2 \\ &\quad -(x_1(k) + 0.5x_2(k))(x_1(k) - 3x_2(k)) - v(k)]^{0.5} \\ u(k) &= -x_1(k) + 1.25x_2(k) - [0.25(2x_1(k) - 2.5x_2(k))^2 \\ &\quad -(x_1(k) + 0.5x_2(k))(x_1(k) - 3x_2(k)) - v(k)]^{0.5} \end{aligned}$$

where $v(k)$ is a new external control variable. If we apply the above given control laws to the system (11.9), it is transformed into:

$$\begin{aligned} x_1(k+1) &= v(k) \\ x_2(k+1) &= -x_1(k) + 1.25x_2(k) + [0.25(x_1(k) - 2.5x_2(k))^2 \\ &\quad -(x_1(k) + 0.5x_2(k))(x_1(k) - 3x_2(k)) - v(k)]^{0.5} \\ y(k) &= x_1(k) \end{aligned} \tag{11.10}$$

or

$$\begin{aligned} x_1(k+1) &= v(k) \\ x_2(k+1) &= -x_1(k) + 1.25x_2(k) - [0.25(2x_1(k) - 2.5x_2(k))^2 \\ &\quad -(x_1(k) + 0.5x_2(k))(x_1(k) - 3x_2(k)) - v(k)]^{0.5} \\ y(k) &= x_1(k) \end{aligned} \tag{11.11}$$

Both (11.10) and (11.11) are clearly I-O linear. The corresponding zero dynamics for $v(k) = 0, x_1(k) = 0, \forall k$ are characterised respectively by:

$$x_2(k+1) = -0.5x_2(k) \text{ and } x_2(k+1) = 3x_2(k)$$

In the first case, the zero dynamics are stable and the second situation they are not. Hence, it is clear that the choice of the feedback law $u(k) = g(x(k), v(k))$ must be taken into account when considering the minimum phase property.

In the previous example we considered I-O linearisation around points $(0, -1)$ and $(0, 0)$ in $X \times U$ space. Here, however, we have that both above I-O linearising control laws linearise the system around the point $(0, 0)$ in $X \times U$ space. This is is due to the fact that the partial derivative in the condition **C2** vanishes at the particular point around which we are linearising the system. This situation is shown in Figure 11.1. □

The given examples show that the known definition of minimum phase systems found in the literature relies heavily on the use of the implicit function theorem. Moreover, in many situations the definition may be inadequate to capture what is going on. It therefore seems appropriate to try to redefine the concept of the minimum phase system property to be able to overcome the above illustrated difficulties, while maintaining compatibility with the definitions that apply in the more restrictive situations where the implicit function theorem tells the complete story. This is attempted in the next section.

11.3 A QEPCAD Based Minimum Phase Tests

In this section we consider stability of zero dynamics for classes of I-O polynomial systems (see Appendix B). In this context the phenomena illustrated by the examples appear in a very natural way, hence our preoccupation with this class of systems.

The class of I-O polynomial systems normally gives rise to implicitly defined zero dynamics. The criterion of choice plays a central role in stability properties of these equations [15]. In [15], necessary and sufficient conditions are given for stability of one dimensional explicit zero dynamics. For a number of sufficient results on stability of higher order explicit zero dynamics (with positive states) see [105]. Although the classes of systems considered in [15, 105] are not the same as the one considered here, some results are closely related. The main difference between our work in this section and the existing results on the topic is that we do not aim at obtaining explicit conditions for stability but rather a method with which we can test stability.

We propose a new method of testing whether there exists a criterion of choice which yields stable zero dynamics for a class of the implicitly defined zero dynamics (Chapter 3 and Appendix B). Recall that in Chapter 3 we showed how QEPCAD can be used in deciding dead-beat controllability of polynomial systems.

11.3.1 Preliminaries

We use the following notation. Euclidean norm of a vector $x \in \mathbb{R}^p$ is denoted as $||x||$. The distance between points $x, y \in \mathbb{R}^p$ and the distance between a set $A \subset \mathbb{R}^p$ and a point $x \in \mathbb{R}^p$ are respectively denoted as $d(x, y) = ||x - y||$ and $d(x, A) = \inf_{y \in A} ||x - y||$. The hyper-ball centred at a point z_2^* with a radius $p > 0$ is denoted as $\mathcal{B}_p(z_2^*) = \{z_2 \in \mathbb{R}^t : d(z_2, z_2^*) < p\}$ and the hyper-cube centred at a point z_2^* with sides $2r > 0$ is denoted as

$$\mathcal{C}_r(z_2^*) = \{z_2 \in \mathbb{R}^t : (|z_{2,1} - \zeta| < r) \wedge (|z_{2,2} - \zeta| < r) \wedge \ldots \wedge (|z_{2,t} - \zeta| < r)\}$$

where $|a|$ is the absolute value of the scalar a.

Polynomial I-O systems [76, 21, 107] typically have non-unique I-O linearising control laws. We consider a class of polynomial I-O models of the form [76]:

$$y(k+1) = F(y(k), y(k-1), \ldots, y(k-s), u(k-t), u(k-t+1), \ldots, u(k)) \tag{11.12}$$

where $y(k) \in \mathbb{R}$ and $u(k) \in \mathbb{R}$ are respectively output and input of the system at time instant k. F is a polynomial function in all its arguments. Notice that if we want to control the output of the system (11.12) to a desired value y^* and keep it for all future time steps at y^*, the system evolves according to:

$$F(y^*, y^*, \ldots, y^*, u(k-t), u(k-t+1), \ldots, u(k)) - y^* = 0 \tag{11.13}$$

The equation (11.13) defines the final regime in output dead-beat control, which we investigate here. We assume that the system is output dead-beat controllable and that after finitely many steps the system evolves according to (11.13). Also, it is assumed that y^* is known and constant. The output value y^* represents a bifurcation parameter for the equation (11.13) and different values of y^* give rise to different equations (11.13). Denote $F(y^*, \ldots, y^*, u(k-t), u(k-t+1), \ldots, u(k)) - y^* \triangleq G(u(k-t), u(k-t+1), \ldots, u(k))$. Henceforth we consider the equation:

$$G(u(k-t), u(k-t+1), \ldots, u(k)) = 0 \tag{11.14}$$

We say that (11.14) defines implicit zero dynamics. Explicit zero dynamics take the form

$$u(k) = G_e(u(k-t), \ldots, u(k-1)) \tag{11.15}$$

If we introduce state variables $u(k-t) = z_{2,1}(k)$, $u(k-t+1) = z_{2,2}(k), \ldots, u(k-1) = z_{2,t}(k)$[2], we obtain the linear system:

$$\begin{aligned} z_{2,1}(k+1) &= z_{2,2}(k) \\ z_{2,2}(k+1) &= z_{2,3}(k) \\ \ldots & \quad \ldots \\ z_{2,t}(k+1) &= u(k) \end{aligned} \tag{11.16}$$

which is constrained on the real variety in $\mathbb{R}^t$ defined by

$$G(z_{2,1}(k), z_{2,2}(k), \ldots, z_{2,t}(k), u(k)) = 0 \tag{11.17}$$

We use a shorter notation $z_2(k) = (z_{2,1}(k) \;\; z_{2,2}(k) \;\; \ldots \;\; z_{2,t}(k))^T$.

Assumption 11.1 $\forall z_2 \in \mathbb{R}^t$, $\exists u \in \mathbb{R}$ such that $G(z_2, u) = 0$. □

The equilibria of the system (11.14) are found using $G(\zeta, \zeta, \ldots, \zeta) = 0$. We denote the equilibria as $z_2^* = (\zeta \; \zeta \; \ldots \zeta)^T \in \mathbb{R}^t$.

Notice that for any initial state $z_2(0)$ we can apply to the linear system (11.16) only controls $u(k) = u(z_2(k))$ which are obtained as solutions of the equation $G(z_2(k), u(k)) = 0$. Since $G(z_2, u)$ is a polynomial in u and z_2, for almost all z_2 we will have finitely many roots u (Note non-uniqueness of roots!).

Our objective is to keep the control $u(k)$ in (11.14) from growing unbounded. The choice we have at our disposal provides us with some design flexibility when dealing with this problem. Suppose that we have a rule according to which for any z_2 we chose a (unique) control u satisfying the constraint (11.17). Then, one can introduce "explicit" dynamics that satisfy the constraint (11.17) and which follow from the chosen rule. We can analyse stability properties of the explicit dynamics and if we find that they are stable, it follows that the rule we have chosen yields a desirable behaviour. If this is not the case we may have a possibility to chose a different rule whose associated explicit dynamics may be stable. Therefore, the constrained stabilisability problem for the system (11.16) with the constraint (11.17) can be reformulated into a stability problem for a number (or infinitely many) *explicit* dynamics. This argument motivates the definitions:

[2] This notation is used to emphasize the relation to the equation (11.5)

Definition 11.2 A criterion of choice is a single valued function $c : \mathbb{R}^t \to \mathbb{R}$ (denoted also as $u(k) = c(u(k-t), \ldots, u(k-1))$) such that

$$G(u(k-t), \ldots, u(k-1), c(u(k-t), \ldots, u(k-1))) = 0, \ \forall u(k-1), \ldots, u(k-t) \in \mathbb{R}. \quad (11.18)$$

□

Definition 11.3 Consider a criterion of choice applied to the system (11.16):

$$\begin{aligned} z_{2,1}(k+1) &= z_{2,2}(k) \\ z_{2,2}(k+1) &= z_{2,3}(k) \\ \ldots & \quad \ldots \\ z_{2,t}(k+1) &= c(z_{2,1}(k), \ldots, z_{2,t}(k)) \end{aligned} \quad (11.19)$$

We call the system (11.19) the "c"-resulting system. □

Observe that different "c"-resulting systems represent the explicit dynamics associated with a chosen rule (criterion of choice) which satisfy the constraint (11.17).

Definition 11.4 The equilibrium z_2^* of the "c"-resulting system (11.19) is:

1. stable if $\forall \varepsilon > 0, \exists \delta > 0, \delta = \delta(\varepsilon)$ such that if $z_2(0) \in \mathcal{B}_\delta(z_2^*)$ then
   $z_2(k, z_2(0)) \in \mathcal{B}_\varepsilon(z_2^*)$,
   $\forall k = 1, 2, \ldots$.

2. attractive if $\exists \Delta > 0$ such that if $z_2(0) \in \mathcal{B}_\Delta(z_2^*)$, then $\lim_{k \to \infty} ||z_2(k, z_2(0)) - z_2^*|| = 0$

3. locally asymptotically stable if 1 and 2 hold

4. globally asymptotically stable if 1 holds and 2 is satisfied $\forall z_2(0) \in \mathbb{R}^t$. □

The above given definitions are in the spirit of a local problem formulation of [123, 124]. However, this is not the only definition of stability of "c"-resulting systems (zero dynamics) found in the literature. For instance, in [15] stability of an invariant set (attractor) was proposed as an alternative. It is our opinion that both of these notions are very important . Point stability of [123, 124] is tailored for the local analysis around a desired operating point, whereas set stability seems to be more appropriate for global analysis of nonlinear systems (it can not occur in linear framework, where point stability is sufficient for global investigations).

It seems that the two definitions are just opposite ends of a large spectrum of different "practical stability" notions that may be introduced and which may be more important for a particular application. Hence, a number of other stability properties may be introduced. Moreover, a nonlinear system may not have point stable zero dynamics and yet a good internal behaviour may be possible while the output is kept constant. This fact comes from the nonlinear structure of the problems that we consider and can not happen in the linear framework. Hence, set stability (attractor) is a nonlinear phenomenon which sometimes may be successfully exploited in the design.

In the sequel we use both of the above mentioned formulations and we define:

Definition 11.5 Consider a criterion of choice c and the "c"-resulting system (see Definition 11.3). A bounded set A ($\sup_{x,y\in A} d(x,y) < \infty$) is invariant if $\forall z_2(0) \in A$ we have that $z_2(k, z_2(0)) \in A, \forall k$. □

We emphasize that this definition of invariance is different than the one introduced in the previous chapters when investigating output dead-beat controllability. Also, we note that an invariant set A is not unique in general and we usually do not work with the smallest such set but rather with the ones which are simple to analyse.

Definition 11.6 An invariant set $A \neq \emptyset$ of a "c"-resulting system (see Definition 11.3) is

1. stable if $\forall \varepsilon > 0, \exists \delta > 0, \delta = \delta(\varepsilon)$ such that if $d(z_2(0), A) < \delta$ it follows that

$$d(z_2(k, z_2(0)), A) < \varepsilon, \forall k$$

2. attractive if $\exists \Delta > 0$ such that if $d(z_2(0), A) < \Delta$ it follows that

$$\lim_{k\to\infty} d(z_2(k, z_2(0)), A) = 0$$

3. asymptotically stable if 1 and 2 hold

4. globally asymptotically stable if 1 holds and 2 holds with $\Delta = \infty$. □

Below we propose definitions of minimum phase systems, which recognise the existence of the criterion of choice notion.

Definition 11.7 The system (11.12) is:

1. point-minimum phase (set-minimum phase) if *there exists* a criterion of choice c such that the equilibrium z_2^* (bounded invariant set A) of the "c"-resulting system is asymptotically stable

2. uniformly point-minimum phase (uniformly set-minimum phase) if *for any* criterion of choice c the equilibrium z_2^* (bounded invariant set A) of the "c"-resulting system is asymptotically stable

3. non-minimum phase if it is neither point nor set-minimum phase □

It is immediately clear that if the I-O linearising control law is unique, minimum phase and uniform minimum phase notions coincide. This corresponds to the situation when the system has a bijective relative degree [159].

To illustrate the introduced definitions we revisit Example 11.2.

Example 11.3 Notice that the I-O description of the system (11.9) is given by:

$$y(k+1) = (u(k) + 0.5u(k-1) + y(k))(u(k) - 3u(k-1) + y(k)) \tag{11.20}$$

Assume that an output dead-beat controller is applied to the system. When the output is equal to zero the system evolves according to:

$$0 = (u(k) + 0.5u(k-1))(u(k) - 3u(k-1)) \tag{11.21}$$

which is obviously implicit dynamics.

We can design an infinity of different recursion control laws that satisfy (11.21). In other words there are infinitely many criteria of choice which yield different point and/or set stability properties of the "c"-resulting systems. Consider the variety $V_z = V((u+0.5v)(u-3v)) \subset \mathbb{R}^2$. It consists of two lines intersecting at the origin. It is easy to see that for any fixed value $v \in \mathbb{R}$ there are two values of $u \in V_z$. Hence, for any value of v we can choose either $u = -0.5v$ or $u = 3v$. Therefore, we can construct infinitely many control laws $u(k) = c(u(k-1))$ satisfying (11.21) and which are called criteria of choice.

Some such control laws are given below:

$$
\begin{array}{lll}
1. & u(k) = & \begin{cases} -0.5u(k-1) & \text{if } u(k-1) \geq 0 \\ 3u(k-1) & \text{if } u(k-1) < 0 \end{cases} \\
2. & u(k) = & -0.5u(k-1)\,,\ \forall u(k-1) \in \mathbb{R} \\
3. & u(k) = & \begin{cases} -0.5u(k-1) & \text{if } |u(k-1)| \geq 10 \\ 3u(k-1) & \text{if } |u(k-1)| < 10 \end{cases} \\
4. & u(k) = & 3u(k-1)\,,\ \forall u(k-1) \in \mathbb{R}
\end{array}
$$

These control laws exhibit very different behaviours. In Case 1 for any $u(-1) \in \mathbb{R} - \{0\}$, the control sequence exhibits an unstable periodic behaviour. Case 2 represents a more acceptable choice since the value of the control signal asymptotically approaches 0 as $k \to \infty$, $\forall u(-1) \in \mathbb{R}$. Case 3 gives different behaviour again. There exists an invariant interval $[-30, +30]$ which is reached $\forall u(-1) \in \mathbb{R} - [-30, +30]$ in finite number of time steps $N(u(-1))$ and then $|u(k)| \leq 30$, $\forall k > N$. In general the size of the invariant interval obviously determines whether the control algorithm is feasible or not. Case 4 yields a control law which should not be implemented since $|u(k)| \to +\infty$ as $k \to \infty$, $\forall u(-1) \in \mathbb{R} - \{0\}$.

In all four cases, the control law which assigns to each $u(k-1) \in \mathbb{R}$ a single $u(k)$ which satisfies (11.21) is called a criterion of choice. Case 2 is an example of a criterion of choice which yields point-minimum phaseness and Case 3 is a criterion of choice which shows the set-minimum phase property. So, stability properties of "c"-resulting systems (zero dynamics) depend on the criterion of choice c that we are using. Cases 1 and 4 illustrate criteria of choice that yield zero dynamics which are not stable. □

Comment 11.1 The above given definitions of minimum phase systems may be generalised. First, one may rephrase the definitions to include a possible non-uniqueness of the equilibria z_2^* and define minimum phase system: "there exists an equilibrium z_2^* and a criterion of choice c which renders the equilibrium asymptotically stable". We note here that the methods that we propose in the next section to check different minimum phase properties of polynomial systems can be used (with modifications) to check more general notions.

We point out that when we talk about the point-minimum phase property, we assume that the equilibrium $z^* \in X$ of interest is known *a priori*. On the other hand, for set-minimum phase properties we normally show how to find an invariant set A. This reflects that the point-minimum phase property is used in a local analysis and the set-minimum phase property is considered when dealing with global behaviour. □

11.3.2 Main Results

Point-minimum phase test

We present below several tests for point-minimum phase properties of I-O polynomial systems. We do not present the most general approach to this problem but rather illustrate a methodology which can be used for this problem. However, we often comment on other formulations and generalisations. For I-O polynomial systems we propose the use of QEPCAD symbolic computation package [33, 34, 35] to check point-minimum phaseness. Computational complexity of the problems may be prohibitive and this is the main hindrance to the proposed method. Nevertheless, for I-O systems of small multi-degrees of the defining polynomial map, the method may yield satisfactory results. Without loss of generality we assume in this subsection that the equilibrium around which we are working is the origin z_2^*=0.

Fix a number $U < 1$ such that $|1 - U| \ll 1$, for instance U=0.9999. In the sequel we exploit the following sets

$$\begin{aligned} S_1^{z_{2,j}} &= \{z_2 \in \mathbb{R}^t : \exists u(0) \in \mathbb{R}, |u(0)| < U|z_{2,j}|, G(z_{2,1}, \ldots, z_{2,t}, u(0)) = 0\} \\ S_2^{z_{2,j}} &= \{z_2 \in \mathbb{R}^t : \exists u(0), u(1) \in \mathbb{R}, |u(1)| < U|z_{2,j}|, G(z_{2,1}, \ldots, z_{2,t}, u(0)) = 0, \\ & \quad G(z_{2,2}, \ldots, z_{2,t}, u(0), u(1)) = 0\} \\ \ldots & \quad \ldots \end{aligned} \tag{11.22}$$

where j=$1, 2, \ldots, t$. Hence, sets $S_k^{z_{2,j}}$ represent states in $\mathbb{R}^t$ for which there is a sequence of controls (criterion of choice) yielding $|z_{2,t}(k)| < U|z_{2,j}(0)|$. The above given sets can be used to check minimum phase properties of the system (11.12). Notice that the sets can be computed using QEPCAD in the familiar way (see Chapter 3) since the inequality $|u(0)| < U|z_{2,j}|$ can be rewritten as four inequalities without absolute values. We, however, use absolute values to shorten notation. These sets are semi-algebraic. We use the notation $S_1^{z_{2,j}}(z_2)$ to denote the expression which defines the set $S_1^{z_{2,j}}$.

The expression $S_1^{z_{2,j}}(z_2)$ can be computed by considering the QE problem

$$(\exists u(0))\,[|u(0)| < U|z_{2,j}| \wedge G(z_2, u(0)) = 0]\,.$$

The above defined sets can be given in certain cases a nice interpretation based on Lyapunov functions. Indeed, assume that the set $S_1^{z_{2,1}}$ is a neighbourhood of the origin. Assume that we consider the explicit zero dynamics (11.15) with the function G continuous and define the Lyapunov function:

$$V(z_2(k)) = \sum_{i=1}^{t} |z_{2,i}(k)|$$

which is positive definite. By considering the difference:

$$V(z_2(k+1)) - V(z_2(k)) = \sum_{i=2}^{t} |z_{2,i}(k)| + |G_e(z_2(k))| - \sum_{i=1}^{t} |z_{2,i}(k)|$$

we obtain $V(z_2(k+1)) - V(z_2(k)) = |G_e(z_2(k))| - |z_{2,1}(k)|$. By definition of the set $S_1^{z_{2,1}}$ we have that $|G(z_2(k))| < |z_{2,1}(k)|$ on the set. Hence, we obtain

$$V(z_2(k+1)) - V(z_2(k)) < 0, \forall z_2 \in S_1^{z_{2,1}}$$

and since $S_1^{z_{2,1}}$ is a neighbourhood of the origin, the origin of the "*c*"-resulting system is asymptotically stable. Notice that in this case we could use the quadratic Lyapunov function

$$V(z_2(k)) = \sum_{i=1}^{t} z_{2,i}^2(k)$$

to arrive at the same conclusion.

This result can be generalised to implicitly defined zero dynamics (11.14) and even when the criterion of choice is a discontinuous mapping. We show below that the sets can be used to prove stability properties without having to resort to Lyapunov functions, namely *by definition*. We explain below in more detail what is meant by this.

Theorem 11.1 *There exists a criterion of choice c such that the origin of the "c"-resulting system is stable if the set $N = \cup_j S_1^{z_{2,j}}$ is a neighbourhood of the origin.* □

Proof of Theorem 11.1: Notice first that if the set N contains a neighbourhood of the origin this guarantees that Assumption 11.1 is satisfied on the neighbourhood.

Given a positive number $s > 0$, we consider the hyper-cube

$$\mathcal{C}_s(0) = \{z_2 \in \mathbb{R}^t : |z_{2,1}| < s \wedge \ldots \wedge |z_{2,t}| < s\}$$

Notice that if the conditions of Theorem 11.1 are satisfied, there exists $s^* > 0$ such that $\mathcal{C}_{s^*}(0) \subset N$. Then, there exists a criterion of choice c such that any hyper-cube $\mathcal{C}_s(0)\,,\, s \in]\,0, s^*[$ satisfies that if $z_2(0) \in \mathcal{C}_s$ then $z_2(k) \in \mathcal{C}_s(0)\,, \forall k{=}1, 2, \ldots$. Indeed, if

$$|z_{2,1}(0)\,| < s \wedge \ldots \wedge |z_{2,t}(0)\,| < s$$

then we have from the structure of the system that

$$|z_{2,1}(1)\,| {=} |z_{2,2}(0)\,| < s \wedge \ldots |z_{2,t-2}(1)\,| {=} |z_{2,t}(0)\,| < s$$

Moreover, by definition of sets (11.22) we have that there exists a criterion of choice c such that

$$|z_{2,t}(1)\,| < U|z_{2,j}(0)\,| < s, j \in \{1, 2, \ldots, t\}$$

and hence we conclude that $z_2(1) \in \mathcal{C}_s(0)$. Notice that this holds for arbitrary $z_2(0) \in \mathcal{C}_s(0)$ and hence we have that $z_2(k) \in \mathcal{C}_s(0)\,, \forall k$.

Consider now any hyper-ball $\mathcal{B}_\varepsilon(0)\,, \varepsilon > 0$ and define $\delta{=}\delta(\varepsilon)\,{=}\min(\varepsilon/2, s^*/2)$. Then if $z_2(0) \in \mathcal{B}_\delta$ we have that $z_2(k, z_2(0)\,) \in \mathcal{C}_\delta(0)\,, \forall k$ since $\delta \in]\,0, s^*[$. Moreover, we have that $\mathcal{C}_\delta(0) \subset \mathcal{B}_\varepsilon(0)\,, \forall \varepsilon > 0$ and hence $z_2(k, z_2(0)\,) \in \mathcal{B}_\varepsilon(0)\,, \forall k$. Therefore there exists a criterion of choice c such that the "c"-resulting system is stable by definition. Q.E.D.

Comment 11.2 QEPCAD can be used to check whether a semi-algebraic set S with the defining expression $S(x)$ is a neighbourhood of a point x^*. Indeed, this can be done by considering the decision problem $(\exists q)\ [S(x) \wedge ||x|| < q]$. □

Theorem 11.2 *Suppose that $\exists j \in \{1, \ldots, t\}$ such that the set $S_1^{z_{2,j}}$ is a neighbourhood of the origin. Then the system is point-minimum phase.* □

Proof of Theorem 11.2: Stability follows from Theorem 11.1. We show now that the system is also locally attractive. We know that there exists a number s^* such that any hyper-cube $\mathcal{C}_s(0)\,, s \in]\,0, s^*[$ is invariant with respect to the solutions $z_2(k)\,, \forall k$. Hence, we have that if

$z_2(0) \in \mathcal{C}_s(0)$

$$|z_{2,t}(k+t-j)| < |z_{2,j}(k)|, \forall k$$

If k=0 we have that $|z_{2,t}(t-j)| = p_0 |z_{2,j}(0)|, p_0 < U < 1$. For k=1 we have that $|z_{2,t}(1+t-j)| = p_1 |z_{2,j}(1)| = p_1 p_0 |z_{2,j}(0)|, p_1 < U < 1$. In general we obtain that

$$|z_{2,t}(N+t-j)| = \prod_{i=0}^{N} p_i |z_{2,j}(0)|, p_i < U < 1$$

and by taking the limit we obtain $\lim_{N\to\infty} |z_{2,t}(N+t-j)| = 0$. Since $z_{2,l}(k+1) = z_{2,l+1}(k)$, $l = 1, \ldots, t-2$ we conclude that

$$\lim_{N\to\infty} |z_{2,l}(N)| = 0, \forall l = 1, \ldots, t-2.$$

In other words, we have that $\lim_{N\to\infty} ||z_2(N)|| = 0$. We can therefore take $\Delta = s^*/2$ and the attractivity of the zero dynamics follows by definition. Q.E.D.

An obvious consequence of the above results is:

Corollary 11.1 *Suppose there exists $j \in \{1, 2, \ldots, t\}$ such that $S_1^{z_{2,j}} = \mathbb{R}^t$. Then there exists a criterion of choice such that the origin of the "c"-resulting system is globally asymptotically stable.* □

We have considered so far only how the sets $S_1^{z_{2,j}}$ can be used to decide on stability of zero dynamics. We show below that for a class of polynomial I-O systems we also may make use of the sets $S_k^{z_{2,j}}$ when dealing with this problem. The following assumptions define the class of systems that we consider.

Assumption 11.2 $\forall z_2 \in \mathbb{R}^t$, *all the solutions u_i to the equation $G(z_2, u) = 0$ satisfy $|u_i| < \infty$.* □

Assumption 11.3 $\forall z_{2,1}, \ldots, z_{2,t-1} \in \mathbb{R}$ there exists a real solution u^* to the equation $G(z_2, u) = 0$ satisfying

$$\lim_{|z_{2,t}|\to 0} |u^*| = 0$$

□

Assumption 11.2 guarantees that the domain of existence of zero dynamics for the set point $y^* = 0$ is the whole state space $\mathbb{R}^t$. Moreover, it implies that there is no criterion of choice which yields finite escape times (for all bounded initial states, all allowable controls that satisfy the constraint (11.17) are bounded for finite time k. For instance, the assumption is satisfied for all explicit zero dynamics (11.15) where the function G_e is a polynomial in all its variables or a rational function with the denominator not having zero values for all values of its arguments.

In general, we can write the implicit zero dynamics (11.14) in the following form

$$g_n(z_2)\,u^n + \ldots + g_0(z_2) = 0.$$

Assumption 11.2 is certainly satisfied if $g_n(z_2) \neq 0, \forall z_2 \in \mathbb{R}^t$ since we have the bound on the roots [20]:

$$|u| < 1 + \sup_i \left|\frac{g_i(z_2)}{g_n(z_2)}\right|$$

A sufficient condition for Assumption 11.3 to be satisfied is that the implicit dynamics have the following form:

$$G(z_2, u) = g_n(z_2)\,u^n + z_{2,t}(\hat{g}_n - 1(z_2)\,u^{n-1} + \ldots + \hat{g}_0(z_2)\,)$$

We use the notation $H = \{z_2 \in \mathbb{R}^t : z_{2,t} = 0\}$. We state now a result for

Theorem 11.3 *Suppose that Assumptions 11.2 and 11.3 are satisfied for the implicit polynomial dynamics (11.14). There exists a criterion of choice such that the origin of the "c"-resulting system is globally attractive if there is an integer N such that $\cup_{i=1}^N S_i^{z_{2,j}} = \mathbb{R}^t - H$ for some $j \in \{1, \ldots, t\}$.* $\square$

Proof of Theorem 11.3: Suppose that conditions of Theorem 11.3 are satisfied. Consider any initial state $z_2(0) \in \mathbb{R}^t$. If $z_2(0) \in H$ then by simply applying $u(k) = 0, \forall k$ we have that $z_2(k) = 0, \forall k \geq t$. If $z_2(0) \in \mathbb{R}^t - H$, then we have that $z_2(0) \in S_{k_1}^{z_{2,j}}, k_1 \in \{1, \ldots, N\}$. By definition of the set $S_{k_1}^{z_{2,j}}$ we have that

$$|z_{2,t}(k_1)\,| = p_{k_1}|z_{2,j}(0)\,|, 0 \leq p_{k_1} < U < 1$$

If $z_2(k_1) \in H$ we trivially have attraction to the origin. Suppose that $z_2(k_1) \notin H$. Then, we have that $z_2(k_1) \in S_{k_2}^{z_{2,j}}, k_2 \in \{1, \ldots, N\}$ and by definition

$$|z_{2,t}(k_2)\,| = p_{k_2}|z_{2,t}(k_1)\,|, 0 \leq p_{k_2} < U < 1$$

Therefore, if we suppose that $z_2(k_i) \notin H, \forall i=1,2,\ldots$ we have that

$$|z_{2,t}(k_N)| = \prod_{i=1}^{N} p_{k_i} |z_{2,j}(0)|, 0 \leq p_{k_i} < U < 1, \forall i$$

and by taking the limit of the above expression we obtain that

$$\lim_{N \to \infty} |z_{2,t}(k_N)| = 0$$

Because of the Assumption 11.3 and since $z_{2,t}(k_N) \to 0$ we have that $z_{2,t-j}(k_N + j) \to 0, j=1,2,\ldots,t-1$ and therefore $||z_2(k)|| \to 0$. The boundedness of $z_2(k), \forall k$ follows trivially from the boundedness of solutions (Assumption 11.2). Q.E.D.

Corollary 11.2 *If the conditions of Theorems 11.1 and 11.3 are satisfied the system (11.14) is point-minimum phase. Moreover, there exists a criterion of choice c such that the origin of the "c"-resulting system is globally asymptotically stable.* □

Notice that Assumption 11.2 is not essential for the global attractivity result and is only used to guarantee that there are no finite escape times.

Comment 11.3 The computational complexity of the decision rules used to define the sets $S_k^{z_{2,j}}$ may be prohibitive and hence it is of utmost importance to investigate ways in which the complexity can be reduced. The required computations may be drastically reduced by first decomposing the polynomial G which defines the implicit zero dynamics (11.14)

$$G(z_{2,1}, \ldots, z_{2,t}, u) = \prod_{i=1}^{M} f_i(z_{2,1}, \ldots, z_{2,t}, u) \tag{11.23}$$

where f_i are all irreducible polynomials. We can do this using some of the symbolic computation packages, e.g. "factor" command in Maple. Notice that $G=0$ if $f_i=0$ for some i and if any of the newly defined implicit zero dynamics

$$f_i(z_{2,1}, \ldots, z_{2,t}, u) = 0$$

satisfies conditions of some of the Theorems 11.1, 11.2 or 11.3, the zero dynamics (11.14) have at least the same properties as the newly defined zero dynamics. Moreover, if some of the polynomials f_i in (11.23) have one of the forms listed below, we can use more explicit tests.

1. If $f_i(u(k-1), u(k))=u(k)-g(u(k-1))$ we can use results from [15]

2. If $f_i(u(k-1), u(k))=g(u(k), u(k-1))$ we can use results from the next section of this chapter

3. If $f_i(u(k-1), u(k))=u(k)+\sum_i b_i u(k-i)$ we can easily check the stability by checking whether all the eigenvalues of the matrix

$$\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & \dots & & \dots & \\ 0 & 0 & 0 & \dots & 1 \\ -b_1 & -b_2 & -b_3 & \dots & -b_t \end{pmatrix}$$

   are inside the unit disk.

The idea of factorising an implicit system into several subsystems can be found in [159] but we presented here tools for obtaining this factorisation and tests of stability of zero dynamics for I-O polynomial systems. □

Comment 11.4 It is important to emphasize that the *uniform point-minimum phase property* can be tested using a very similar method. The only difference is that we redefine the decision problems used to compute the sets as follows. The expression $S_1^{z_{2,j}}(z_2)$ is computed by considering:

$$(\forall u(0))\,[|u_0| < U|z_{2,j}| \wedge G(z_2, u_0)=0]$$

Hence, instead of the quantifier $\exists$ we use $\forall$ in the computation of the sets. QEPCAD computes the expressions that define these sets in the same way as before. □

A Set-Minimum Phase Test

The following sufficient condition for set minimum phaseness is easily proved.

Theorem 11.4 *Suppose that $S_1^{z_{2,1}}=\mathbb{R}^t - L$, where L is a bounded set ($\sup_{x,y\in L} d(x,y) < \infty$). Then the system is set-minimum phase.* □

Proof of Theorem 11.4: From the definition of the set $S_1^{z_{2,1}}$ we have that there exists a criterion of choice c such that the "c"-resulting system has the property $||z_2(1)|| < ||z_2(0)||, \forall z_2(0) \in S_1^{z_{2,1}}$.

Moreover, because of Assumption 11.2 we have that $||z_2(1)|| < \infty, \forall z_2(0)$. Hence, we can find a number $s < \infty$ such that the hyper-ball $\mathcal{B}_s(0)$, $L \subset \mathcal{B}_s(0)$ is an invariant set. Global asymptotic stability of $\mathcal{B}_s(0)$ follows directly from the fact that $\mathbb{R}^t - \mathcal{B}_s(0) \subset S_1^{z_{2,1}}$. Q.E.D.

Comment 11.5 In a very similar way we can prove other stability properties and also we can use other sets for this purpose. For example, we could work with sets

$$S_1 = \{z_2 : \exists u \in \mathbb{R}, ||z(1)||^2 < U||z_2(0)||^2, G(z_2(0), z_{2,t}(1)) = 0\}, etc.$$

□

Comment 11.6 QEPCAD can be used to compute (or construct) Lyapunov functions if we know from converse Lyapunov theorems [79] that they belong to a class of polynomial functions. For example, all polynomial systems which have stable linearisation allow for quadratic Lyapunov functions [79]. So suppose we want to check stability of the system:

$$x(k+1) = F(x(k)) \tag{11.24}$$

with $x \in \mathbb{R}^n$ and F is a vector polynomial function in all its arguments. We can choose the quadratic function:

$$V(x(k)) = x^T(k)\, H\, x(k),\ H = H^T,\ H > 0,\ H \in \mathbb{R}^{n \times n}$$

By considering the following set of formulas:

$$(\exists H)\,[(H - H^T = 0) \wedge (H > 0) \wedge (F^T(x)\, H\, x F(x) - x^T H x < 0)$$

we can compute the existence of a quadratic Lyapunov function. Moreover, *all possible matrices* H which define Lyapunov functions can be found using QEPCAD. With any such Lyapunov function we can then estimate the domain of attraction which is very important in applications.

Another important observation to be made here is that all systems (11.24) whose equilibrium is attractive must necessarily have the property that $\exists N$ such that $\cup_j S_N^{z_{2,j}}$ is a neighbourhood of the origin. This means that if we want to check also the stability, we can just consider the formulas:

$$(\forall \varepsilon)\,(\exists \delta)\,[(\varepsilon > 0) \wedge (\delta > 0) \wedge (||x|| < \delta) \wedge (||F(x)|| < \delta) \wedge \ldots \wedge (||F^{N-1}(x)|| < \delta)]$$

which qualify as input formulas to the QEPCAD algorithm and hence can be decided in finite time.

We do not know *a priori* the number N, which can be very large. Notice that if the origin of the system is stable but not attractive, we can not check stability by definition using QEPCAD since we need to check an infinite number of conditions $||F^i(x)|| < \delta, \forall i$. $\square$

11.4 Scalar Implicit Dynamics

The purpose of this section is to present more explicit conditions for set-minimum phase properties for a class of one dimensional implicit polynomial dynamics. Although the class of systems is very simple, the algebraic conditions that we obtain are not. Results of this section indicate that similar conditions for higher order implicit dynamics, even if obtained, would be very complicated. This, in a way, motivates the algorithmic approach that we presented in the first part of this chapter. We extend Theorem 6.2 in [15] to the class of polynomial implicitly defined polynomial systems. Theorem 6.2 in [15] gives necessary and sufficient conditions for global stability of an invariant bounded set (interval) for the class of dynamical systems defined by $u(k) = g(u(k-1))$ with g continuous. We consider implicitly defined polynomial systems. The equation that defines the behaviour of the system is given below:

$$f(u(k-1), u(k)) = 0, \tag{11.25}$$

where $f(v, u) \in \mathbb{Q}[v, u]$. Assumption 11.2 holds for systems (11.25). We now cite Theorem 6.2 from [15] which is used in the proof of the main result of this section.

Theorem 11.5 *[15] Consider the map $g : \mathbf{D} \to \mathbf{D}$, $\mathbf{D} \subset \mathbb{R}$. Let $\mathcal{A} \stackrel{\triangle}{=} [a, b] \subset \mathbb{R}$ such that:*

1. *$\mathbf{D} \cap \mathcal{A}$ is invariant under g: $g(\mathbf{D} \cap \mathcal{A}) \subset \mathbf{D} \cap \mathcal{A}$*

2. *$(\mathbb{R} -]a, b[) \subset \mathbf{D}$*

3. *g is continuous on $(\mathbb{R} -]a, b[)$*

Then $\mathcal{A}$ is globally attracting interval of the iterative map $u(k+1) = g(u(k))$ if and only if the following conditions hold:

$$\forall x < a \qquad\qquad g(x) > x \tag{11.26}$$

$$\forall x > b \qquad\qquad g(x) < x \tag{11.27}$$

$$\forall x < a \quad \textit{such that}\ \exists(x,z) \in G_R^{-1} \quad g(x) < z \tag{11.28}$$

$$\forall x > b \quad \textit{such that}\ \exists(x,z) \in G_L^{-1} \quad g(x) > z \tag{11.29}$$

The domain **D** represents the domain of definition of zero dynamics. Other symbols used in the statement of Theorem 11.5 are given below:

$$G = \{(x, g(x)) : x \in \mathbb{R} - [a,b]\} \tag{11.30}$$

$$G_L = \{(x, g(x)) \in G : x < a\} \tag{11.31}$$

$$G_R = \{(x, g(x)) \in G : x > b\} \tag{11.32}$$

$$G_L^{-1} = \{(g(x), x) : (x, g(x)) \in G_L\} \tag{11.33}$$

$$G_R^{-1} = \{(g(x), x) : (x, g(x)) \in G_R\} \tag{11.34}$$

Comment 11.7 Because of Assumption 11.2, the domain of definition of zero dynamics is the whole real line, that is **D**=$\mathbb{R}$. Therefore, Condition 2 of Theorem 11.5 does not need to be verified. □

Given $T \geq 0$ a real number, the following sets will be used in the sequel:

$$S_1 = \{(v,u) \in \mathbb{R}^2 : v < -T\}; \quad S_2 = \{(v,u) \in \mathbb{R}^2 : v > T\} \tag{11.35}$$

Also, we use the variety:

$$V_z = \{(v,u) \in \mathbb{R}^2 : f(v,u) = 0\} \tag{11.36}$$

All results that are obtained in this section are based on the concept of an "inverse graph" of the variety V_z (11.36) which is given by:

$$V_z^{-1} = \{(v,u) \in \mathbb{R}^2 : f(u,v) = 0\} \tag{11.37}$$

So the "inverse graph" of the variety V_z is obtained by simply interchanging variables v and u in the defining polynomial.

A very important feature of polynomial systems which is crucial for the stability of zero dynamics is given in the lemma below.

Lemma 11.1 *Consider the real variety V_z defined by (11.36). There exists $D_1 \geq 0$ such that there are constant numbers n_1 and n_2 of continuous branches*[3] *of variety V_z on sets $]-\infty, -D_1] \times \mathbb{R}$ and $[D_1, +\infty[\times \mathbb{R}$.* □

Lemma 11.1 was proved in Chapter 5.

Comment 11.8 The main results that are presented in this section are based on geometric arguments. Indeed, Lemma 11.1 states that it is possible to find an interval $v \in [-D_1, D_1]$ inside which all bifurcations of the variety V_z occur. Furthermore, from the theorem on the continuity of roots (see Appendix B, Theorem B.7) we see that all intersections between branches of the variety V_z occur inside the same interval. Intersections between V_z (modulo common components) and bisectors $u{=}v$ and $u{=}-v$ also occur on a set of the form $[-D_2, D_2] \times \mathbb{R}, D_2 > 0$. Finally, all intersections between V_z and the variety V_z^{-1} can be confined to a set of the form $[-D_3, D_3] \times \mathbb{R}, D_3 > 0$. This is proved in the proof of Theorem 11.6. We note that hereafter we assume that the set S_1 and S_2 are defined using $T{=}\max(D_1, D_2, D_3)$. □

Lemma 11.2 *A necessary condition for the system (11.25) to be set-minimum phase is*

$$\sup_{|v|<K} \inf_{(v,u)\in V_z} |u| < +\infty, \quad \forall K \in]0, +\infty[$$

□

Proof of Lemma 11.2: Suppose that there exists a criterion of choice c whose "c"-resulting system is stable. Suppose that there exists $v{=}u(k-1)^*$ which belongs to the invariant interval such that all branches of the variety V_z have a vertical asymptote at $v{=}u(k-1)^*$. In other words, the condition of Lemma 11.2 is not satisfied for any neighbourhood of the origin that contains $u(k-1)^*$. It is then obvious that the invariant interval must have one of the following forms: $]-\infty, +\infty[$, $[K, +\infty[$ or $]-\infty, K]$ and we have a contradiction since none of these intervals is bounded. Suppose now that $u(k-1)^*$ does not belong to the invariant interval. In this case, there does not exist a criterion of choice for which the invariant interval is asymptotically stable because

[3]The term "branch of V_z" that we use corresponds to parts of irreducible varieties (curves) from which the variety V_z is composed [20, 37, 61] that belong to sets $]-\infty, -D_1] \times \mathbb{R}$ and $[D_1, +\infty[\times \mathbb{R}$.

for $u(k-1)$ such that $u(k-1) \to u(k-1)^*$ we have that $|u(k)| \to +\infty$, so we again obtain a contradiction. Q.E.D.

Now we can give definitions of maximal and minimal branches of the variety V_z.

Definition 11.8 Consider the variety V_z on sets S_1 and S_2. The maximal branch of V_z in S_2 is given by $V_M^{S_2}=\{(v,u)\in V_z : v\in S_2,\ u=\max_{(v,y)\in V_z,\ y<v} y\}$. The minimal branch of V_z in S_1 is defined as $V_m^{S_1}=\{(v,u)\in V_z : v\in S_1,\ u=\min_{(v,y)\in V_z,\ y>v} y\}$. □

Notice that minimal and maximal branches are well defined parts of irreducible varieties of V_z, following from the theorem on continuity of roots (see Appendix B, Theorem B.7) and Bezout's theorem (see Appendix B, Theorem B.8). Bezout's theorem says that we can find a set $[-D_3, D_3]\times\mathbb{R}$ inside which all intersections between the variety V_z and the bisector $u=v$ occur (this excludes components of V_z which have infinitely many common points with the bisector and which are defined by polynomials of the form $(u-v)^i,\ i\in\{1,2,\ldots\}$). Also notice that if there are no branches in S_2 that are above the bisector $u=v$, then by definition $V_M^{S_2}=\emptyset$.

Comment 11.9 Suppose that we can find a criterion of choice such that outside a bounded interval $[-T,T]$ all orbits are bounded, converge to the interval and enter it in finite time from any given $u(-1)$. Then it is easy to show that when Lemma 11.2 holds there exists an interval (perhaps larger than $[-T,T]$ but bounded) such that it is invariant and stable. Consequently, we will concentrate only on the existence of a bounded asymptotically stable interval and Lemma 11.2 guarantees that we can always have a criterion of choice for all $u(-1)\in[-T,T]$ which renders the interval invariant. □

Now we can state the main result.

Theorem 11.6 *A polynomial system with implicitly defined zero dynamics (11.25) is set-minimum phase if and only if the criterion of choice $u(k)=c(u(k-1))$ defined as*

$$u(k)=\begin{cases} y & \text{such that } (u(k-1),y)\in V_m^{S_1} \text{ if } u(k-1)<-T \\ y & \text{such that } (u(k-1),y)\in V_M^{S_2} \text{ if } u(k-1)>T \\ y & \text{such that } (u(k-1),y)\in V_z \text{ if } u(k-1)\in[-T,T] \text{ and } y \text{ has the smallest} \\ & \text{absolute value} \end{cases}$$

satisfies equations (14)-(17) of Theorem 11.5 and Lemma 11.2 holds. □

Proof of Theorem 11.6:

Sufficiency: Consider the criterion of choice:

$$u(k)=\begin{cases} y & \text{such that } (u(k-1),y)\in V_m^{S_1} \text{ if } u(k-1)<-T \\ y & \text{such that } (u(k-1),y)\in V_M^{S_2} \text{ if } u(k-1)>T \\ y & \text{such that } (u(k-1),y)\in V_z \text{ if } u(k-1)\in[-T,T] \text{ and } y \text{ has the smallest} \\ & \text{absolute value} \end{cases}$$

It is obvious that all the conditions of Theorem 11.5 are satisfied and for this criterion of choice the "c"-resulting system has an invariant interval which is globally asymptotically stable.

Necessity: We only have to show that the conditions (11.26),(11.27),(11.28),(11.29) are necessary for set-minimum phase property. We can find a set inside which all intersections between the variety V_z and the bisector $u{=}v$ occur and denote it as $[-D_3,D_3]\times\mathbb{R}$. Moreover, we can find another set inside which all the intersections between V_z and V_z^{-1} occur (modulo common components which may have infinitely many common points) and denote it as $[-D_2,D_2]\times\mathbb{R}$. We again emphasize that all the arguments are given for the sets S_1 and S_2 defined by the number $T{=}\max[D_1,D_2,D_3]$. Sets S_1 and S_2 (11.35) defined in this way obviously have the property that (modulo common components) there are no intersections between V_z and V_z^{-1} on the sets, there are no bifurcations of the variety V_z on the sets and, finally, minimal and maximal branches $V_m^{S_1}$ and $V_M^{S_2}$ are either parts of continuous curves or they are empty sets.

Suppose that the zero dynamics are stable and that condition (11.26) is not satisfied. Since $V_m^{S_1}{=}\emptyset$, all branches are below the bisector $u{=}v$ and as a consequence we have that $u(k)\to-\infty$ as $k\to\infty$, $\forall u(-1)\in]-\infty,-T]$. A similar situation happens when the condition (11.27) is not satisfied. In other words $V_M^{S_2}{\neq}\emptyset$ and $V_m^{S_1}{\neq}\emptyset$ are necessary for set-minimum phaseness.

Consider now what happens if condition (11.28) is not satisfied. Since $V_M^{S_2}$ is such that all branches of V_z in S_2 are above it, all their inverses will lay on the left hand side (or below) of $(V_M^{S_2})^{-1}$. Thus, we suppose that no branch of V_z^{-1} satisfies condition (11.28). Moreover, if we use pieces of branches of V_z to construct a piecewise continuous one to one function and use the modified Theorem 11.5 [14] we can see that no such functions would satisfy the conditions of Theorem 11.5. Therefore, the system is not set-minimum phase. The contradiction completes the proof. Conditions (11.28) and (11.29) are symmetric and they are either both satisfied or both not. Q.E.D.

11.4.1 An Algebraic Set-Minimum Phase Test

Theorem 11.6 extends Theorem 11.5 [14] to a case where zero dynamics are defined by an implicit polynomial equation. We now present an algebraic method to check the conditions of Theorem 11.6. Moreover, we classify all possible cases that may happen when the conditions of Theorem 11.6 are satisfied. We also illustrate the method by two examples. First, we provide a means of verifying the conditions of Lemma 11.2.

We write the function (11.25) as

$$f(v,u) = g_n(v)\, u^n + \ldots + g_0(v) \tag{11.38}$$

The only critical points that we have to check are the ones for which the leading coefficient $g_n(v)$ (11.38) vanishes [20, pg. 10, pg. 39]. Therefore, the first step is to find all real solutions v to $g_n(v) = 0$. It is then necessary to check whether

$$f(v,u) = 0 \tag{11.39}$$

has real roots u, for all critical values of v. We define the following sets:

$$\mathcal{A} = \{v : g_n(v) = 0\} \tag{11.40}$$

$$\mathcal{D}(v) = \{u \in \mathbb{R} : f(v,u) = 0, v \in \mathcal{A}\} \tag{11.41}$$

$$\mathcal{E} = \{(v,u) : v \in \mathcal{A},\ u \in \mathcal{B}(v)\} \tag{11.42}$$

There must be at least one real root $u \in \mathcal{D}(v)$, $\forall v \in \mathcal{A}$, otherwise Assumption 11.2 would not be satisfied. We can now use the implicit function theorem [43]. For all pairs of controls $(v,u) \in \mathcal{E}$ the equation (11.25) holds. If for every $v \in \mathcal{A}$ there exists at least one $u \in \mathcal{D}(v)$ for which:

$$\frac{\partial f}{\partial u}\Big|_{(v,u)} \neq 0 \tag{11.43}$$

then the implicit function theorem guarantees the existence of a function $u = c(v)$, which is C^∞ since we deal with polynomials, such that

$$f(v, c(v)) = 0.$$

The implicit function theorem gives only sufficient conditions to check Lemma 11.2 but they are easy to check. If (11.43) does not hold, we may check whether Lemma 11.2 is satisfied. The easiest way to do this is to draw the variety V_z around every point (v, u) in $\mathcal{E}$ using Matlab (the set $\mathcal{E}$ contains finitely many points) and check whether there exists a branch of V_z which does not have a vertical asymptote at (v, u).

Before we give the classification of all possible situations we define bisectors and octants that we use.

$$B_1=\{(v,u)\in\mathbb{R}^2 : v=u\};\ B_2=\{(v,u)\in\mathbb{R}^2 : -v=u\}$$

$$O_1=\{(v,u)\in\mathbb{R}^2 : v>0,\ u>0,\ u<v\};\ O_2=\{(v,u)\in\mathbb{R}^2 : v>0,\ u>0,\ u>v\}$$

$$O_3=\{(v,u)\in\mathbb{R}^2 : v<0,\ u>0,\ u>-v\};\ O_4=\{(v,u)\in\mathbb{R}^2 : v<0,\ u>0,\ u<-v\}$$

$$O_5=\{(v,u)\in\mathbb{R}^2 : v<0,\ u<0,\ u>v\};\ O_6=\{(v,u)\in\mathbb{R}^2 : v<0,\ u<0,\ u<v\}$$

$$O_7=\{(v,u)\in\mathbb{R}^2 : v>0,\ u<0,\ u<-v\};\ O_8=\{(v,u)\in\mathbb{R}^2 : v>0,\ u<0,\ u>-v\}$$

We also use notation A_1 and A_2 to denote lines v=0 and u=0 in $\mathbb{R}^2$. It is easy to check that if a point on a variety V_z is in the first octant O_1, the corresponding point on V_z^{-1} is in the second octant O_2 and vice versa. We use the following notation to summarise all possible situations:

$$O_2 \leftrightarrow O_1,\ O_3 \leftrightarrow O_8,\ O_4 \leftrightarrow O_7,\ O_5 \leftrightarrow O_6$$

In some cases the position of branches $V_M^{S_2}$ and $V_m^{S_1}$ provide sufficient information to conclude on the stability of zero dynamics since the conditions on the inverse graph are automatically satisfied. We summarise these trivial cases in the Lemma below.

Proposition 11.1 *1. If one of the following conditions holds*

(a) $V_m^{S_1} \in O_5$ *and* $V_M^{S_2} \in O_1$

(b) $V_m^{S_1} \in O_5$ *and* $V_M^{S_2} \in O_8$

(c) $V_m^{S_1} \in O_5$ *and* $V_M^{S_2} \in O_7$

(d) $V_m^{S_1} \in O_4$ *and* $V_M^{S_2} \in O_1$

(e) $V_m^{S_1} \in O_4$ *and* $V_M^{S_2} \in O_8$

(f) $V_m^{S_1} \in O_3$ *and* $V_M^{S_2} \in O_1$

then there exist a criterion of choice which yields stable zero dynamics.

2. *If* $V_m^{S_1} \subset B_2$ ($V_M^{S_2} \subset B_2$) *then there exists a criterion of choice which yields stable zero dynamics if and only if* $V_M^{S_2}$ ($V_m^{S_1}$) *belongs to the cone* $\{(v,u) \in \mathbb{R}^2 : |v| < |u|\}$.

3. *If* $V_m^{S_1} \subset A_2$ *or* $V_M^{S_2} \subset A_2$*, the zero dynamics are stable.*

4. *If* $V_m^{S_1}{=}\emptyset$ *or* $V_M^{S_2}{=}\emptyset$ *or* $V_m^{S_1}{=}\emptyset$ *and* $V_M^{S_2}{=}\emptyset$ *then the zero dynamics are unstable.*

5. *If* $V_m^{S_1} \in O_3$ *or* $V_M^{S_2} \in O_7$ *or* $V_m^{S_1} \in O_3$ *and* $V_M^{S_2} \in O_7$ *then the zero dynamics are unstable.*

□

Comment 11.10 If point 3 of Proposition 11.1 holds, and if there exists a control law which zeroes the output in C steps (the systems is output dead-beat controllable), then there exists a control law which zeroes the state of the system in $C+1$ steps (the system is state dead-beat controllable). □

It can easily be checked that the only remaining cases are:

1. $V_m^{S_1} \in O_3$ and $V_M^{S_2} \in O_8$

2. $V_m^{S_1} \in O_4$ and $V_M^{S_2} \in O_7$

Only in these cases do we have to use "inverses" $(V_m^{S_1})^{-1}$ and $(V_M^{S_2})^{-1}$. Since we are dealing with polynomial systems, we can use the algebraic structure of these systems in order to obtain a "box" inside which all intersections between V_z and V_z^{-1} occur (modulo common components). We will use the theory of resultants to compute such a box. We denote $f_1{=}f(v,u)$ and $f_2{=}f(u,v)$.

Resultants procedure:

First, we find the greatest common divisor of f_1 and f_2 which is denoted as $GCD(f_1,f_2) \in \mathbb{Q}[v,u]$. Then we compute "common components free" polynomials:

$$\begin{aligned} f_1^{ccf} &= \frac{f_1}{GCD(f_1,f_2)} \\ f_2^{ccf} &= \frac{f_2}{GCD(f_1,f_2)} \end{aligned} \tag{11.44}$$

Now, we can regard polynomials f_1^{ccf} and f_2^{ccf} as polynomials in v whose coefficients are polynomials in u. Now we can find the resultant of the two polynomials:

$$R(f_1^{ccf}, f_2^{ccf}) = \sum_{i=0}^{p} a_i u^i \tag{11.45}$$

The resultant $R(f_1^{ccf}, f_2^{ccf})$ is a polynomial in u. We know that polynomials f_1^{ccf} and f_2^{ccf} have no common roots if $R(f_1^{ccf}, f_2^{ccf}) \neq 0$. We can find a number D_2 which is such that all absolute values of real roots of the resultant are less than D_2.

Second, we estimate the number D_2 using formulas for bounds on roots, e.g. $\hat{D}_2 = 1 + \sup_i |a_i|$, where $a_i, i = 0, 1, \ldots, p$ are coefficients of the resultant. Outside the box defined by $\{(v, u) \in \mathbb{R}^2 : |v| \leq \hat{D}_2$ and $|u| \leq \hat{D}_2\}$ the varieties V_z and V_z^{-1} have no intersections modulo common branches.

Third, we pick $\hat{u}$ such that $|\hat{u}| > |\hat{D}_2|$ and find sets of solutions:

$$\Sigma_1 = \{v \in \mathbb{R} : f(v, \hat{u}) = 0\}; \quad \Sigma_2 = \{v \in \mathbb{R} : f(\hat{u}, v) = 0\} \tag{11.46}$$

We can see that the sets Σ_1 and Σ_2 give a complete picture about the branches of varieties V_z and V_z^{-1} and therefore can be used to check whether zero dynamics are stable for the two remaining cases. The criterion for the stability of zero dynamics of the two last cases, which are not covered by Lemma 11.2, is given in the following proposition.

Proposition 11.2 *If*

1. $V_m^{S_1} \in O_3$ *and* $V_M^{S_2} \in O_8$ *or*

2. $V_m^{S_1} \in O_4$ *and* $V_M^{S_2} \in O_7$

then the system is set-minimum phase if there exist $\sigma_1 \in \Sigma_1$ *and* $\sigma_2 \in \Sigma_2$ *such that* $\sigma_1 < \sigma_2$. *In the first case sets* Σ_1 *and* Σ_2 *(11.46) are calculated using* $\hat{u} > \hat{T}$ *and in the second case* $\hat{u} < -\hat{T}$.

□

Proof of Proposition 11.2: It trivially follows from Theorem 11.6 and the above given procedure.

The method to check the set-minimum phase property consists of several steps:

1. Check the conditions of Lemma 11.2 as described before.

2. Form the Sturm sequence and find all leading coefficient functions. Using results from Chapter 5 and bounds on roots (Appendix B), determine the estimate $\hat{D}_1$.

3. Find the box inside which all intersections between the variety V_z and B_1, B_2, A_1 and A_2 occur. This is done in the following way. Find the following estimates:

$$\hat{D}_3{=}1+\max_i |n_i|;\ \hat{D}_4{=}1+\max_i |m_i|;\ \hat{D}_5{=}1+\max_i |k_i|;\ \hat{D}_6{=}1+\max_i |l_i|$$

where $n_i,\ m_i,\ k_i,\ l_i \in \mathbb{R}$ are respectively coefficients of polynomials $f(v,v)$, $f(v,-v)$, $f(0,u)$ and $f(v,0)$.

4. Find the estimate $\hat{T}$ of T using:

$$\hat{T}{=}\max(\hat{D}_1, \hat{D}_3, \hat{D}_4, \hat{D}_5, \hat{D}_6) \tag{11.47}$$

5. Pick any $v^* \in]-\infty, -\hat{T}[$ and compute all real roots of

$$f(v^*, u) = 0 \tag{11.48}$$

Pick any $v^{**} \in]\hat{T}, +\infty[$ and compute all real roots of

$$f(v^{**}, u) = 0 \tag{11.49}$$

6. Determine to which octants the pairs (v^*, real root to (11.48)) and (v^{**}, real root to (11.49)) belong and check whether Proposition 11.1 holds (remember that checking the position of a single point of the variety implies that the whole branch has the same position). If Proposition 11.1 is not satisfied then proceed onto the next step.

7. Compute $\hat{D}_2{=}1+\max_i |t_i|$ where t_i are the coefficients of the resultant $R(f_1^{ccf}, f_2^{ccf})$, redefine $\hat{T}{=}\max(\hat{D}_1, \hat{D}_2, \hat{D}_3, \hat{D}_4, \hat{D}_5, \hat{D}_6)$ and apply the resultants procedure which is used to check conditions of Proposition 11.2.

11.4.2 *Examples*

The following example illustrates the method for checking the existence of stable zero dynamics.

Example 11.4 Check the existence of stable zero dynamics for the following system:

$$y(k+1) \;=\; -2(1+y(k)^2)\,u^5(k) - 2u^3(k) + 2u(k)\,u(k-1)\,(1+y(k)^4)$$

$$+ 2u(k)\, u(k-1)^2 + u(k-1)\, u(k)^4 + u(k-1)\, u(k)^2 - u(k-1)^2$$
$$- u(k-1)^3 + y(k)^3$$

For $y(k) = 0$ we have:

$$-2u^5(k) - 2u^3(k) + 2u(k)\, u(k-1) + 2u(k)\, u(k-1)^2 + u(k-1)\, u(k)^4$$
$$+ u(k-1)\, u(k)^2 - u(k-1)^2 - u(k-1)^3 = 0 \qquad (11.50)$$

Therefore, the variety V_z is defined by:

$$V_z = \{(v, u) \in \mathbb{R}^2 : -2u^5 - 2u^3 + 2uv + 2uv^2 + vu^4 + vu^2 - v^2 - v^3 = 0\}.$$

Step 1: Since $g_5(v) = -2$ the conditions of Lemma 11.2 are satisfied.

Step 2: Using Maple software package (command “rem”) we obtain the following Sturm sequence:

$$\begin{aligned}
f(v,u) &= -2u^5 - 2u^3 + 2uv + 2uv^2 + vu^4 + vu^2 - v^2 - v^3 \\
f_1 &= -10u^4 - 6u^2 + 2v + 2v^2 + 4vu^3 + 2vu \\
f_2 &= -(-\frac{4}{5} + \frac{2}{25}v^2)\, u^3 - \frac{12}{25}vu^2 - (\frac{41}{25}v^2 + \frac{8}{5}v)\, u + \frac{24}{25}(v^2 + v^3) \\
f_3 &= -25\frac{(-24 + 7v^4 + 8v^3 - 80v - 82v^2)\, u^2}{(-10 + v^2)^2} \\
&\quad + 50\frac{v(-15v^2 + 4v^3 + 4v^4 - 16v - 4)\, u}{(-10 + v^2)^2} \\
&\quad - 50\frac{4v^3 + v^4 + v^5 + 4 + 4v + 4v^2}{(-10 + v^2)^2} \\
f_4 &= -[8(12800v + 41680v^2 + 68240v^3 + 52516v^4 + 7268v^5 - 10960v^6 \\
&\quad - 3152v^7 + 449v^8 + 133v^9 + 8v^{10} + 4v^{11} + 1600] / [25(-24 + 7v^4 \\
&\quad + 8v^3 - 80v - 82v^2)^2] + [v(161600v + 548160v^2 + 923680v^3 \qquad (11.51) \\
&\quad + 727392v^4 + 113716v^5 - 142400v^6 - 41100v^7 + 4456v^8 + 1033v^9 \\
&\quad + 196v^{10} + 100v^{11} + 19200)\, u] / [25(-24 + 7v^4 + 8v^3 - 80v - 82v^2)^2] \\
f_5 &= [50(49v^{15} + 161v^{14} - 2148v^{13} - 8948v^{12} + 27908v^{11} + 175332v^{10} \\
&\quad + 5760v^9 - 1338048v^8 - 2333952v^7 + 1619072v^6 + 10299904v^5 + 15313920v^4 \\
&\quad + 11967488v^3 + 5407744v^2 + 1351680v + 147456)\, v] / [(25v^5 + 24v^4
\end{aligned}$$

$$+ 728v^3 + 1360v^2 + 848v + 192)^2 (- 10 + v^2)^2]$$

From the Sturm sequence we find the leading coefficient functions:

$$\begin{aligned}
&-2,\ -10,\ -(- \frac{4}{5} + \frac{2}{25}v^2),\\
&-25\frac{(- 24 + 7v^4 + 8v^3 - 80v - 82v^2)}{(- 10 + v^2)^2},\\
&[v(161600v + 548160v^2 + 923680v^3 + 727392v^4 + 113716v^5\\
&-142400v^6 - 41100v^7 + 4456v^8 + 1033v^9\\
&+ 196v^{10} + 100v^{11} + 19200)] / [25(- 24 + 7v^4 + 8v^3 - 80v - 82v^2)^2], \qquad (11.52)\\
&[50(49v^{15} + 161v^{14} - 2148v^{13} - 8948v^{12} + 27908v^{11} + 175332v^{10}\\
&+ 5760v^9 - 1338048v^8 - 2333952v^7 + 1619072v^6 + 10299904v^5 + 15313920v^4\\
&+ 11967488v^3 + 5407744v^2 + 1351680v + 147456)\, v] / [(25v^5 + 24v^4\\
&+ 728v^3 + 1360v^2 + 848v + 192)^2 (- 10 + v^2)^2]
\end{aligned}$$

Using the formula for bounds on roots (see Appendix B) we find that the highest coefficient functions do not change their signs for v that belongs to intervals $]-\infty, -312529.98[$ and $]\,312529.98, +\infty[$. In other words, the estimate of D_1 is $\hat{D}_1$=312529.98.

Step 3: All intersections of the variety V_z with A_1, A_2, B_1 and B_2 lay in the interval $]-4, +4[$. It is easy to check that $\hat{D}_3$=2, $\hat{D}_4$=4, $\hat{D}_5$=2 and $\hat{D}_6$=3.

Step 4: Therefore, the estimates of sets S_1 and S_2 are defined using the number $\hat{T}$=312529.98.

Step 5: We now substitute any number v from the interval $]-\infty, -312529.98[$ into (11.50) and find all real roots. By choosing v^*=−312530, we obtain the following set of points in $\mathbb{R}^2$:

$$\{(- 312530, u) : (- 312530, + 559.04293), (- 312530, -559.04293), (- 312530, -156265)\}$$

Similarly, with v^{**}=312530, we obtain the set of roots

$$\{(312530, u) : (+ 312530, 559.04383), (312530, -559.04383), (312530, 156265)\}$$

These points indicate the positions of branches and hence $V_m^{S_1} \subset O_5$ and $V_M^{S_2} \subset O_1$.

Step 6: We conclude that this system is set-minimum phase since point 1.a of Proposition 11.1 is satisfied.

From this example we see that although we started with a polynomial with the highest degree in $u(k-1)$ equal to 3 the highest exponent of $u(k-1)$ in the Sturm sequence is 15. This is a drawback of the method. Also, we could work with better bounds on roots in order to obtain better estimates to intervals S_1 and S_2 or even find the exact roots for polynomials in the Sturm sequence. However, the proposed method is able to check existence of zero dynamics quickly. □

Example 11.5 Consider zero dynamics which evolve according to:

$$(u(k)\,u(k-1)+1)\,(u(k)^2+u(k-1)^2)=0 \tag{11.53}$$

In this case Assumption 11.2 is satisfied and we have that

$$V_z=\{(v,u)\in\mathbb{R}^2:(uv+1)\,(u^2+v^2)=0\}$$

Therefore, the variety V_z consists of the origin $(0,0)$ and the hyperbola $u=-1/v$. It is important to notice that the variety V_z is such that for any fixed value of v there is only one value of u such that $(v,u)\in V_z$. In this example we have that

$$\sup_{|v|<K}\ \inf_{(v,u)\in V_z}|u|=\sup_{|v|<K,\ (v,u)\in V_z}|u|=+\infty,\quad \forall K\in]0,+\infty[.$$

It is obvious now that the condition of Lemma 11.2 is not satisfied and we conclude that the system is not set-minimum phase. In other words, it is impossible to find a bounded interval $A\subset\mathbb{R}$ which is asymptotically stable. In this case, we have that the origin is a fixed point and trajectories from any other initial point $u(-1)$ oscillate with period two between points $u(-1)$ and $-1/u(-1)$. □

11.4.3 Output Dead-Beat Control Law With Stable Zero Dynamics

As an example for application of our results, we present an output dead-beat controller for set-minimum phase systems of the form

$$y(k+1)=f(y(k),u(k-1),u(k)) \tag{11.54}$$

We can check output dead-beat controllability using the results from Chapter 6. We can then check whether the system is set-minimum phase by using the material from the previous section. A minimum-time output dead-beat controller for set-minimum phase systems (stable invariant

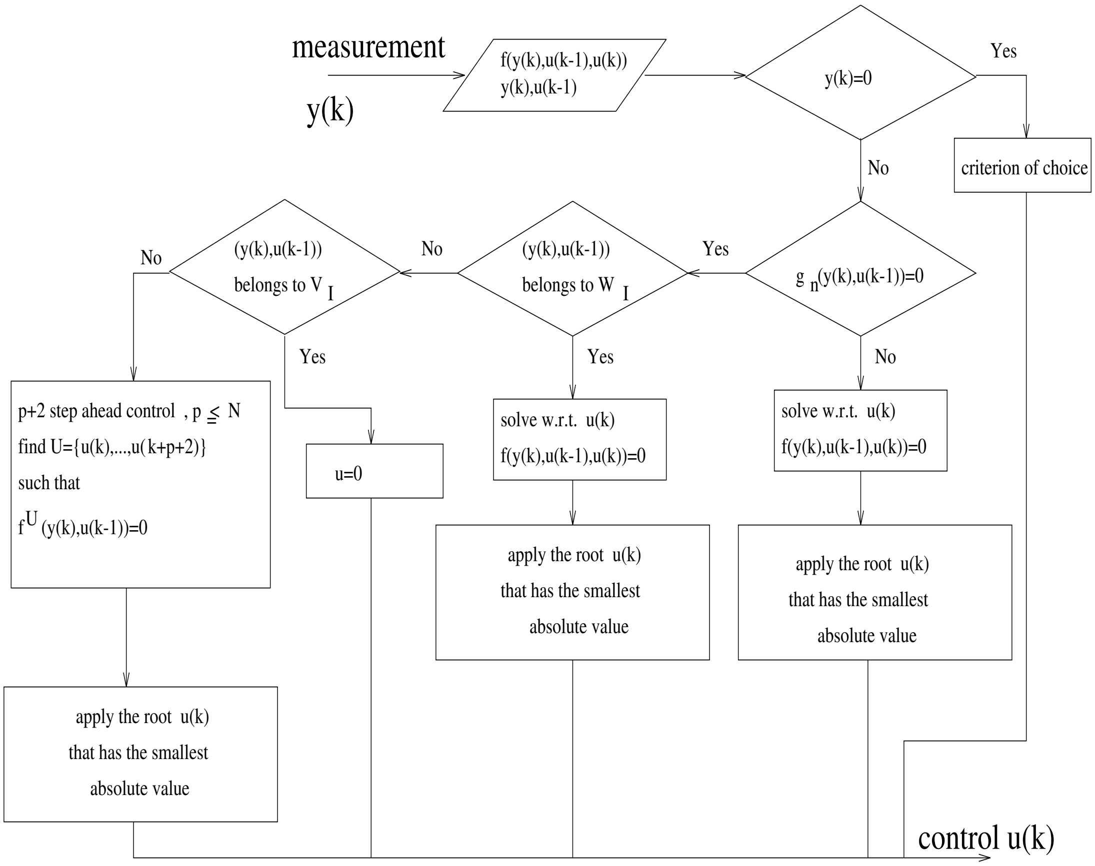

Figure 11.2: Output dead-beat controller - algorithm

interval) is presented below. Finally, a short case study of a fan and a radiator system is presented.

The output dead-beat controller (algorithm) is outlined in Figure 11.2. The obtained controller uses static feedback to compute the value of control signal at any time instant k. The closed loop system can be written in the form:

$$y(k+1) = f(y(k), u(k-1), u(k)) \tag{11.55}$$

$$u(k) = g(y(k), u(k-1)) \tag{11.56}$$

The control signal is obtained as a solution to a polynomial algebraic equation and since there may be more than one solution we need a criterion of choice to define the control law $g(y(k), u(k-1))$. One criterion for the choice may be: apply the control signal that has the least absolute value. We may be able to shape the transient response and keep the control signals as small as feasible, using a different criterion of choice. The question of which choice is not so critical if the output is not zero. Having zeroed the output, the criterion of choice becomes crucial for the stability of zero dynamics and, consequently, for the stability of the closed loop system (11.56).

A criterion of choice which yields stable zero dynamics is given by:

$$u(k) = \begin{cases} u \in V_m^{S_1} & \text{if } (v, u) \in S_1 \\ u \in V_M^{S_2} & \text{if } (v, u) \in S_2 \\ u \text{ s.t. it has minimum absolute value} & \text{if } v \in [-\hat{T}, \hat{T}] \end{cases} \tag{11.57}$$

This choice does not guarantee the fastest convergence to the invariant interval and other choices may be better in this sense than this control law. The tradeoff between the speed of convergence to the invariant interval and the shape of the transient response is a difficult problem in its own right but very often it is possible to successfully tackle this problem on a case by case basis.

One can notice that in the above control law we used $\hat{T}$ instead of T. As we have already pointed out, it is very easy to obtain $\hat{T}$ whereas T requires sometimes intensive calculations. Since the used value determines the diameter of the invariant interval, the control law with $\hat{T}$ may not be possible to implement. The difference between the two control laws is large when the exponents of $u(k)$ and $u(k-1)$ are high. Then, working with poor bounds on roots, such as the one that we have used, may yield the estimate $\hat{T}$ which is much larger than T. Computing exact roots, on the other hand, yields a smaller size of the invariant interval.

Blocks in which we need to check whether $(y(k), u(k-1))$ belong to W_I or V_I are equivalent to testing whether a finite number of polynomials which define W_I and V_I are zero when evaluated at $(y(k), u(k-1))$.

11.4.4 Case Study 3: a Fan and Radiator System

A heat exchanger, which consists of a radiator and a fan, was studied in [21] and a model of the system was identified. Heated water is passed through the radiator and the fan blows air across it (see Figure 11.3). The water circulation system consists of a pump and a heater tank. The control objective is to control the temperature drop across the radiator together with the air flow rate across it by adjusting the inputs to the heater and the fan. This is a two-input two-output system and its block diagram is given in Figure 11.4. The subsystems G_{12} and G_{22} in Figure 11.4 are shown to be linear [21] whereas the subsystem G_{11} is nonlinear.

A detailed description of the identification procedure for the subsystem G_{11} can be found in [21]. The following NARMAX model is obtained:

$$y(k+1) \quad = \quad 2.301 + 0.9173y(k) + 0.449u(k) + 0.04557u(k-1) - 0.01889y^2(k)$$

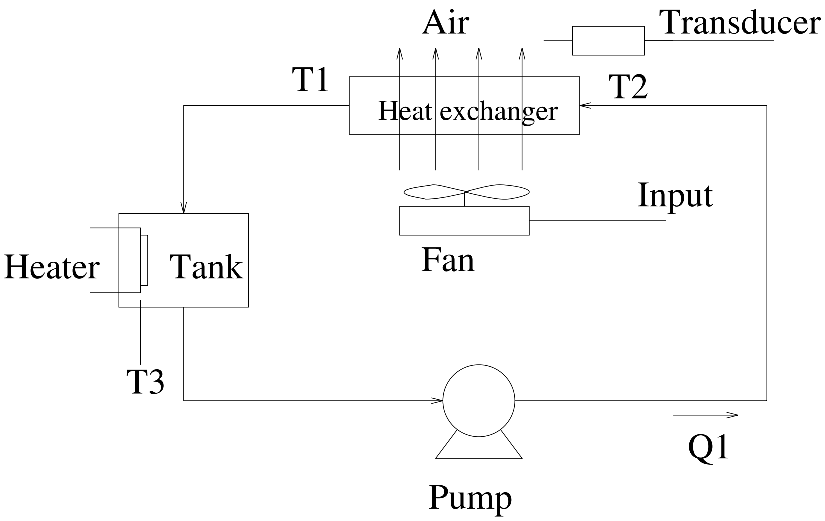

Figure 11.3: A heat exchanger: radiator and fan.

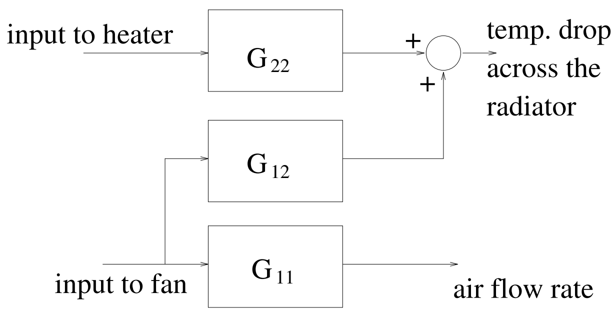

Figure 11.4: Block diagram of the system.

$$-0.00999u^2(k) - 0.002099y^2(k)\,u(k) - 0.002434u^3(k) + e(k+1)$$
$$-0.004e(k) + 0.038e(k-1) + 0.2745e(k-2) + 0.1037e(k-3) \quad (11.58)$$

We investigate output dead-beat control of the disturbance free system:

$$\begin{aligned} y(k+1) &= 2.301 + 0.9173y(k) + 0.449u(k) + 0.04557u(k-1) - 0.01889y^2(k) \\ &\quad - 0.00999u^2(k) - 0.002099y^2(k)\,u(k) - 0.002434u^3(k) \end{aligned} \quad (11.59)$$

Notice that the system (11.59) is odd and of the form (11.54). Moreover, Assumption 11.2 is satisfied for any set point $y{=}y^*$ since the equation

$$y^*{=}f(y^*, v, u)$$

has a real solution u, $\forall v, y^* \in \mathbb{R}$.

The critical variety is $V_C{=}\emptyset$ and hence the system is one step output dead-beat controllable to any fixed set-point y^* (see Chapter 6). Assume that the set point is $y^*{=}10$ and consider the

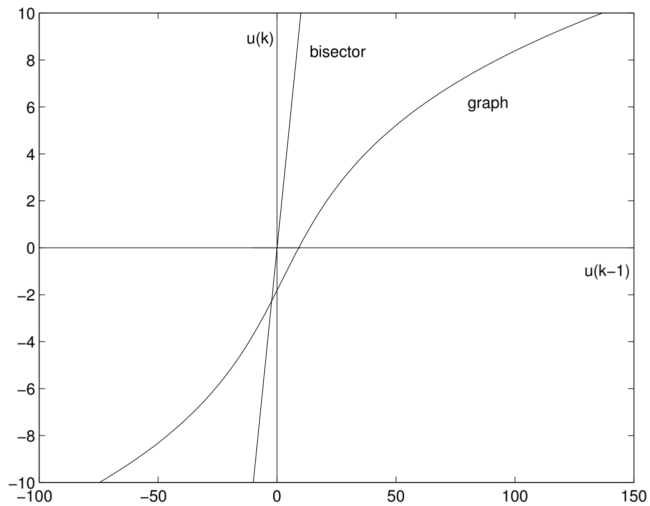

Figure 11.5: Graph of "zero" dynamics.

implicit dynamics defined by

$$0 = -0.415 + 0.04557u(k-1) - 0.2391u(k) - 0.00999u^2(k) - 0.002434u^3(k) \tag{11.60}$$

Since the implicit dynamics (11.60) are very simple we do not have to use the algebraic test from this chapter to check their stability. Indeed, the graph of the variety V_z is drawn in Figure 11.5 and a direct application of Proposition 11.1 shows that the system is point-minimum phase since the point $u(k) = -2.2632$ is a globally asymptotically stable equilibrium. However, it is not difficult to check that the the system is set-minimum phase by using the algebraic test that we presented. A minimum-time output dead-beat controller with stable zero dynamics for a set point y^* is presented in Figure 11.6. The dead-beat controller transfers the output to the set-point y^* in one step.

11.5 Conclusion

In this chapter we presented some results on the problem of stability of zero dynamics and minimum phase polynomial systems. We revisited the definitions of minimum phase systems with the aim of showing that they need to be changed (generalised) in order to be well defined for general nonlinear systems. The fact that we have choice over many controllers when I-O linearising the system had not been studied in the literature. This introduces design procedures into the problem of stability of zero dynamics and in a sense we are talking about "stabilisability of zero dynamics".

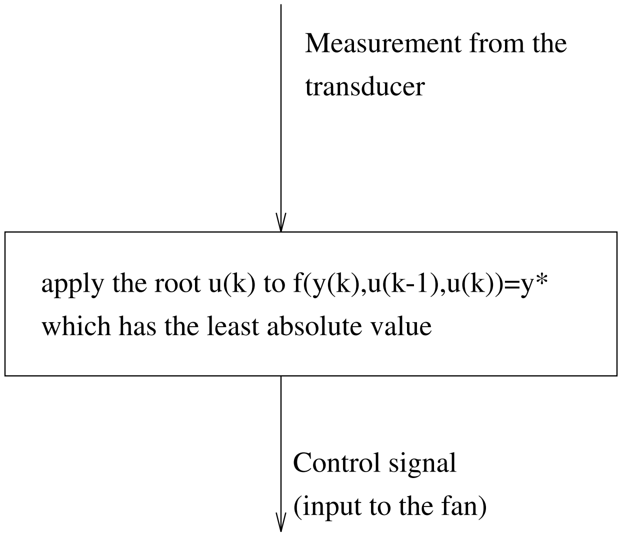

Figure 11.6: The minimum-time output dead-beat controller with stable "zero" dynamics.

We proposed QEPCAD to test different minimum phase properties. We point out that our intention was just to illustrate how it is possible to use this tool (QEPCAD) and we have not presented the most comprehensive or the most general solutions. The use of QEPCAD for stability investigation can in our opinion pave the way for the construction of Lyapunov functions for classes of stable autonomous polynomial systems, which is one of the fundamental questions in nonlinear control theory. It seems that there exists a strong motivation for strengthening the converse Lyapunov theorems [79] by identifying classes of polynomial systems which have polynomial (for example quadratic or quartic) Lyapunov functions. QEPCAD is then the tool which can be used to *compute* Lyapunov functions. Moreover, QEPCAD can be used for the estimation of domains of attraction and hence is a very important tool in analysis and synthesis. Furthermore, it is straightforward to modify the methods that we presented to deal with some other control problems, such as stabilisability of polynomial systems.

Moreover, we showed that stability properties of some classes of polynomial systems can be checked by definition. An interesting question arises:

Is it computationally less complex to check stability by constructing Lyapunov functions or by definition?

At this moment, it seems that the answer to the above question would depend on the class of systems that we are considering. We emphasize that one can easily construct examples for which testing stability by definition is easier to compute! This is a rather unexpected result which sheds a new light on the stability problem.

In the second part of the chapter we provided necessary and sufficient conditions for set-minimum phase property of systems that have scalar implicit zero dynamics. The conditions are not trivial to check despite a seemingly simple structure. This indicates that even if we found analytic/explicit conditions for minimum phaseness of higher order zero dynamics they might be very difficult to check. Bearing this in mind, the algorithmic approach seems to be more natural to use in this setting.

Chapter 12

Conclusions and Further Research

12.1 Conclusions

This thesis has been concerned with dead-beat controllability, control and stability issues for polynomial systems. Here we want to reconsider our contributions and plan a way ahead. There is still a lot of work to be done in the area of polynomial systems theory.

One of the main features of our results is that they can be classified into algorithmic and analytic. We presented **algorithmic** tests for deciding dead-beat controllability and stability. These algorithmic results are applicable to substantial classes of polynomial systems but suffer in general from the computational complexity curse. An important engineering feature of our algorithms is that they naturally lead to a design/implementation of a (dead-beat) control law. To complement the algorithmic work, we also derived a number of **analytic results**, which by necessity only apply to specific subclasses of polynomial systems. In the instances when we were able to derive analytic results our algorithmic approach does also apply, hence providing us with both insight and computational means to understand the problem in full detail.

It is our opinion that analytic and algorithmic methods are complementary. Moreover, despite the computational complexity curse, we have found that the generically applicable decision algebra/quantifier elimination package QEPCAD, can feasibly solve a number of interesting control theoretic and control design problems in the area of general polynomial systems. From an engineering perspective this is very important.

At the core of **the algorithmic approach** which we used are symbolic computation packages - QEPCAD and the Gröbner basis method. Several algorithms for testing different dead-beat controllability properties and/or design of dead-beat controllers have been presented in the thesis.

We give below a final account of advantages and shortcomings of this approach.

Positive aspects:

1. This thesis contains results that can be regarded as a **new** *symbolic computation approach to deciding controllability/stability* for very large classes of polynomial discrete-time systems. Although the basic idea of our approach is simple and can be found in the earliest works on controllability of linear systems, some recently discovered symbolic computation algorithms were used for this purpose. Moreover, we are not aware of any results which take a similar algorithmic approach in the investigation of stability of polynomial systems.

2. QEPCAD allows us to state dead-beat controllability tests for generic polynomial systems in a unified way.

3. Minimum-time dead-beat controllers can be designed systematically using this method. Also, even if we do not opt for minimum-time dead-beat control, QEPCAD still may provide us with information (sets S_k in Chapter 3) which is invaluable in a controller design.

4. Results of Sontag [166] strongly indicate that the algorithmic approach seems to be more natural when formulating controllability tests for larger classes of systems since analytic controllability conditions, even if obtained, would not be easy to check (it is an NP-hard problem).

5. The algorithmic approach indicated that a classification of polynomial systems according to the computational complexity of their controllability problem seems to be more natural in this framework as opposed to the classification based on the structure of system (linear, bilinear, Wiener-Hammerstein, etc.). We summarise in Table 12.1 the results on the complexity of dead-beat controllability tests and controller design methods which we obtained. For example, in Chapter 5 (see Table 12.1) we obtained for scalar polynomial systems that both the dead-beat controllability test and dead-beat controller are generically finitely computable.

6. The power of the approach based on QEPCAD is that we can easily state controllability tests for more general problems, such as controllability of polynomial systems with constraints on controls and/or states and MIMO polynomial systems.

Chapter No.	Controllability conditions		Controllers	
	output	state	output	state
3	not finitely computable in general	not finitely computable in general	not finitely computable in general	not finitely computable in general
4	not finitely computable in general	-	not finitely computable in general	-
5 (odd)	-	finitely computable	-	finitely computable
5 (even)	-	generically finitely computable	-	generically finitely computable
6	finitely computable	-	finitely computable	-
7	-	finitely computable [50]	-	finitely computable
8	-	finitely computable	-	finitely computable

Table 12.1: Summary of results on complexity of dead-beat controllability tests

Negative aspects:

1. The computational complexity of the proposed tests may be formidable in certain situations (see Appendix B for more information on the complexity of the used algorithms). QEPCAD is still being perfected and for certain situations more efficient algorithms can be exploited. In particular, several reported QE algorithms for classes of problems, such as linear [101] or quadratic [173] QE, are substantially better and can be applied to "large scale" problems. For similar results refer also to [112, 83]. Nevertheless, the curse of computational complexity is still unsurmountable in general and this again indicates that some fundamental problems in control theory probably hinge on new computer technologies which might enable us to compute more efficiently. We emphasize that we have solved the dead-beat control problem for polynomial systems but there are many relevant examples that are not computable. We illustrated how it is possible to use some structural assumptions and analytic results in order to reduce the required computations (for instance, in Chapters 5 and 6). In order to reduce computations we use the Gröbner basis approach which behaves much better with respect to the number of variables. This line of reasoning is in the spirit of [80].

2. We have already indicated that the algorithmic approach does not reveal any important structural properties of the system due to which we may lose controllability, which is not desirable.

The analytic approach was very often used in the thesis in cases where we could simplify the controllability tests and reduce the computational complexity of the proposed algorithms. In doing so we obtained some very interesting insights into the underlying phenomena that may cause loss of dead-beat controllability for classes of polynomial systems. The simplifying assumptions, which we imposed on the structure of general polynomial systems in order to obtain the analytical results, lead to investigation of several subclasses of systems. We can group them into three categories:

1. odd systems (Chapters 4, 5 and 6)
2. scalar based polynomial systems (Chapter 5)
3. Hammerstein systems (Chapters 7 and 8)
4. Interconnected polynomial systems (Chapter 9)

We summarise below concisely the important analytic results that we have obtained for each of the above categories.

First, we note that we discussed different forms of invariant sets when dealing with *odd polynomial systems*. In Chapter 5 we considered scalar odd polynomial systems. Scalar odd polynomial systems exhibit a number of important phenomena which could be generalised to higher order odd polynomial systems. Indeed, it was noticed that for scalar odd polynomial systems dead-beat controllability depends on the properties of *invariant sets* of *the control independent set*, on which the system is running in an open loop mode without our ability to affect the behaviour by the control signal. In particular the union of all invariant sets, which is called *the maximal invariant set*, is important. Moreover, it was noticed that the equilibria and periodic points insensitive to control, which are subsets of the maximal invariant set, exhibit a kind of *strong invariance* and the properties of these sets alone determine whether the system is dead-beat controllable. The odd systems considered in Chapter 6 were proved to have another interesting property. Namely, the invariant sets of the *critical variety*, which is usually a larger set than the control independent set, are important for output dead-beat controllability. Nevertheless, the properties of *strongly invariant subsets* of the maximal invariant set represent again a generalisation of results for scalar odd polynomial systems. In Chapter 4, we showed how it is possible to compute the maximal invariant set for the most general class of odd polynomial systems that we considered by using the Gröbner basis method. However, the method must be in general complemented with the QEPCAD method in order to test output dead-beat controllability. Another interesting analytic result for odd systems is the output dead-beat controllability test presented in Chapter 6. In the case of second order systems, the test requires checking whether a set of polynomial divisions is satisfied or not and it does not resemble any of the known analytic results that we are aware of.

Second, scalar polynomial system have the structure which allowed us to solve the dead-beat problem for even systems as well. The test can be based on QEPCAD but we presented another method which can be implemented in Maple. We proved that we can decide whether a system is dead-beat controllable in a generic sense. This is a surprising result which probably does not generalise to higher order systems. Results on dead-beat controllability of scalar polynomial systems may be used to understand the controllability problem for higher order systems with special structure.

Both, simple and generalised Hammerstein systems were investigated in Chapters 7 and 8. The structure of these systems is very close to linear, which allows us to find very simple state and output dead-beat controllability tests. The tests hinge on the results on complete controllability of linear systems with positive controls that existed in literature [50]. Our main contribution in Chapter 7 is a number of non-minimum and minimum-time state and output dead-beat controllers that we design without resorting to QEPCAD. The results are supported by simulations which indicate that the class of proposed controllers seems to be natural to use in this setting. Moreover, linear QE methods of [101] can be efficiently used to design a number of non-minimum and minimum-time dead-beat controllers for simple Hammerstein systems.

The special structure of generalised Hammerstein systems in Chapter 8 allows us to prove a very simple state dead-beat controllability test. We emphasize that Hammerstein systems show very good properties when using the QEPCAD algorithm to design dead-beat controllers. Moreover, the quadratic QE methods presented in [83, 173] show that a dead-beat controller design, using the methodology which we presented in Chapter 3, is feasible for large scale problems. We emphasize that the result on dead-beat controllability of generalised Hammerstein systems was unexpected. Indeed, in the case of a parallel connection of two linear systems, controllability of subsystems **does not guarantee always** the controllability of the overall system. For instance, if we take a parallel connection of a controllable systems with its exact copy, the overall system is not controllable. We can regard generalised Hammerstein systems as a parallel connection of a linear systems and a simple Hammerstein system with a quadratic input nonlinearity. Our result states that controllability of the subsystems **always guarantees** controllability of the parallel connection.

The topology of the interconnection of some structured systems was investigated in Chapter 9 with the aim of reducing the required computations in the controllability tests. Three large classes of structured polynomial systems and a number of their subclasses were identified to have a structure which may help us reduce the computational complexity of deciding the dead-beat problems. More complicated structured systems can be tackled in a similar fashion. The triangular structures of these systems allows us to use a kind of the "backstepping" or "forwarding" procedures in testing controllability and design of dead-beat controllers for these systems.

In Chapter 11 we showed that QEPCAD can be used to decide different minimum phase properties of classes of polynomial systems. It appears that QEPCAD can be used to pave the way towards the computation of Lyapunov functions for classes of polynomial systems. Our results illustrate how QEPCAD can be used to check stability of classes of autonomous polynomial systems in a rather unexpected way (by definition). Indeed, due to the fact that new QE tools, such as QEPCAD, can be used to compute the sets presented in Chapter 11, we are able for the first time to view the stability and stabilisability problems for polynomial systems from a completely new perspective.

Last but not least, we would like to emphasize that a thorough research of the existing literature on applications of polynomial systems has been carried out. A number of examples were investigated using the developed techniques and a long list of applications of polynomial systems is included in Appendix A.

12.2 Further Research

There are several different ways in which the results of this thesis can be extended or pursued further. We divide them into the following main areas:

1. Tests for dead-beat controllability/stabilisability for polynomial systems.

2. Design and implementation of dead-beat controllers, robustness issues.

3. Stability questions for polynomial systems.

4. Mathematical tools.

5. Dead-beat control for non-polynomial systems.

6. Other control laws.

12.2.1 Conditions for Dead-Beat Controllability/Stabilisability for Polynomial Systems

Dead-beat controllability tests based on QEPCAD (as stated in Chapters 3 and 4) are too computationally complex to be used in general and large reductions in computations are needed in order to make the tests more feasible to use. In our opinion, the most important question that needs to be addressed in future is that of reducing the required computations by exploiting some analytic results. The reductions in computations can be achieved in two ways. First, a systematic investigation of some classes of polynomial systems, similar to odd polynomial systems, may yield similar tests to the ones presented in Chapter 6 (finitely computable). Second, a completely different approach may be investigated. Nevertheless, we think that a QE algorithm, similar to QEPCAD, would necessarily be at the core of any such algorithmic controllability test.

A number of important classes of systems with simpler structure may be investigated and simpler controllability (non-algorithmic) tests obtained. There are very few results in this direction and we have already referred to some of them (a class of SISO bilinear systems [48] and simple Hammerstein systems [50]). Output dead-beat controllability offers even more possibilities since we are not aware of any results, except the ones presented in the thesis, which give output controllability conditions for classes of polynomial systems.

It is our opinion that for classes of polynomial block oriented models, such as classes of Wiener-Hammerstein systems, we may develop dead-beat controllability tests that are simple and finitely computable. Also their parallel and/or series connections may be regarded as building blocks of more complex state affine polynomial systems and in this respect the characterisation of controllability properties for these systems may lead to more general results. For instance, we conjecture that a parallel connection of a simple Hammerstein system and a linear system is dead-beat controllable if the subsystems are dead-beat controllable. This result was already verified in the case of quadratic input polynomial for the simple Hammerstein subsystem (the parallel connection is then a generalised Hammerstein system considered in Chapter 8). We have already obtained some results in this direction but a complete analysis is lacking.

An avenue that would be interesting to explore is other controllability notions, such as point-to-point or complete controllability. In particular, controllability of polynomial systems with saturating controls is practically very important and should be addressed in future. We believe that for some classes of polynomial systems it is possible to derive tests for controllability with bounded controls similar to linear systems [174] and bilinear systems [170].

12.2.2 Design and Implementation of Dead-Beat Controllers, Robustness Issues

Design methodologies for dead-beat controllers when any of the assumptions in Chapter 2 are relaxed gives rise to several subproblems, which are crucial for implementation. It would be very important to address these issues in a systematic way.

The design of dead-beat controllers for polynomial systems with saturations is one of the most important issues that needs to be addressed. Results of Chapter 3 can be used in a straightforward manner to include bounds on controls and states. However, apart from the linear case [174], there are no analytical controllability results in the literature, which analyse saturation.

An important class of systems whose controllability properties are well understood but for which there is no design strategies for the design of dead-beat controllers is the class of linear MIMO systems whose controls belong to arbitrary convex sets [52, 55, 56]. Results on dead-beat controllers for linear systems with bounded controls [174] and positive controls [145] could probably be used as a basis for the design of controllers for these more general systems.

Sensitivity of the proposed control laws to the effects of noise and structure and parametric uncertainties is equally important for a good design. We believe that some classes of polynomial systems, such as classes of bilinear or Hammerstein systems, could be treated in an analytic fashion. Notice that for parametric uncertainties QEPCAD may in principle be used to analyse robustness of the proposed algorithms. However, computational requirements are much larger in this case since the uncertainties would have to be regarded as new variables (old variables are the controls and states) in the input polynomials.

We indicated in the introduction that polynomial models can be obtained when approximating a sampled continuous time polynomial system by its Euler or higher order approximation. The question arises whether it is possible to obtain good closed loop behaviour if we apply the controllers presented in this thesis (which are computed for the approximate discrete model) to the sampled system. The simulation study of a bioreactor presented in Chapter 10 shows that in certain situations this method may produce well behaved control schemes. More explicit conditions and/or guidelines would be highly desirable.

In view of the above comments, the continuity of the obtained control laws is very important since it alone guarantees a kind of robustness to structural/parametric uncertainties. Therefore, a procedure which would be based on QEPCAD and which would produce continuous (not necessarily minimum-time) dead-beat control laws seems to be an important practical question that could be investigated in future.

Finally, it would be very interesting to implement some of the presented control laws to real plants. We would probably have to modify the controller on a case-by-case basis. All of the above given issues would influence the performance of the controller and most of them are still open problems in control theory. However, in a particular situation we might probably use a rule of thumb to obtain an implementable control law.

12.2.3 Stability Questions for Polynomial Systems

The results that we presented on the use of QEPCAD in tackling the problem of stability of zero dynamics and stability of autonomous polynomial systems raise several interesting issues that could be addressed in future.

We again emphasize that QEPCAD can be used to compute (or to check the existence of) polynomial Lyapunov functions that belong to a certain class, such as quadratic polynomial functions. Notice here that this is not equivalent to checking stability of the linearisation. It is immediately clear that for systems whose linearised system does not have any poles on the unit circle we can use the class of quadratic Lyapunov functions to establish local stability. In this sense we can use QEPCAD to construct (compute) Lyapunov functions for polynomial systems. We strongly believe that QEPCAD is the tool which would prove instrumental in obtaining systematic methods for computing Lyapunov functions for polynomial systems, which is one of the most important problems in control theory.

Furthermore, there seems to exist a strong motivation for strengthening converse Lyapunov theorems in the following way. We can pose the following question:

Can we identify classes of polynomial systems for which, if asymptotically stable, there exist **polynomial Lyapunov functions of certain form***?*

In this way, we can use the **known** class of polynomial functions (e.g. quartic polynomials) and check using QEPCAD whether there exist a Lyapunov function that belongs to this class of functions. Notice, that if this was possible, by using the Lyapunov theory and QEPCAD, we could obtain algorithms that stop in finite time and which produce Lyapunov functions for polynomial systems. Hence, we would obtain a tool to check necessary and sufficient conditions for stability of classes of polynomial systems. Moreover, for a given Lyapunov function we can find domains of attraction and/or stability by using the same tool.

We presented several test for stability which are either based on the definition of asymptotic stability or on Lyapunov theory. Since we use QEPCAD to test stability, it seems very important to look at the following question: which formulation yields computationally cheaper tests? At this stage, it seems that checking asymptotic stability by definition may be computationally cheaper than computing Lyapunov functions for certain classes of problems. Moreover, it seems that stronger stability properties such as exponential stability are easier to check by definition whereas stability without attractivity is impossible to be checked in this way (we need to check infinitely many conditions). More in depth analysis of these questions appears to be fundamental since it would definitely cast completely new light on applications of Lyapunov's second method.

12.2.4 Mathematical Tools

It would be very important to design a toolbox which could be used in solving the above discussed dead-beat controllability and stability problems for polynomial systems. QEPCAD and the Gröbner basis method would be the core of any such toolbox.

Moreover, we emphasize that a number of other important problems can also be solved using QEPCAD. We mention just a few of them: stabilisation with output feedback of linear systems, pole placement with static periodic output feedback, motion planning, robust control, etc. We note here that also a number of inverse eigenvalue problems can be in principle solved using QEPCAD. So the toolbox might be applicable to a much wider range of control problems than the ones considered in this thesis.

Furthermore, it may be attempted to incorporate some other algorithms, such as Grigor'ev algorithm [73], since they may be less computationally expensive for certain problems (we are not aware whether the Grigor'ev algorithm has been implemented on the computer). A number of sub-algorithms that are used in QEPCAD can be modified to suit special classes of problems, such as solving polynomial strict inequalities. This and a number of other possible improvements of QEPCAD are discussed in some detail in [33].

However, it is still difficult to see when it will be possible to undertake the design of such a toolbox since QEPCAD is still not available commercially. We conclude by saying that we believe that this ongoing area of research will prove instrumental in attacking and solving a number relevant control problems in years to come. However, at this moment it is not possible to anticipate the level of the future impact of symbolic computation packages for elementary real algebra and QE on control theory without being speculative, due to the rapid progress of the area.

12.2.5 Dead-Beat Controllability of Non-Polynomial Systems

Although polynomial systems may be used to model a large number of plants, they can not model all systems of interest. Consequently, controllability properties of more general classes of systems can be investigated. The next step might be the investigation of rational or analytic systems. In particular, it would be interesting to investigate systems described by neural network models since they are known to be very good approximators and can model a large number of nonlinear systems. The question of whether the controllability of a polynomial approximation of an analytic system implies controllability of the original system also appears to be important since our results could be applied in a straightforward manner.

12.2.6 Other Control Laws

We have already indicated in the introduction the good and bad aspects of using dead-beat controllers. The main shortcomings of this control strategy is that it may not be robust and sometimes large magnitudes of control signals are required. If the designed dead-beat controller does not have these undesirable properties, it is certainly a good and simple option for the control engineer. However, there is no guarantee that the closed loop system will possess these good properties. Hence, there exists a strong motivation for considering other control paradigms for the classes of polynomial systems. For example, optimal control, receding horizon control, minimum variance control, predictive control and/or adaptive control could be considered and their relation to dead-beat control investigated. It was proved that dead-beat control has strong connections with solutions of singular Riccati equations [94] and minimum variance control [47]. We believe that similar relationships can be established for classes of polynomial systems. Furthermore, we think that optimal control via dynamic programming may provide a number of interesting results and control strategies for polynomial systems.

Most of the above mentioned control strategies have been introduced in the context of different classes of nonlinear systems but we believe that by revisiting the controller design for polynomial systems we may obtain stronger results/improve performance by exploiting the polynomial structure of the system. In particular, we think that classes of simpler polynomial systems, such as simple Hammerstein systems, might even allow for solutions in closed form.

Part III

Appendices

Appendix A

Polynomial Models

A.1 Applications of Polynomial Models

We give below several examples of applications of polynomial models with references. It is not our intention to give a comprehensive survey of applications but just to show the versatility of different processes that fall into this category. Also, a list of some applications of polynomial systems with references is presented in Table A.1. The systems followed by (P) have polynomial prediction models, which are not considered in this thesis.

Example A.1 A liquid level system which consists of interconnected tanks was investigated and its model identified in [21]. One of the tanks has a conical cross section and induces nonlinearities. The input volume flow rate is the input u to the systems and the level of liquid in the conical tank is the output y (for more detailed analysis see [21] and references therein). The model is derived for the sampling interval of T=9.6sec:

$$\begin{aligned} y(k) \;=\;& 0.43y(k-1)+0.681y(k-2)-0.149y(k-3)+0.396u(k-1) \\ &+0.014u(k-2)-0.071u(k-3)-0.351y(k-1)\,u(k-1)-0.03y^2(k-2) \\ &-0.135y(k-2)\,u(k-2)-0.027y^3(k-2)-0.108y^2(k-2)\,u(k-2) \\ &-0.099u^3(k-2)+e(k)+0.344e(k-1)-0.201e(k-2) \end{aligned}$$

where $e(k)$ is the output measurement noise. This is an example of a polynomial NARMAX model.

Example A.2 A binary distillation column was identified in [18]. This column is fed with a mixture of isobutane and normalbutane which have the same composition C_4H_{10} but different

System	Reference
MODELLING	
nuclear fission	[119]
blood pressures	[113]
cell dynamics in the immune system	[119]
compound interest	[70]
neural network	[70]
population growth	[70]
investment firm policy (P)	[13]
nonlinear channel modelling	[115]
signal detector	[125]
applications in communications and radar design	[125]
applications in image processing	[163]
neutron kinetics, etc.	[107]
IDENTIFICATION	
column type grain dryer	[107]
cement mill (grinding mills)	[104]
heat exchanger	[76]
liquid level system	[21]
binary distillation column	[18]
turbo generator set	[106]
diesel generator	[23]
heat exchanger (radiator and fan)	[21]
blast furnace (P)	[106]
distillation column (reflux flow vs. top temperature)	[77]
distillation column (reflux flow and vapour flow vs. two temperatures)	[168]
flexible manipulator arm	[110]
continuous stirred reactors	[114]
super heater in a thermal power plant	[117]
steam exchanger in a nuclear power plant	[117]
wave propagation in a hydraulic power plant	[117]
cooling water circulation of a thermal power plant	[11]
effects of drugs	[26, 42]
power system control	[120]
aircraft control	[120]
vehicle dynamics	[175]
peak flood (P)	[177]
environmental modelling (P)	[46]
air-pollution modelling (P)	[176]
river flow modelling (P)	[85]
human static and dynamic strengths (P)	[118]
EULER DISCRETISATION	
d.c. motor	[119]
bacterial growth systems, etc.	[44]

Table A.1: Some applications of polynomial models structures and boiling points. Since the isobutane is more volatile it is possible to extract it from the top of the column; the normalbutane is present in the bottom flow. The input of the system u is the percentage of the isobutane in the inlet flow whereas the percentage of isobutane in the output flow is the output. The identified I-O polynomial model is given below:

$$\begin{aligned} y(k+3) &= 0.898y(k+2) + 0.248u(k+2) - 0.106 \cdot 10^{-2} u(k+1)\, y(k) \\ &\quad + 0.173 \cdot 10^{-5} u^2(k)\, u(k+1)\, y(k+3) - 0.159 \cdot 10^{-10} u^3(k)\, u^2(k+1)\, u(k+2)\, y(k) \end{aligned}$$

Example A.3 A dog was subjected to a treatment in which a drug (Nitropruside) was infused into the dog's blood to control the blood pressure [42]. The input signal is the input infusion rate in mlh^{-1}. The output is the main arterial pressure of the dog measured in $mm\ Hg$.

The following state affine model was identified in [42]:

$$\begin{aligned} x(k+1) &= [F_0 + u(k)\, F_1 + u^2(k)\, F_2]\, x(k) + u(k)\, G_1 + u^2(k)\, G_2 \\ y(k) &= [H_0 + u(k)\, H_1]\, x(k) \end{aligned}$$

where

$$F_0 = \begin{pmatrix} 0.8088 & 1 & 0.3614 \\ 0.0857 & 0 & -0.296 \\ -0.1692 & 0 & 0.0898 \end{pmatrix}$$

$$F_1 = \begin{pmatrix} 0.0247 & -0.0241 & 0.0049 \\ 0.0105 & 0.0053 & 0.004 \\ -0.0055 & -0.0025 & -0.0012 \end{pmatrix}$$

$$F_2 = \begin{pmatrix} 0.0002 & -0.0001 & 0 \\ -0.0002 & 0.0002 & 0 \\ -0.0002 & 0.0001 & 0 \end{pmatrix}$$

$$G_1 = (0\ 1\ 0)^T \quad G_2 = (0.0151\ -0.0289\ 0.0085)^T$$

$$H_0 = (-0.1024\ 0.019\ -0.0539) \quad H_1 = (-0.0031\ -0.002\ -0.0004)$$

A.2 Classes of Polynomial Models Used in the Literature

The purpose of this section is to give the mathematical descriptions for classes of polynomial models that are often used in the literature and referred to in the thesis. The list is by no means comprehensive. In all cases $x \in \mathbb{R}^n, y \in \mathbb{R}, u \in \mathbb{R}$.

Simple Hammerstein model [75, 76]:

$$\begin{aligned} x(k+1) &= Ax(k) + b(t_0 + t_1 u(k) + t_2 u^2(k)) \\ y(k) &= cx(k) + d(t_0 + t_1 u(k) + t_2 u^2(k)) \end{aligned}$$

Generalised Hammerstein model [75, 76, 104]:

$$\begin{aligned} x(k+1) &= \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ b_2 \end{pmatrix} u^2(k) \\ y(k) &= (c_1 \;\; c_2)\, x(k) + t_0 + t_1 u(k) + t_2 u^2(k) \end{aligned}$$

Simple Wiener model [75, 76]:

$$\begin{aligned} x(k+1) &= Ax(k) + bu(k) \\ y(k) &= t_0 + t_1(cx(k) + tu(k)) + t_2(cx(k) + t_2 u(k))^2 \end{aligned}$$

Generalised Wiener model [75, 76]:

$$\begin{aligned} x(k+1) &= \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u(k) \\ y(k) &= t_0 + c_1 x(k) + t_1 u(k) + (c_2 x(k) + t_2 u(k))^2 \end{aligned}$$

Wiener-Hammerstein cascade model [75, 76]:

$$\begin{aligned} x(k+1) &= \begin{pmatrix} A_1 & 0 \\ t_1 b_2 c_2 & A_2 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ t_1 p_1 b_2 \end{pmatrix} u(k) + t_0 \begin{pmatrix} 0 \\ b_2 \end{pmatrix} + t_2 \begin{pmatrix} 0 \\ b_2 \end{pmatrix} (c_1^* x(k) + p_1)^2 \\ y(k) &= (p_2 t_1 c_1 \;\; c_2)\, x(k) + p_2 t_0 + t_1 p_1 p_2 u(k) + p_2 t_2 (c_1^* x(k) + p_1 u(k))^2 \end{aligned}$$

For more general block oriented models, such as generalised and extended Wiener-Hammerstein models, we refer to [75, 76].

Homogeneous bilinear systems [48, 70]:

$$x(k+1) = (A + u(k)B)x(k)$$

Inhomogeneous bilinear systems [49, 107]:

$$x(k+1) = (A + u(k)B)x(k) + cu(k)$$

Input-output (NARMAX) polynomial systems [76, 21, 184]:

$$y(k+1) = F(y(k), \ldots, y(k-t), u(k-s), \ldots, u(k-1), u(k)), \text{ where}$$

$$F(y_0, y_1, \ldots, y_t, u_s, \ldots, u_0) \in \mathbb{R}[y_0, y_1, \ldots, y_t, u_s, \ldots, u_0]$$

State affine polynomial models [163, 117]:

$$\begin{aligned} x(k+1) &= (A_0 + u(k)A_1 + \ldots + u^n(k)A_n)x(k) + b_0 + b_1u(k) + \ldots + b_mu^m(k) \\ y(k) &= h(x(k)), \ h \in \mathbb{R}[x], \ A_i \in \mathbb{R}^{n \times n}, \ b_i \in \mathbb{R}^{n \times 1} \end{aligned}$$

Appendix B

Mathematical Background Material

B.1 Algebraic Geometry

In this section we review some notions from algebra and algebraic geometry and introduce some notation useful to our developments. We use [37] as a main reference for most of the results from algebraic geometry unless otherwise indicated.

We use standard definitions of rings and fields. We work over the field of real numbers which is denoted as $\mathbb{R}$. For computational purposes the field of rational numbers $\mathbb{Q}$ plays an important role. $\mathbb{R}^n$ is a set of all n-tuples of elements of $\mathbb{R}$, where $n \in \mathbb{N}$. The set of integers is denoted by $\mathbb{Z}$. The ring of polynomials in n variables over the real field $\mathbb{R}$ is denoted as $\mathbb{R}[x_1, x_2, \ldots, x_n]$.

Theorem B.1 *Let k be an infinite field, and let $f \in k[x_1, x_2, \ldots, x_n]$. Then f=0 in $k[x_1, \ldots, x_n]$ if and only if $f : k^n \to k$ is the zero function.* $\square$

Let $f_1, f_2, \ldots, f_s$ be polynomials in $\mathbb{R}[x_1, x_2, \ldots, x_n]$. Then we define

$$V(f_1, f_2, \ldots, f_s) = \{(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n : f_i(a_1, a_2, \ldots, a_n) = 0 \text{ for all } 1 \leq \text{i} \leq \text{s}\}.$$

We call $V(f_1, f_2, \ldots, f_s)$ the real algebraic set or real variety defined by the polynomials $f_1, f_2, \ldots, f_s$. Since the defining polynomials of a real variety are often clear from the context, we may denote it simply as V.

Theorem B.2 *If $V, W \subset \mathbb{R}^n$ are real varieties, then so are $V \cup W$ and $V \cap W$.* $\square$

A subset $I \subset \mathbb{R}[x_1, x_2, \ldots, x_n]$ is an ideal if

1. $0 \in I$.

2. If $f,\ g \in I$, then $f+g \in I$.

3. If $f \in I$ and $h \in \mathbb{R}[x_1, \ldots, x_n]$, then $hf \in I$.

Let $f_1, f_2, \ldots, f_s$ be polynomials in $\mathbb{R}[x_1, x_2, \ldots, x_n]$. Then the set $\langle f_1, \ldots, f_s\rangle$ defined as

$$\langle f_1, \ldots, f_s\rangle = \{\sum_{i=1}^{s} h_i f_i : h_1, \ldots, h_s \in \mathbb{R}[x_1, \ldots, x_n]\,\}$$

is called the ideal generated by $f_1, f_2, \ldots, f_s$. The product $J_1 \cdot J_2$ of ideals J_1 and J_2 is defined to be the ideal generated by all polynomials $f \cdot g$ where $f \in J_1$ and $g \in J_2$.

Definition B.1 A real variety $V \subset \mathbb{R}^n$ is **irreducible** if whenever V is written in the form $V = V_1 \cup V_2$, where V_1 and V_2 are real varieties then either $V_1 = V$ or $V_2 = V$. [37, pp. 196]. □

For the relationship between ideals and varieties, or the so called algebra-geometry dictionary, see Chapter 4 of [37]. The following theorems are immediate consequences of the Hilbert basis theorem which says that every ideal $I \in \mathbb{R}[x_1, \ldots, x_n]$ is finitely generated.

Theorem B.3 *Any descending chain of varieties $V_1 \supset V_2 \supset V_3 \supset \ldots$ (ascending chain of ideals $I_1 \subset I_2 \subset I_3 \subset \ldots$) in $\mathbb{R}^n$ ($\mathbb{R}[x_1, \ldots, x_n]$) must stabilise. That is, there exist a positive integer N such that $V_N = V_{N+1} = \ldots$ ($I_N = I_{N+1} = \ldots$).* □

Theorem B.4 *Let $V \subset \mathbb{R}^n$ be a real variety. Then V can be written as a finite union of irreducible varieties:*

$$V = V_1 \cup V_2 \cup \ldots \cup V_m$$

where each V_i is an irreducible variety. □

B.2 Gröbner Bases

Roughly speaking, a Gröbner basis of a set of polynomials is another set of "simpler" polynomials with respect to some simplification rules. One usually specifies the simplification rules (monomial ordering) and it should be noted that different such rules (monomial orderings) produce different Gröbner bases. The first algorithm for the computation of Gröbner bases was discovered by B. Buchberger in 1965 and since then a number of its modifications have been reported [37, 29, 17]. Gröbner bases are an important tool in algebra [29, 37] which can be used in solving the following important problems [37]:

1. The Ideal Description Problem: Does every ideal $I \subset k[x_1, \ldots, x_n]$ have a finite generating set? In other words, can we write $I = \langle f_1, \ldots, f_s \rangle$ for some $f_i \in k[x_1, \ldots, x_n]$?

2. The Ideal Membership Problem: Given $f \in k[x_1, \ldots, x_n]$ and an ideal $I = \langle f_1, \ldots, f_s \rangle$, determine if $f \in I$. Geometrically, this is closely related to the problem of determining whether $V(f_1, \ldots, f_s)$ lies on the variety $V(f)$.

3. The Problem of Solving Polynomial Equations: Find all common solutions in k^n of a system of polynomial equations:

$$f_1(x_1, \ldots, x_n) = \ldots = f_s(x_1, \ldots, x_n) = 0.$$

4. The Implicitisation Problem: Let V be a subset of k^n given parametrically as:

$$\begin{array}{rcl} x_1 & = & g_1(t_1, \ldots, t_m) \\ \vdots & & \vdots \\ x_n & = & g_n(t_1, \ldots, t_m) \end{array}$$

   If the g_j are polynomials or rational functions in the variables t_i, then V will be an affine variety or a part of one. Find a system of polynomial equations (in the x_k) that define the variety.

We give below formal definitions and properties of the Gröbner bases.

Definition B.2 A monomial ordering on $k[x_1, \ldots, x_n]$ is any relation $\succ$ on $\mathbb{N}^n$, or equivalently, any relation on the set of monomials $x^\alpha, \alpha \in \mathbb{N}^n$, satisfying:

1. $\succ$ is a total (or linear) ordering on $\mathbb{N}^n$. That is, for any $\alpha, \beta \in \mathbb{N}^n$ only one of the expressions $\alpha \succ \beta, \alpha = \beta, \alpha \prec \beta$ is true.

2. If $\alpha \succ \beta$ and $\gamma \in \mathbb{N}^n$, then $\alpha + \gamma \succ \beta + \gamma$.

3. $\succ$ is a well-ordering on $\mathbb{N}^n$. This means that every nonempty subset of $\mathbb{N}^n$ has a smallest element under $\succ$. □

Many different orderings can be defined, but for our developments the ordering does not play a crucial role, except for the fact that all computations should be carried out with only one ordering.

Hence, there is no loss of generality if we assume that throughout the thesis we use the so called lexicographic ordering.

Definition B.3 The lexicographic ordering is defined by:

$$\alpha \succ \beta \;\Leftrightarrow\; \exists k : \alpha_k > \beta_k, \forall j < k : \alpha_j = \beta_j$$

□

For example, using lexicographic ordering we can write: $x_1 x_2^2 x_3^4 \succ x_1 x_2^2 x_3^2$ since $(1\,2\,4) \succ (1\,2\,2)$. Also, $x_1 \succ x_2 \succ \ldots \succ x_n$ since $(1\,0\,\ldots\,0) \succ (0\,1\,\ldots\,0) \succ \ldots \succ (0\,0\,\ldots\,1)$.

Let $\alpha \in \mathbb{N}^n$. Consider monomials in $k[x_1, \ldots, x_n]$. The following notation is used:

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}, \alpha = multideg(x^\alpha), |\alpha| = \sum_{i=1}^{n} \alpha_i.$$

Definition B.4 Let $f = \sum_\alpha a_\alpha x^\alpha$ be a non zero polynomial in $k[x_1, x_2, \ldots, x_n]$ and let $\succ$ be a monomial order.

1. The multi-degree (or total degree) of f is $multideg(f) = \max(\alpha \in \mathbb{N}^n : a_\alpha \neq 0)$ (the maximum is taken with respect to $\succ$).

2. The leading coefficient of f is $LC(f) = a_{multideg(f)} \in k$.

3. The leading monomial of f is $LM(f) = x^{multideg(f)}$ (with coefficient 1). The leading term of f is $LT(f) = LC(f) \cdot LM(f)$. □

Definition B.5 Fix a monomial order (see [37]). A finite subset $G = \{g_1, g_2, \ldots, g_t\}$ of an ideal I is said to be a Gröbner basis or standard basis for I if

$$\langle LT(g_1), \ldots, LT(g_t) \rangle = \langle LT(I) \rangle$$

where $LT(g_i)$ is the leading term of g_i and $\langle LT(I) \rangle$ is the ideal generated by the set of leading terms $LT(f_i)$ of polynomials $f_i \in I$. □

Proposition B.1 Let $G = \{g_1, g_2, \ldots, g_t\}$ be a Gröbner basis for an ideal $I \subset \mathbb{R}[x_1, x_2, \ldots, x_n]$ and let $f \in \mathbb{R}[x_1, x_2, \ldots, x_n]$. Then there is a unique $r \in \mathbb{R}[x_1, x_2, \ldots, x_n]$ with the following properties:

1. No term of r is divisible by one of $LT(g_1), \dots, LT(g_t)$.

2. There is $g \in I$ such that $f{=}g + r$. □

Corollary B.1 Let $G{=}\{g_1, g_2, \dots, g_t\}$ be a Gröbner basis for an ideal $I \subset \mathbb{R}[x_1, x_2, \dots, x_n]$ and let $f \in \mathbb{R}[x_1, x_2, \dots, x_n]$. Then $f \in I$ if and only if the remainder on division of f by G is zero. □

Theorem B.5 *Let $I{=}\langle f_1, f_2, \dots, f_s\rangle{\neq}\{0\}$ be a polynomial ideal. Then a Gröbner basis for I can be constructed in a finite number of algebraic computations by an algorithm (the algorithm is given in [37, pg. 89]).* □

Definition B.6 A reduced Gröbner basis for a polynomial I is a Gröbner basis G for I such that :

1. $LC(p) {=}1$ for all $p \in G$.

2. For all $p \in G$, no monomial of p lies in $\langle LT(G - \{p\})\rangle$. □

Proposition B.2 Let $I{\neq}\{0\}$ be a polynomial ideal. Then, for a given monomial ordering, I has a **unique reduced Gröbner basis**. □

From the above given proposition it follows that we can compare two ideals by comparing their **reduced Gröbner bases** for the same monomial ordering.

Theorem B.6 *(The Elimination Theorem [37]) Let $I \subset k[x_1, \dots, x_n]$ be an ideal and let G be a Gröbner basis of I with respect to lexicographic order where $x_1 \succ x_2 \succ \dots \succ x_n$. Then, for every $0 \leq k \leq n$, the set*

$$G_k{=}G \cap k[x_1, \dots, x_n]$$

is a Gröbner basis of the kth elimination ideal I_k. □

To illustrate the use of Gröbner bases consider the following:

Example B.1 A Gröbner basis of the ideal defined by the polynomials:

$$x + y - z^3, x^2 + y - z, -x + y^2 + z$$

with the lexicographic ordering $x \succ y \succ z$ is

$$x + y - z^3, y - z^3 + y^2 + z, 6z - 3z^3 - 2z^4 - z^9 + 4yz, 4z - 8z^3 + 3z^5 + z^{11}$$

Notice that the Gröbner basis has a kind of triangular structure. The last polynomial in the Gröbner basis depends on z only. This is very important if we want to find the solutions of the original set of polynomials. Indeed, solving the polynomial equation in one variable $4z - 8z^3 + 3z^5 + z^{11}$=0 is much easier than polynomials in more variables. When we find all the solutions we can substitute them into polynomial equations that depend on y and z and solve them in y, etc. □

The triangular structure of the Gröbner basis resembles the structure obtained when applying the Gauss elimination algorithm for systems of linear equations. Actually, systems of equations in the reduced echelon form that are obtained by the application of the Gauss algorithm are special cases of the Gröbner bases.

B.2.1 Complexity of Gröbner Basis Constructions

We denote the set of input polynomials to the algorithm by F and its Gröbner basis as G (we use [17] as a main reference for this section). There are n indeterminates (variables) in the polynomials F. We also use the following notation:

$\lvert F\rvert$	the number of polynomials in the set F
maxdeg(F)	the maximal multi (total) degree in the polynomials F
maxsize(F)	the maximal size of the coefficients of the polynomials under given coding
D	the maximal degree of any polynomial occurring during computation
S	the maximal size of the coefficients of any polynomial occurring during computation

It can be shown that D as well as $|G|$ are bounded by recursive functions of n, $|F|$ and maxdeg(F). These functions are independent of the ground field, the monomial ordering and the size of input coefficients. Secondly, the maximal size M of any coefficient appearing in the construction is bounded by a recursive function of n, $|F|$, maxdeg(F) and maxcoeff(F), again independently of the monomial ordering. If all coefficients are represented as rational expressions in the input coefficients, then this bound is independent of the ground field.

The computation time, that is the number of steps, required for a Gröbner basis construction is bounded by a recursive function of n, $|F|$ and maxdeg(F) when an arithmetic operation and an equality test in the ground field and a comparison of terms in the term order are counted as one step each. When computations in the ground field are performed in polynomial time, then for fixed n, $|F|$ and maxdeg(F), the Gröbner basis G can be constructed in polynomial time in maxcoeff(F).

The following bound on the degrees appearing in the Gröbner bases can be found in [17]:

$$\text{maxdeg}(G) \leq f(\text{maxdeg}(F))$$

where f is a polynomial of degree a^n with $a \leq \sqrt{3}$. It was presented in [37] that the Gröbner basis can contain polynomials of multi-degree proportional to $2^{2^{\text{maxdeg}(F)}}$. However, these bonds occur in "worst case" analysis and it can be shown [29] that the polynomials in the reduced Gröbner basis, with probability 1, stay below $d_1 + \ldots + d_l - n + 1$, where d_i are the degrees of the input polynomials.

The Gröbner basis method has better computation time than QEPCAD for problems that we consider (for more explicit computation time bounds of QEPCAD see the next section). In particular, the number of variables in the input polynomials does not hinder the computation of the Gröbner bases as much as it does the computation of CAD. Hence, by reformulating a problem so that the Gröbner basis method can be used, we obtain computationally less expensive tests. This approach is similar to [80].

B.3 Semi-Algebraic Geometry

Results from algebraic geometry are usually valid over algebraically closed fields such as the field of complex numbers $\mathbb{C}$. However, in many situations the existence of a complex solution still does not solve a problem since some variables are constrained to have real values only. For instance, a value of the distance, temperature or concentration, does not have a meaning if it is computed to be a complex number. Therefore, the area of mathematics which deals with real objects often seems to be more natural to use than classical algebraic geometry. This area of mathematics is called real algebraic and/or semi-algebraic geometry (or real algebra). A very good introduction to the material that we need is presented in [92], which we use as the main reference unless otherwise stated.

Similarly to algebraically closed fields in algebra, we introduce *real closed fields* as ground fields in real algebra.

Definition B.7 [92] An ordered field is a field k together with a subset $P \subset k$, the set of positive elements, such that:

1. $0 \notin P$.

2. If $a \in k$, then either $a \in P$, a=0 or $-a \in P$.

3. If $a, b \in P$, then $a + b \in P$ and $ab \in P$, that is P is closed under addition and multiplication. □

Notice that in ordered fields we can introduce ordering: $a > b$ if $a - b \in P$.

Definition B.8 An ordered field k is real closed if:

1. Every positive element of k has a square root in k.

2. Every polynomial $f(x) \in k[x]$ of odd degree has a root in k. □

For example, the set of real numbers $\mathbb{R}$ is a real closed field whereas the set of rational numbers $\mathbb{Q}$ is not since $\sqrt{2} \notin \mathbb{Q}$. Hereafter, it is assumed that the ground field is the field of real numbers $\mathbb{R}$.

Definition B.9 A subset of $\mathbb{R}^n$ is semi-algebraic if it can be constructed from finitely many applications of union, intersection and complementation operations on sets of the form

$$\{x \in \mathbb{R}^n : f(x) \geq 0\}$$

where $f \in \mathbb{R}[x_1, x_2, \ldots, x_n]$. □

For example, the set:

$$\begin{aligned} S &= \{x \in \mathbb{R}^n : x_1^2 + x_2{=}0\} \cap \{x \in \mathbb{R}^n : x_1 - 2x_2 > 0\} \\ &= \{x \in \mathbb{R}^n :(x_1^2 + x_2{=}0) \wedge (x_1 - 2x_2 > 0)\} \end{aligned}$$

is semi-algebraic. In general, we can represent a semi-algebraic set as follows:

$$S{=}\{x \in \mathbb{R}^n : \bigvee_{i=1}^{p} (\bigwedge_{j=1}^{r_i} f_{i,j}(x) \quad m_{i,j} \quad 0)\}$$

where $f_{i,j} \in \mathbb{R}[x_1, \ldots, x_n]$ and $m_{i,j} \in \{>, =\}$ and $\wedge$ and $\vee$ are respectively the "and" and "or" Boolean operators.

Theorem B.7 *(Continuity of roots) [20, pg. 38] Let T be a connected topological space, $a_0(t), \ldots, a_n(t)$ continuous functions: $T \to \mathbb{R}$ such that:*

1. $a_n(t) \neq 0$ ($t \in T$)

2. *the number of complex roots (whose imaginary part is nonzero) of* $P_t(x) = a_0(t) + a_1(t)\,x + \ldots + a_n(t)\,x^n$ *is constant for* $t \in T$.

Then:

1. *If* $a_i(t) \in \mathbb{R}$*, the number of real roots of* $P_t(x)$ *is also constant*

2. *there exist continuous functions* $g_j : T \to \mathbb{R}$ *(*$1 \leq j \leq r$*,* r *is the number of real roots) such that*

   - $g_j(t)$ *is a root of* $P_t(x)$ *(*$1 \leq j \leq r$*)*

   - $g_j(t) \neq g_l(t)$ *,* $\forall t \in T$ *if* $j \neq l$ □

Theorem B.8 *[20, pp. 299-302] Let, for* $1 \leq i \leq n,\ P_i \in C[x_1, x_2, \ldots, x_{n+1}]$ *be a homogeneous polynomial of degree* d_i*. Then if all the solutions of the system* (S) *:* $P_i = 0$ $(1 \leq i \leq n)$ *are non-degenerate, their number is* $d_1 d_2 \ldots d_{n+1}$. □

Theorem B.9 *[20, pg. 19] If we have a polynomial*

$$p(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

and if we denote as α *a real root of* p*, then we have* $|\alpha| < 1 + \sup(|a_i|)$. □

A very important area of real algebraic geometry that we use is the first-order theory of real closed fields. An expression consisting of polynomial equations and inequalities combined with Boolean operators $\wedge$ (and), $\vee$ (or), $\rightarrow$ (implies), $\neg$ (not) where some of the variables are quantified using the quantifiers $\exists$ (there exists) and $\forall$ (for all) constitutes a sentence in the first-order theory of real closed fields. A classical question is: given any sentence in the first-order (or elementary) theory of real closed fields, is it possible to obtain a quantifier-free formulas, which are equivalent to the un-quantified ones. To clarify the terminology let us first consider a few examples.

Suppose that all variables in a sentence are quantified. Deciding whether the sentence is true or not is called *a decision problem*. For example, the sentence

$$(\forall x_1)\,(\forall x_2)\,[x_1^2 + x_2^2 + 1 > 0]$$

is TRUE since the expression $x_1^2 + x_2^2 + 1$ is positive for all **real** values of x_1 and x_2. Consider the following sentence

$$(\exists u)\,[u^2 + 2u + 4{=}0]$$

It is NOT TRUE since the equation $u^2 + 2u + 4{=}0$ does not have any **real** roots. We emphasize that all variables are assumed to be real.

Suppose that some of the variables in a sentence are quantified but some of them are not. In this case, we can talk about **a quantifier elimination problem** which consists of finding an equivalent expression in the unquantified variables only. For example, consider the sentence:

$$(\exists y)\,[(x + y{=}0) \wedge (x^2 - y{=}0)\,]$$

The solution, or a quantifier free formula is $x^2 + x{=}0$. It is clear that the question of quantifier elimination is closely related to many important problems, such as the conditions for the existence of a real root of a univariate polynomial or the existence of a real root of a set of polynomial equations. Observe the generality of the quantifier elimination problem, which includes a number of subproblems that are known to be difficult.

In general we have:

Definition B.10 A formula in the first-order theory of real closed fields is an expression in the variables $x{=}(x_1,\ \ldots,\ x_n)$ of the following type:

$$(Q_1 x_1)\,(Q_2 x_2)\ldots(Q_s x_s)\,[\mathcal{F}(f_1(x)\,,\ \ldots,\ f_r(x)\,)\,]$$

where Q_i is one of the quantifiers $\forall$ or $\exists$, $\mathcal{F}(f_1(x)\,,\ \ldots,\ f_r(x)\,)$ is a quantifier free Boolean formula and $f_j \in \mathbb{R}[x_1, \ldots, x_n]$. □

Some important results from this research area of mathematics are referred to below. The oldest result on the existence of real roots of a univariate polynomial dates back to the beginning of last century and it is due to Sturm.

Theorem B.10 *(Sturm's Theorem) [87, pp.295-299] Let $f(x)$ be a polynomial of positive degree with coefficients in a real closed field $\mathbb{R}$ and let $\{f_0(x)\,, f_1(x)\,, \ldots, f_s(x)\,\}$ be the standard sequence for $f(x)$ defined by:*

$$f_0(x) \quad = \quad f(x)$$

$$\begin{aligned} f_1(x) &= f'(x) \text{ formal derivative of } f(x) \qquad \text{(B.1)} \\ &\ldots \\ f_{i+1} \text{ is such that } f_{i-1}(x) &= q_i(x)\, f_i(x) - f_{i+1}(x)\,, deg f_{i+1} < deg f_i \\ &\ldots \\ f_{s-1}(x) &= q_s(x)\, f_s(x)\,, \text{ that is } f_{s+1}(x) = 0. \end{aligned}$$

Assume that $[a, b]$ *is an interval such that* $f(a) \neq 0$, $f(b) \neq 0$. *Then the number of distinct (real) roots of* $f(x)$ *in* (a, b) *is* $K_a - K_b$ *where* K_c *denotes the number of variations in sign of* $\{f_0(c), f_1(c), \ldots, f_s(c)\}$. □

Tarski discovered in 1930 that in the first-order theory of real closed fields any quantifier elimination method also provides a decision method, which helps us decide whether any sentence of the theory is true or false. The result is given below.

Theorem B.11 *(Tarski's Theorem [87]) Let* φ *be a finite set of polynomial equations, inequations and inequalities of the form*

$$\begin{aligned} F(t_1, \ldots, t_r, x_1, \ldots, x_n) &= 0 \\ G(t_1, \ldots, t_r, x_1, \ldots, x_n) &\neq 0 \\ H(t_1, \ldots, t_r, x_1, \ldots, x_n) &> 0 \end{aligned}$$

where $F, G, H \in \mathbb{Q}[t_1, \ldots, t_r, x_1, \ldots, x_n]$. *Then we can determine in a finite number of steps a finite collection of finite sets* ψ_j *of polynomial equations, inequations and inequalities of the same type in the parameters* t_i *alone such that, if* R *is any real closed field, then the set* φ *has a solution for the* x*'s in* R *for* $t_i = C_i$, $1 \leq i \leq r$, *if and only if the* C_i *satisfy all the conditions of one of the sets* ψ_j. □

Tarski also provided an algorithm for quantifier elimination. Although of utmost importance, the Tarski's method is highly impractical for non trivial problems even with today's powerful computers. Actually, it can be shown that the computation time of Tarski's method can not be estimated by any tower of exponentials [73]. A number of other quantifier elimination methods were provided in literature, such as Seidenberg's [157], Cohen's [31], Collins' method [33] and more recently Grigor'ev's [73]. An interested reader should refer to [4] for a very good bibliography on the quantifier elimination problem.

Note the generality of Tarski's result: quantifier elimination is possible for *any sentence* in the first-order theory of real closed fields. Moreover, there exists an algorithm such that any problem of the above mentioned type, no matter how complex, can be solved in finite time. The generality of the formulation of the problem implies that the computational complexity would reduce the practicality of any such algorithm. Indeed, even today there are no algorithms which are computationally efficient and which can tackle "large scale" or very complex problems.

From a practical point of view, the Collins' method is probably the most important quantifier elimination method of today. CAD and its use for QE was discovered in 1973 by G. E. Collins [33, 35, 34]. The Collins' method is constructive and an algorithm follows from his method. The original algorithm is divided into CAD algorithm and QE algorithm. In order to carry out a quantifier elimination CAD should be computed first and then the quantifier elimination is done by using the QE algorithm. These two algorithms are implemented in a symbolic computation package called QEPCAD. It is based on the SACLIB package which was developed by prof. Collins and a number of other researchers. We describe below this method in some detail.

B.3.1 Cylindrical Algebraic Decomposition (CAD) and Quantifier Elimination (QE)

The following terminology is used in real computational algebra and in particular in the CAD algorithm.

Definition B.11

1. A region R is a connected subset of $\mathbb{R}^n$.

2. The set $Z(R) = R \times \mathbb{R} = \{(a,x) \colon a \in R,\ x \in \mathbb{R}\}$ is called a cylinder over R.

3. Let f, f_1, f_2 be continuous, real-valued functions on R. An f-section of $Z(R)$ is the set $\{(a, f(a)) \colon a \in R\}$ and a (f_1, f_2)-sector of $Z(R)$ is the set $\{(a,b) \colon a \in R, f_1(a) < b < f_2(a)\}$.

4. Let $X \subset \mathbb{R}^n$. A decomposition of X is a finite collection of disjoint regions (or components) whose union is X, that is $X = \cup_i X_i$, $X_i \cap X_j = \emptyset$, $i \neq j$.

5. A stack over R is a decomposition which consists of f_i-sections and (f_i, f_{i+1})-sectors where $f_0(x) < \ldots < f_{k+1}(x)$ for all $x \in R$ and $f_0 = -\infty$, $f_{k+1} = +\infty$. □

Definition B.12 A decomposition $\mathcal{D}$ of $\mathbb{R}^n$ is cylindrical if:

n=1 $\mathcal{D}$ is a partition of $\mathbb{R}^1$ into a finite set of numbers, and the finite and infinite open intervals bounded by these numbers.

$n > 1$ $\mathcal{D}'=F_1 \cup \ldots \cup F_m$ is a cylindrical decomposition of $\mathbb{R}^{n-1}$ and over each F_i there is a stack which is a subset of $\mathcal{D}$. □

Definition B.13 Let $X \in \mathbb{R}^n$ and $f \in \mathbb{R}[x_1, \ldots, x_n]$. Then f is invariant on X if one of $f(x) < 0$, $f(x) = 0$, $f(x) > 0$ holds for all $x \in X$. The set $\mathcal{F}=\{f_1, \ldots, f_r\} \in \mathbb{R}[x_1, \ldots, x_n]$ of polynomials is invariant on X if each f_i is invariant on X. X is $\mathcal{F}$-invariant if $\mathcal{F}$ is invariant on X. □

Definition B.14 A decomposition is algebraic if each of its components is a semi-algebraic set. □

Finally, we can state the definition of cylindrical algebraic decomposition (CAD) which plays a major role throughout the thesis.

Definition B.15 A Cylindrical Algebraic Decomposition (CAD) of $\mathbb{R}^n$ is a decomposition which is both cylindrical and algebraic. The components of CAD are called cells. □

We note that in the thesis we often refer to the algorithm which is used to compute CAD also as CAD. No confusion should arise from this. The input to the CAD algorithm is a set $\mathcal{F}$ of n-variate polynomial with rational coefficients and the output is a representation of a CAD of $\mathbb{R}^n$. All cells of the CAD are $\mathcal{F}$-invariant. The output also includes a sample point for each cell that can be used to determine the signs of the polynomials over that cell.

The CAD algorithm consists of three phases:

- **Projection:** The projection phase consists of a number of steps. At each step a new set of polynomials is constructed. The zero sets of the constructed polynomials represent the projection of "significant" points of the zero set of the preceding polynomials, such as isolated points, vertical tangent points, cusps, etc. At each step the number of variables is decreased by one and hence the projection phase consists of $n - 1$ steps.

- **Base:** The base phase consists of the isolation of the real roots $a_i \in \mathbb{R}$ of the univariate polynomials which are the output from the projection phase. Each root and one point in the each interval between two roots are chosen as sample points of a decomposition of $\mathbb{R}^1$.

- **Extension:** In the extension phase sample points of all cells of the CAD of $\mathbb{R}^n$ are constructed. The extension phase consists of $n - 1$ steps. In the first step a sample point $(a_i, b_j) \in \mathbb{R}^2$ of each cell of the stack over the cells of the base phase is constructed. The same procedure is repeated until we obtain sample points of all cells of the CAD of $\mathbb{R}^n$.

Once the CAD has been computed, the truth value of a formula can easily be decided [92]. The quantifier free Boolean expression in the original formula $\mathcal{F}(f_1(x), \ldots, f_2(x))$ is evaluated at each sample point and depending how a variable is quantified $\mathcal{F}(f_1(x), \ldots, f_2(x))$ has to be true for all (in the case of $\forall$ quantifier) or for some (in the case of $\exists$ quantifier) of the sample points.

If we want to do quantifier elimination, we evaluate $\mathcal{F}(f_1(x), \ldots, f_2(x))$ over the sample points. The cells corresponding to the sample points for which this formula is true can be characterised by the sign of the polynomials from the projection phase of the CAD algorithm. The solution formula can then be constructed by combining such partial formulas. We emphasize that once the CAD has been computed for a given set of polynomials, it is possible to solve **any real polynomial system defined by these polynomials**. For example, to determine whether a real polynomial system has a real solution it is sufficient to determine the signs of $\mathcal{F}$ at all sample points (of each cell) since $\mathcal{F}$ is invariant in each cell by construction.

B.3.2 Computational Complexity of the QEPCAD Algorithm

We use the following notation for a set of polynomials F (with integer coefficients) [33, 112]:

r the number of indeterminates $x_1, \ldots, x_r$ in F

m number of polynomials in F

d the maximum degree of any polynomial in F in any of the variables x_i

l the maximum norm length (the norm of an r-variate polynomial is the sun of absolute values of its integer coefficients).

An atomic formula is an expression of the form $f(x_1, \ldots, x_r)\,\delta g(x_1, \ldots, x_r)$, where $f, g \in \mathbb{Q}[x_1, \ldots, x_r]$ and δ is one of the relations $=, \geq, \leq, \neq$ [173]. The number of atomic formulas in the quantified expression (input formula) is denoted as a.

The maximum computation time of the CAD algorithm is dominated by [33]:

$$P_r(d, m, l) = (2d)^{2^{2r+8}} m^{2^{r+6}} l^3$$

Notice that for a fixed r the expression $P_r(d, m, l)$ is a polynomial in d, m and l. However, it

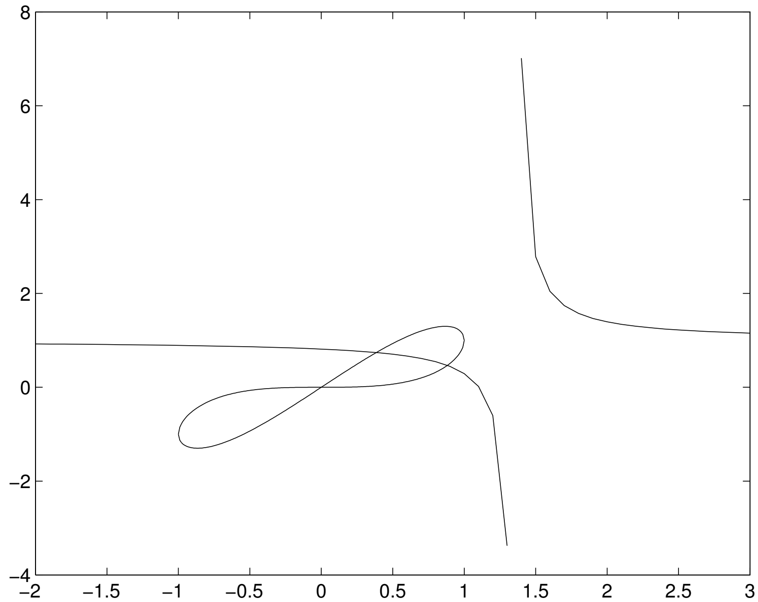

Figure B.1: The zero solution set for polynomials f_1=0 and f_2=0.

depends doubly exponentially on the number of variables r. Hence, the larger the number of variables in F, the larger the computation time of the CAD algorithm. In practice, this is reflected in such a way that tackling more than four variables in F is almost impossible apart from some special cases.

The QE algorithm has the computation time dominated by [33]:

$$P_r(d, m, l, a) = (2d)^{2^{2r+8}} m^{2^{r+6}} l^3 a$$

and it is also doubly exponential in the number of variables

B.3.3 An Illustrative Example

Given the following polynomials [92]

$$\begin{aligned} f_1 &= x_2^2 - 2x_1x_2 + x_1^4 \\ f_2 &= (2431x_1 - 3301)\,x_2 - 2431x_1 + 2685 \end{aligned}$$

find a CAD of $\mathbb{R}^2$. The zero set of the given polynomials is given in Figure B.1.

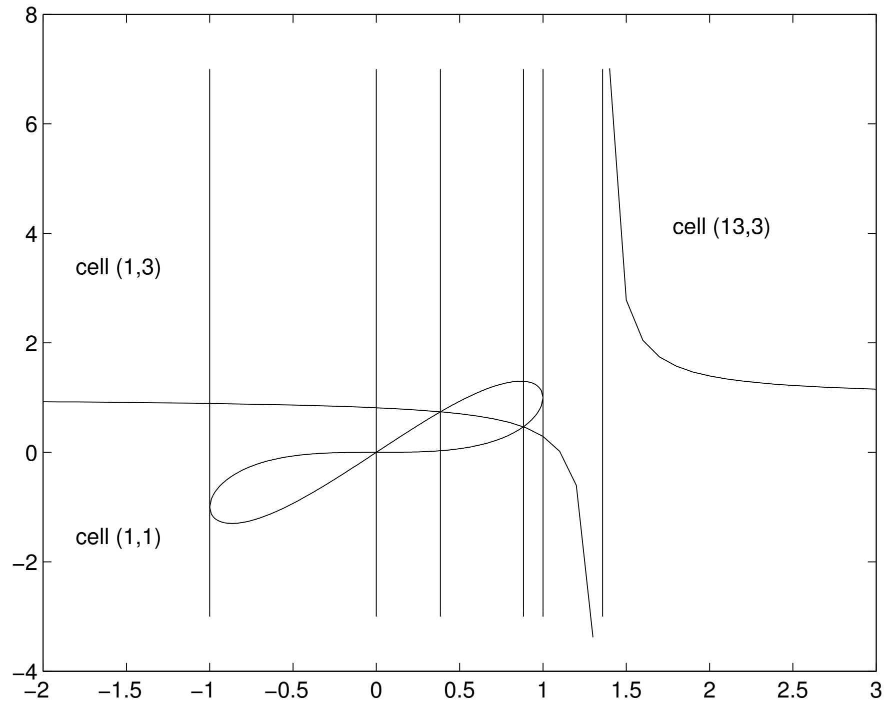

Figure B.2: CAD of $\mathbb{R}^2$

Projection: In the projection phase the following univariate polynomials are obtained:

$$\begin{aligned}
\mathrm{proj}_1(f_1) &= \{-2x_1, 1, x_1^4\} \\
\mathrm{proj}_1(f_2) &= \{-2431x_1 + 2685, 2431x_1 - 3301\} \\
\mathrm{proj}_2(f_1) &= \{4x_1^2(x_1 - 1)(x_1 + 1)\} \\
\mathrm{proj}_2(f_2) &= \{\} \\
\mathrm{proj}_3(\{f_1, f_2\}) &= \{-x_1(-4862x_1 + 5370 + 2431x_1^4 \\
&\quad -3301x_1^3), (17x_1 - 15)(13x_1 - 5)(26741x_1^4 \\
&\quad -38742x_1^3 - 8854x_1^2 - 51552x_1 + 96123)\}
\end{aligned}$$

Base: The real roots of the above given polynomials are

$$-1,\ 0,\ \frac{5}{13},\ \frac{15}{17},\ a \approx 0.93208,\ 1,\ \frac{2685}{2431},\ \frac{3301}{2431},\ b \approx 1.59982$$

where a and b are the real zeroes of $2431x_1^4 - 3301x_1^3 - 4862x_1 + 5370$. We only need the following five roots to determine a CAD of $\mathbb{R}^2$: $-1,\ 0,\ \frac{5}{13},\ \frac{15}{17},\ 1,\ \frac{3301}{2431}$. We also need sample points from each interval between the above roots, for example

$$-2,\ -\frac{1}{2},\ \frac{1}{4},\ \frac{1}{2},\ -\frac{9}{10},\ -\frac{5}{4},\ 2.$$

The base phase produces 13 sample points. The base phase therefore produces the following decomposition of $\mathbb{R}^1$:

$$]-\infty,-1[,\ -1,\ ]-1,0[,\ 0,\ ]0,\frac{5}{13}[,\ \frac{5}{13},\ ]\frac{5}{13},\frac{15}{17}[,$$

$$\frac{15}{17},\ ]\frac{15}{17},1[,\ 1,\ ]1,\frac{3301}{2431}[,\ \frac{3301}{2431},\ ]\frac{3301}{2431},+\infty[.$$

This decomposition consists of 13 regions. It is possible to construct a stack over each of these regions and this leads to a complete CAD of $\mathbb{R}^2$. We note here that each cell of CAD of $\mathbb{R}^2$ is enumerated and the signs of polynomial f_1 and f_2 are obtained for each of these cells. The enumeration is from left to right and from bottom to top in the usual sense. Examples of notation for several cells are presented in Figure B.2. The stack constructed over the region 1 ($]-\infty,-1[$) consists of three cells $(1,1)$, $(1,2)$, $(1,3)$. The cells $(1,1)$ and $(1,3)$ are the "white patches" denoted in Figure B.2 and the cell $(1,2)$ is the curve between them.

Extension: Only the sample points of cells $(7,j)$, $j{=}1,\ldots,7$ ($x_1{=}\frac{1}{2}$) are computed to illustrate the procedure. We have

$$f_1(\frac{1}{2},x_2) = x_2^2 - x_2 + \frac{1}{16}$$

$$f_2(\frac{1}{2},x_2) = -\frac{4171}{2}x_2 + \frac{2939}{2}$$

with real roots $\frac{1}{2} \pm \frac{1}{4}\sqrt{3}$ and $\frac{2939}{4171}$ respectively. Together with the four sample points in the intermediate intervals we get seven sample points, see Table B.1. The whole CAD of $\mathbb{R}^2$ consists of 63 cells, see Figure B.2. Given the signs of f_1 and f_2 over all cells in the CAD we can solve *any* real polynomial system defined by f_1 and f_2.

In order to illustrate how quantifier elimination can be carried out with the use of the computed CAD, let us consider the quantifier elimination problem:

$$(\exists x_2)\,[(f_1 \geq 0) \wedge (f_2{=}0)]$$

Cell number	Sample point	sign(f_1)	sign(f_2)
(7,1)	$]\frac{1}{2}, 0[$	+	+
(7,2)	$]\frac{1}{2}, \frac{1}{2} - \frac{1}{4}\sqrt{3}[$	0	+
(7,3)	$]\frac{1}{2}, \frac{1}{2}[$	-	+
(7,4)	$]\frac{1}{2}, \frac{2939}{4171}[$	-	0
(7,5)	$]\frac{1}{2}, \frac{3}{4}[$	-	-
(7,6)	$]\frac{1}{2}, \frac{1}{2} + \frac{1}{4}\sqrt{3}[$	0	-
(7,7)	$]\frac{1}{2}, 2[$	+	-

Table B.1: Sample points and signs of f_1 and f_2 for the cells $(7, j)$, j=$1, \ldots, 7$.

We consider again only the 7th stack whose sample points are given in Table B.1. We see that the polynomial f_2 is equal to zero only for the cell $(7, 4)$ but for the polynomial f_1 is negative on this cell. Hence, we can write that $(\forall x_1 \in]\frac{5}{13}, \frac{15}{17}[)\ (\nexists x_2)\ [f_1 \geq 0 \wedge f_2{=}0]$. In other words the semi-algebraic set defined by $\frac{5}{13} < x_1 < \frac{15}{17}$ is not a part of the semialgebraic set defined by the solution to the given QE problem. Notice, on the other hand, that the decision problem $(\exists x_1)\ (\exists x_2)\ [f_1 > 0 \wedge f_2 > 0]$ is TRUE since both polynomials are positive on the cell $(7, 1)$. We emphasize that with the obtained CAD we can solve *any QE/decision problem* formulated by means of the polynomials f_1, f_2.

References

[1] D. Aeyels and J. L. Willems, "Pole assignment for linear time-invariant second-order systems by periodic static output feedback," *IMA J. Math. Contr.*, vol. 8, pp. 267-274, 1991.

[2] D. Aeyels and J. L. Willems, "Pole assignment for linear time-invariant systems by periodic memoryless output feedback," *Automatica*, vol. 28, pp. 1159-1168, 1992.

[3] B. D. O. Anderson, N. K. Bose and E. I. Jury, "Output feedback stabilisation and related problems - solution via decision methods," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 53-66, 1975.

[4] D. S. Arnon, "A bibliography of quantifier elimination for real closed fields," *J. Symbolic Comput.*, vol. 5, pp. 267-274, 1988.

[5] D. S. Arnon, G. E. Collins, and S. McCallum, "Cylindrical algebraic decomposition: the basic algorithm," *SIAM J. Comput.*, vol. 13, pp. 865-877, 1984.

[6] K. J. Åström and B. Wittenmark, *Computer controlled systems*. Prentice-Hall: New Jersey, 1990.

[7] F. Albertini and E. D. Sontag, "Discrete-time transitivity and accessibility: analytic systems," *SIAM J. Contr. Optimiz.*, vol. 31, pp. 1599-1622, 1993.

[8] F. Albertini and E. D. Sontag, "Further results on controllability properties of discrete-time nonlinear systems," *Report SYCON-92-06*, 1992.

[9] J. Baillieul, " Controllability and observability of polynomial dynamical systems," *Nonlin. Analysis, Meth. Appl.*, vol. 5, pp. 543-552, 1981.

[10] J. Baillieul, " The geometry of homogeneous polynomial dynamical systems," *Nonlin. Analysis, Meth. Appl.*, vol. 5, pp. 879-900, 1981.

[11] W. Bamberger and R. Isermann, "Adaptive on-line steady-state optimization of slow dynamic processes," *Automatica*, vol. 14, pp. 223-230, 1978.

[12] R. Bars and R. Haber, "Long-range predictive control of nonlinear systems given by Volterra series," in *proc. 8th IFAC Symp. on Identification and System Parameter Estimation*, Beijing, China, pp. 939-946, 1988.

[13] A. Basseti, "Bifurcation and chaos in investments firm policy," in *proc. IFAC Dynamic Modelling and Control of National Economies*, Edinburgh, UK, pp. 441-444, 1989.

[14] G. Bastin, F. Jarachi and I. M. Y. Mareels, "Dead-beat control of recursive nonlinear systems," in *proc. 32nd IEEE Conf. Decision Contr.*, San Antonio, Texas, pp. 2965-2971, 1993.

[15] G. Bastin, F. Jarachi and I. M. Y. Mareels, "Output dead-beat control of nonlinear systems in one step: feasibility and stability conditions," preprint, 1994.

[16] G. Bastin, F. Jarachi and I. M. Y. Mareels, "Dead-beat control of recursive nonlinear systems," *Technical report 93.04*, Catholic University, Louvain-la-Neuve, Belgium, 1993.

[17] T. Becker and V. Weispfenning, *Gröbner bases: a computational approach to commutative algebra*. Springer-Verlag: New York, 1993.

[18] S. Beghelli and R. Guidorzi, "Bilinear systems identification from input-output sequences," in *proc. 4th IFAC Congress*, Tbilisi, USSR, pp.1759-1769, 1978.

[19] A. Benalou, D. A. Mellichamp and D. E. Seborg, " On the number of solutions of multivariable polynomial systems," *IEEE Trans. Automat. Contr.*, vol. AC-28, pp. 224-227, 1983.

[20] R. Benedetti and J. J. Risler, *Real algebraic and semi-algebraic sets*. Hermann: Paris, 1990.

[21] S. A. Billings and M. B. Fadzil, "The practical identification of systems with nonlinearities," in *proc. 7th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, York, UK, pp. 155-160, 1985.

[22] S. A. Billings, " An overview of nonlinear systems identification," in *proc. 7th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, York, UK, pp. 725-729, 1985.

[23] S. A. Billings, M. B. Fadzil, J. L. Sulley and P. M. Johnson, "Identification of a non-linear difference model of an industrial diesel generator", *Mech. Syst. Sign. Processing*, vol. 2, pp. 59-76, 1988.

[24] S. A. Billings and W. S. F. Voon, "A prediction-error and stepwise-regression estimation algorithm for non-linear systems," *Int. J. Control*, vol. 44, pp.803-822, 1986.

[25] C. C. Bissell, "Revisiting the classics: early control texts in English, German and Russian," in *proc. IFAC 13th Triennial World Congress*, San Francisco, vol. G, pp. 123-128, 1996.

[26] Y. Biran and B. McInnis, "Optimal control of bilinear systems to identify time varying effects of antitumor drugs," in *proc. IFAC 7th Triennial World Congress*, Helsinki, Finland, pp. 549-555, 1978.

[27] V. Blondel, M. Gevers and A. Lindquist, "Survey on the state of systems and control," *Eropean J. Contr.*, vol. 1, pp. 5-23, 1995.

[28] N. K. Bose and A. R. Modarressi, " General procedure for multivariable polynomial positivity test with control applications," *IEEE Trans. Automat. Contr.*, vol. AC-21, pp. 696-701, 1976.

[29] B. Buchberger, "Gröbner bases: an algorithmic method in polynomial ideal theory," in *Multidimensional systems theory*, N. K. Bose ed., D. Reidel Publishing Co.: Dordrecht, pp. 184-232, 1985.

[30] G-S. J. Cheng, T. J. Tarn and D. L. Elliott, "Controllability of bilinear systems," in *Variable structure systems with applications to economics and biology*, A. Ruberti and R. R. Mohler, eds., Springer-Verlag: Berlin, 1975.

[31] P. J. Cohen, "Decision procedures for real and p-adic fields," *Comm. Pure and Applied Math.*, vol. 22, pp. 131-151, 1968.

[32] G. E. Collins and R. Loos, " Real zeroes of polynomials," *Computing, Suppl.*, vol. 4, pp. 83-94, 1982.

[33] G. E. Collins, "Quantifier elimination for real closed fields by cylindrical algebraic decomposition," *Lect. Notes Comp. Sci.*, vol. 33, pp. 134-183, 1975.

[34] G. E. Collins, "Quantifier elimination by cylindrical algebraic decomposition - twenty years of progress," to appear in *Quantifier elimination and cylindrical algebraic decomposition*, Springer-Verlag: Berlin, 1996.

[35] G. E. Collins and Hoon Hong, "Partial cylindrical algebraic decomposition for quantifier elimination," *J. Symbolic Comp.*, vol. 12, pp. 299-138, 1991.

[36] G. E. Collins, Personal correspondence, 1995-96.

[37] D. Cox, J. Little and D. O'Shea, *Ideals, varieties and Algorithms*. Springer-Verlag: New York, 1992.

[38] W. de Melo and S.van Strien, *One-dimensional dynamics*. Springer-Verlag: Berlin, 1991.

[39] C. A. Desoer and J. Wing, "An optimal strategy for a saturating sampled-data system," *IRE Trans. Automat. Contr.*, vol. AC-6, pp. 5-15, 1961.

[40] C. A. Desoer and J. Wing, "A minimal time discrete system," *IRE Trans. Autom. Contr.*, vol. AC-6, pp. 111-125, 1961.

[41] R. L. Devaney, *An introduction to chaotic dynamical systems*. Addison-Wesley Publishing Company: Redwood City, 1989.

[42] H. Diaz and A. A. Desrochers, "Modelling of nonlinear discrete-time systems from input-output data," *Automatica*, vol. 24, pp. 629-641, 1988.

[43] J. A. Dieudonne, *Foundations of modern analysis*. Academic Press: New York, 1960.

[44] D. Dochain and G. Bastin, "Adaptive Identification and control algorithms for nonlinear bacterial growth systems," *Automatica*, vol. 20, pp. 621-634, 1984.

[45] P. Dorato, W. Yang and C. Abdallah, "Robust multi-objective feedback design by quantifier elimination," *J. Symb. Comp.*, vol. 11, pp. 1-7, 1996.

[46] J. J. Duffy and M. A. Franklin, "A learning identification algorithm and its application to an environmental system," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-5, pp. 226-240, 1975.

[47] J. C. Engwerda and P. W. Otter, "On the choice of weighting matrices in the minimum variance controller," *Automatica*, vol. 25, pp. 279-285, 1989.

[48] M. E. Evans and D. N. P. Murthy, "Controllability of a class of discrete time bilinear systems," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 78-83, 1977.

[49] M. E. Evans and D. N. P. Murthy, "Controllability of discrete time inhomogeneous bilinear systems," *Automatica*, vol. 14, pp. 147-151, 1978.

[50] M. E. Evans and D. N. P. Murthy, "Controllability of discrete time systems with positive controls," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 942-945, 1977.

[51] M. E. Evans and D. N. P. Murthy, "Comments on "The controllability of discrete linear systems with output feedback"," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 672-673, 1977.

[52] M. E. Evans, "The convex controller: controllability in finite time," *Int. J. Sys. Science*, vol. 16, pp. 31-47, 1985.

[53] M. E. Evans, "Bounded control and discrete time controllability," *Int. J. Systems Sci.*, vol. 17, pp. 943-951, 1986.

[54] M. E. Evans, "Pole-assignment with periodic output feedback: computer assisted study on a 2-dimensional linear system," *Int. J. Systems Sci.*, vol. 19, pp. 2661-2672, 1988.

[55] M. E. Evans M. E., "Convex input sets and controllability of discrete time linear systems," in *proc. 12th IFAC Triennial World Congress*, vol. 1, pp.187-193, 1993.

[56] M. E. Evans, "Controllability with convex input sets, a complete characterisation," preprint, 1995.

[57] T. Fliegner, "Contributions to the control of nonlinear discrete-time systems", Ph. D. dissertation, Univ. Twente, Enschede, The Netherlands, 1995.

[58] K. H. Fasol and G. M. Pohl, "Simulation, controller design and field tests for a hydro-power plant - a case study," *Automatica*, vol. 26, pp. 475-485, 1990.

[59] M. Fliess and D. Normand-Cyrot, "A group-theoretic approach to discrete-time nonlinear controllability," in *proc. IEEE Conf. Decision Contr.*, Albuquerque, NM, pp. 551-557, 1981.

[60] M. Fliess and D. Normand-Cyrot, "On the approximation of nonlinear systems by some simple state-space models," in *proc. IFAC Symp. Identification and Systems Parameter Estimation*, Washington D. C., pp. 511-514, 1982.

[61] W. Fulton, *Algebraic curves*. W. A. Benjamin, Inc.: New York, 1969.

[62] K. Forsman and S. T. Glad, "Gröbner bases - a new tool in nonlinear control theory," *Report LiTH-ISY-I-1068*, Linköping University, Sweden, 1990.

[63] A. T. Fuller, " Root location criteria for quartic equations," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 777-782, 1981.

[64] Y. Funahashi and H. Katoh, "Robust-tracking dead-beat control," *Int. J. Control*, vol. 56, pp. 213-225, 1992.

[65] Y. Funahashi, "Comments on "Controllability of a class of discrete time bilinear systems," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 667-668, 1979.

[66] S. T. Glad, "Output dead-beat control for nonlinear systems with one zero at infinity," *Syst. Contr. Lett.*, vol. 9, pp.249-255, 1987.

[67] S. T. Glad, "Dead-beat control for nonlinear systems," in *Analysis and Control of Nonlinear Systems*, C. I. Byrnes, C. F. Martin and R. E. Saeks, eds., North-Holland: Amsterdam, 1988.

[68] S. T. Glad, "An algebraic approach to bang-bang control," in *proc. 3rd European Contr. Conf.*, Rome, Italy, pp. 2892-2895, 1995.

[69] G. C. Goodwin, B. McInnis and R. S. Long, "Adaptive control algorithms for waste water treatment and pH neutralisation," *Opt. Contr. Appl. Methods*, vol. 3, pp. 443-459, 1982.

[70] T. Goka, T. J. Tarn and J. Zaborszky, "On the controllability of a class of discrete bilinear systems," *Automatica*, vol. 9, pp. 615-622, 1973.

[71] O. M. Grasselli, A. Isidori and F. Nicolo, "Dead-beat control of discrete-time bilinear systems," *Int. J. Control*, vol. 32, pp. 31-39, 1980.

[72] J. P. Greschak and G. C. Verghese, "Periodically varying compensation of time-invariant systems," *Sys. Contr. Lett.*, vol. 2, pp. 88-93, 1982.

[73] D. Y. Grigor'ev, "Complexity of deciding Tarski algebra," *J. Symbolic Comp.*, vol. 5, pp. 65-108, 1988.

[74] A.-M. Guillaume, G. Bastin and G. Campion, "Sampled-data adaptive control of a class of continuous nonlinear systems," *Int. J. Control*, vol. 60, pp. 569-594, 1994.

[75] R. Haber and L. Keviczky, "Identification of nonlinear dynamic systems," in *proc. 4th IFAC Symp. Identification and System Parameter Estimation*, Tbilisi, U.S.S.R., pp. 79-126, 1978.

[76] R. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems - a survey of input/output approaches," *Automatica*, vol. 26, pp. 651-677, 1990.

[77] R. Haber and R. Ziefuß, "Identification of nonlinear models between the reflux flow and the top temperature of a distillation column," in *proc. 9th IFAC/IFORS Symp. on Identification and and System Parameter Estimation*, Budapest, Hungary, pp. 486-491, 1991.

[78] R. Haber, L. Keviczky and H. Unbehauen, "Application of adaptive control on nonlinear dynamic processes - a survey on input-output approaches," in *proc. 10th IFAC Triennial World Congress*, Munich, Germany, pp. 1-14, 1987.

[79] W. Hahn, *Stability of motion*. Springer-Verlag: Berlin, 1967.

[80] J. Heintz, M.-F. Roy and P. Solernó, "On the theoretical and practical complexity of the existential theory of reals," *The Comp. Journal*, vol. 36, pp. 427-431

[81] R. Herman and A. J. Krener, "Nonlinear controllability and observability," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 728-740, 1977.

[82] P. Hollis and D. N. P. Murthy, "Study of uncontrollable discrete bilinear systems," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 184-186, 1982.

[83] H. Hong, "Quantifier elimination for formulas constrained ny quadratic equations via slope resultants," *The Comp. Journal*, vol. 36, pp. 439-449.

[84] A. Ichikawa and H. Katayama, "The design of dead-beat controllers by state transition graph," *IEEE Trans. Automat. Contr.*, vol. AC-36, pp.752-756, 1991.

[85] S. Ikeda, M. Ochiai and Y. Sawaragi, "Sequential GMDH algorithm and its application to river flow prediction," *IEEE Trans. Syst. Man Cybern.*, SMC-6, pp. 473-479, 1976.

[86] A. Isidori, *Nonlinear control systems - third edition*. Springer-Verlag: Berlin, 1995.

[87] N. Jacobson, *Basic Algebra I*. W. H. Freeman and Co.: San Francisco, 1974.

[88] B. de Jager, "The use of symbolic computation in nonlinear control: is it viable?," *IEEE Trans. Automat. Contr.*, vol. AC-40, pp. 84-89, 1995.

[89] U. J. Jaaksoo, E. M. Talvis and J. M. Ummer, "Microprocessor-based grain dryer control," in *proc. 3rd IFAC/IFIP Symp. on Software for Computer Control*, Madrid, Spain, pp. 469-473, 1982.

[90] B. Jakubczyk and E. D. Sontag, “Controllability of nonlinear discrete-time systems: a Lie algebraic approach,” *SIAM J. Contr. Optimiz.*, vol. 28, pp. 1-33, 1990.

[91] B. Jakubczyk, “ Feedback linearisation of discrete-time systems,” *Syst. Contr. Lett.*, vol. 9, pp. 411-416, 1987.

[92] M. Jirstrand, *Algebraic methods for modelling and design in control (Thesis no. 540)*. Linköping Studies in Science and Technology: Linköping, 1996.

[93] V. Jurdjevic, “ Polynomial control systems,” *Math. Ann.*, vol. 272, pp. 361-368, 1985.

[94] B. Leden, “Dead-beat control and the Riccati equation,” *IEEE Trans. Autom. Contr.*, vol. AC-21, pp. 791-792, 1976.

[95] B. Leden, “Multivariable dead-beat control,” *Automatica*, vol.13, pp. 185-188, 1977.

[96] H-G. Lee and S. I. Marcus, “ On input-output linearization of discrete-time nonlinear systems,” *Syst. Contr. Lett.*, vol. 8, pp. 249-259, 1987.

[97] I. J. Leonitaris and S. A. Billings, “Input-output parametric models for nonlinear systems. Part I: deterministic nonlinear systems,” *Int. J. Contr.*, vol. 41, pp. 303-328, 1985.

[98] I. J. Leonitaris and S. A. Billings, “Input-output parametric models for nonlinear systems. Part II: stochastic nonlinear systems,” *Int. J. Contr.*, vol. 41, pp. 329-344, 1985.

[99] F. Lewis, “Riccati equation solution to the minimum time output control problem,” *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 763-766, 1981.

[100] F. Lewis, “A general Riccati equation solution to the dead-beat control problem,” *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 186-188, 1982.

[101] R. Loos and V. Weispfenning, “Applying linear quantifier elimination,” *The Comp. Journal*, vol. 36, pp. 450-462, 1993.

[102] T. Kailath, *Linear Systems*. Prentice Hall: New Jersey, 1980.

[103] S. S. Keerthi and E. G. Gilbert, “Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations,” *J. Optimisation Theory and Applications*, vol. 57, pp. 265-293, 1988.

[104] L. Keviczky, "Nonlinear dynamic identification of a cement mill to be optimised," in *Identification and system parameter estimation*, N. S. Rajban ed., North-Holland: Amsterdam, pp. 1185-1195, 1978.

[105] V. L. Kocic and G. Ladas, *Global behaviour of nonlinear difference equations of higher order with applications*. Kluwer Academic Publishers: Dordrecht, 1993.

[106] M. Kortmann and H. Unbehauen, "Two algorithms for model structure determination of nonlinear dynamic systems with applications to industrial processes," in *proc. 8th IFAC Symp. on Identification and System Parameter Estimation*, Beijing, China, pp. 939-946, 1988.

[107] U. Kötta and U. Nurges, "Identification of input-output bilinear systems," in *proc. 9th IFAC Triennial World Congress*, Budapest, Hungary, pp. 723-727, 1984.

[108] U. Kötta, *Inversion method in the discrete time nonlinear control synthesis problems*. Springer-Verlag: Berlin, 1995.

[109] M. Krstić, I. Kanellakopoulos, P. Kokotović, *Nonlinear and adaptive control design*. John Wiley & Sons: New York, 1995.

[110] A. Krzyzak and J. Sasiadek, "Displacement identification of flexible manipulator arm using Hammerstein model," in *proc. ACC*, Pittsburgh, Pensilvania, pp.2360-2363, 1989.

[111] V. Kučera, "A dead-beat servo problem," *Int. J. Control*, vol. 32, pp. 107-113, 1980.

[112] S. McCallum, "Solving polynomial strict inequalities using cylindrical algebraic decomposition," *The Comp. Journal*, vol. 36, pp. 432-438, 1993.

[113] B. C. McInnis, Deng L-z and R. Vogt, "Adaptive pole assignment control of blood pressures using bilinear models," in *proc. 7th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, York, UK, pp. 1209-1211, 1985.

[114] B. R. Maner , F. J. Doyle, B. A. Ogunnaike and R. K. Pearson, "A nonlinear model predictive control scheme using second order Volterra models," in *proc. ACC*, Baltimore, Maryland, pp. 3253-3257, 1994.

[115] E. Masry and S. Cambains, "Signal identification after noisy, nonlinear transformations," *IEEE Trans. Info. Theory*, vol. IT-28, pp. 50-58, 1980.

[116] D. Q. Mayne and W. R. Schroeder, "Nonlinear control of linear systems with state and control constraints," preprint, 1995.

[117] D. V. H. Mien and D. Normand-Cyrot, "Nonlinear state affine identification methods: applications to electrical power plants," *Automatica*, vol. 20, pp. 175-188, 1984.

[118] A. Mital, "Prediction of human static and dynamic strengths by modified basic GMDH algorithm," *IEEE Trans. Syst. Man Cybern.*, SMC-14, pp. 773-776, 1984.

[119] R. R. Mohler, X. Yang and Z. Tang, "Discrete bilinear system identification and control," in *proc. ACC*, San Diego, California, pp. 1915-1916, 1990.

[120] R. R. Mohler, V. Rajkumar and R. R. Zakrewski, "Nonlinear time series and control applications," in *proc. IEEE Conf. Decision Contr.*, Brighton, England, pp. 2917-2919, 1991.

[121] S. Monaco and D. Normand-Cyrot, "On the immersion of a discrete-time polynomial analytic system into a polynomial affine one," *Syst. Contr. Lett.*, vol. 3, pp. 83-90, 1983.

[122] S. Monaco and D. Normand-Cyrot, "Invariant distributions for discrete-time nonlinear systems," *Syst. Contr. Lett.*, vol. 5, pp. 191-196, 1984.

[123] S. Monaco and D. Normand-Cyrot, "Minimum-phase nonlinear discrete-time systems," in *proc. 28th Conf. Decision Contr.*, Los Angeles, CA, pp. 979-986, 1987.

[124] S. Monaco and D. Normand-Cyrot, "Zero dynamics of sampled nonlinear systems," *Syst. Contr. Lett.*, vol. 11, pp. 229-234, 1988.

[125] R. E. Mortensen, *Random signals and systems*. Willey: New York, 1987.

[126] C. T. Mullis, "Time optimal discrete regulator gains," *IEEE Trans. Automat. Contr.*, vol. AC-17, pp. 265-266, 1972.

[127] C. T. Mullis, "On the controllability of discrete linear systems with output feedback," *IEEE Trans. Automat. Contr.*, vol. AC-18, pp. 608-614, 1973.

[128] D. N. P. Murthy, "Controllability of a discrete time bilinear system," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 974-975, 1979.

[129] D. N. P. Murthy, "Controllability of a linear positive dynamic system," *Int. J. Systems Sci.*, vol. 17, pp. 49-54, 1986.

[130] H. Nijmeijer and A. van der Shaft, *Nonlinear dynamical control systems*. Springer-Verlag: New York, 1990.

[131] D. Nešić, I. M. Y. Mareels, G. Bastin and R. Mahony, "ν-step controllability of scalar polynomial systems,"in *proc. 3rd European Contr. Conf.*, Rome, Italy, pp. 277-282, 1995.

[132] D. Nešić, I. M. Y. Mareels, G. Bastin and R. Mahony, "Necessary and sufficient conditions for output dead-beat controllability for a class of polynomial systems," in *proc. IEEE Conf. Decision Contr.*, New Orleans, pp. 7-13, 1995.

[133] D. Nešić and I. M. Y. Mareels, "Invariant sets and output dead-beat controllability for odd polynomial systems: the Gröbner basis method"' in *proc. 13th IFAC World Congress*, San Francisco, vol. E, pp. 221-226, 1996.

[134] D. Nešić, I. M. Y. Mareels, G. Bastin and R. Mahony, "Stability of implicitly defined polynomial dynamics: the scalar case," presented at *MTNS*, St. Louis, 1996.

[135] D. Nešić and I. M. Y. Mareels, "Deciding dead-beat controllability using QEPCAD," presented at *MTNS*, St. Louis, 1996.

[136] D. Nešić and I. M. Y. Mareels, "Minimum-time dead-beat control of simple Hammerstein systems," presented at *MTNS*, St. Louis, 1996.

[137] D. Nešić and I. M. Y. Mareels, "An output dead-beat controllability test for a class of odd polynomial systems," submitted, 1996.

[138] D. Nešić and I. M. Y. Mareels, "On some triangular structures and the state dead-beat problem for polynomial systems," submitted, 1996.

[139] D. Nešić and I. M. Y. Mareels, "Stability of high order implicit polynomial dynamics," submitted, 1996.

[140] D. Nešić and I. M. Y. Mareels, "The definition of minimum phase discrete-time nonlinear systems revisited," to appear in *proc. 4th Int. Conf. on Control, Automation, Robotics and Vision*, 1996.

[141] D. Nešić and I. M. Y. Mareels, "Scalar polynomial systems, triangular structures and dead-beat controllability," submitted, 1996.

[142] D. Nešić and I. M. Y. Mareels, "Dead-beat control of polynomial scalar systems," in revision, 1994.

[143] D. Nešić, I. M. Y. Mareels, G. Bastin and R. Mahony, "Output dead-beat control for a class of planar polynomial systems," submitted, 1995.

[144] D. Nešić and I. M. Y. Mareels, "Dead-beat controllability of polynomial systems: symbolic computation approaches," submitted, 1995.

[145] D. Nešić and I. M. Y. Mareels, "Dead-beat control of simple Hammerstein models," in revision, 1995.

[146] D. Nešić, "Controllability of generalised Hammerstein systems," submitted, 1995.

[147] D. Nešić and I. M. Y. Mareels, "State dead-beat controllability of structured polynomial systems," in preparation, 1996.

[148] D. Nešić and I. M. Y. Mareels, "Stability of implicit and explicit polynomial dynamics: symbolic computation approaches," in preparation, 1996.

[149] E. Nobuyama, "Parametrisation of all ripple-free dead-beat controllers," *Syst. Contr. Lett.*, vol. 21, pp. 217-224, 1993.

[150] T. Nomura and K. Furuta, "On the span reachability of polynomial state-affine systems," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 630-632, 1979.

[151] J. O'Reilly, "The linear time invariant time-optimal control problem - an overview," *Automatica*, vol. 17, pp. 363-370, 1981.

[152] M. J. Panik, *Fundamentals of convex analysis*. Kluwer Academic Publishers: Dordrecht, 1993.

[153] R. T. Rockafellar, *Convex analysis*. Princeton University press: Princeton, New Jersey, 1970.

[154] N. Rouche, P. Habets and M. Laloy, *Stability theory by Liapunov's direct method*. Springer-Verlag: Berlin, 1977.

[155] N. Samardzija, " Controllability, pole placement, and stabilisability for homogeneous polynomial systems," *IEEE Trans. Automat. Contr.*, vol. AC-29, pp. 1042-1045, 1984.

[156] S. Schaal and C. G. Atkeson, “Robot juggling: implementation of memory-based learning,” *IEEE Control Systems*, February, pp. 57-71, 1994.

[157] A. Seidenberg, “A new decision method for elementary algebra,” *Annals of Math.*, vol. 60, pp. 365-374, 1954.

[158] R. Sepulchre, M. Janković and P. Kokotović, “Integrator forwarding: a new recursive nonlinear robust design,” preprint, 1996.

[159] C. Simões, *On stabilisation of discrete time nonlinear systems*, PhD Thesis, University of Twente, The Netherlands, 1996.

[160] C. Simões, H. Nijmeijer and J, Tsinias, “Nonsmooth stabilisability and feedback linearisation of discrete time nonlinear systems,” *Int. J. Robust and Nonlinear Control*, to appear, 1995.

[161] C. Simões and H. Nijmeijer, “Stability and stabilizability of implicit discrete-time nonlinear systems,” *Memorandum no. 1222*, University of Twente, The Netherlands, 1994.

[162] C. Simões and H. Nijmeijer, “Stability, stabilisation and factorisation of implicit discrete-time nonlinear systems,” preprint, 1995.

[163] E. D. Sontag, *Polynomial response maps*. Springer-Verlag: New York, 1979.

[164] E. D. Sontag and Y. Rouchaleau, “On discrete-time polynomial systems,” *Nonlinear Analysis, Theory, Methods and Applications*, vol. 1, pp. 55-64, 1976.

[165] E. D. Sontag, “An algebraic approach to bounded controllability of linear systems,” *Int. J. Control*, vol. 39, pp. 181-188, 1984.

[166] E. D. Sontag, “Controllability is harder to decide than accessibility,” *SIAM J. Contr. Optimiz.*, vol. 26., pp. 1106-1118, 1988.

[167] E. D. Sontag, “A Lyapunov-like characterisation of asymptotic controllability,” *SIAM J. Contr. Optimiz.*, vol. 21, pp. 462-471, 1983.

[168] K. B. Strömborg, K. E. Häggblom, H. T. Toivonen and K. V. Waller, “Nonlinear and adaptive multivariable control of a distillation column,” in *proc. ACC*, Baltimore, Maryland, pp. 3516-3519, 1994.

[169] J. Uspensky, *Theory of equations*. McGraw-Hill: New York, 1948.

[170] T. J. Tarn, D. L. Elliott and T. Goka, "Controllability of discrete time bilinear systems with bounded control," *IEEE Trans. Automat. Contr.*, vol. AC-18, pp. 298-301, 1973.

[171] T. J. Tarn, "Singular control of bilinear discrete systems," *Information and Control*, vol. 21, pp. 211-234, 1972.

[172] S. Vajda, " Deterministic and algebraic invariants for polynomial systems," *IEEE Trans. Automat. Contr.*, vol. AC-32, pp. 182-184, 1987.

[173] V. Weispfenning, "Quantifier elimination for real algebra - the quadratic case and beyond," preprint, 1996.

[174] J. Wing and C. A. Desoer, "The multiple-input minimal time regulator problem (general theory)," *IEEE Trans. Automat. Contr.*, vol. AC-8, pp. 125-136, 1963.

[175] T. Yoshimura, D. Deepack and H. Takagi, "Track/vehicle systems identification by a revised group method of data handling (GMDH)," *Int. J. Syst. Sci.*, vol. 16, pp. 131-144, 1985.

[176] T. Yoshimura, R. Kiyozumi, K. Nishino and T. Soeda, "Prediction of air pollutant concentrations by revised GMDH algorithms in Tokushima prefecture," *IEEE Trans. Syst. Man Cybern.*, SMC-12, pp. 50-55, 1982.

[177] T. Yoshimura, U. S. Pandey, T. Takagi and T. Soeda, "Prediction of the peak flood using revised GMDH algorithm," *Int. J. Syst. Sci.*, vol. 13, pp. 547-557, 1982.

[178] Y. Zhao, "The proof of Goka's conjecture," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 972-974.

[179] Y. Zhao and H. Kimura, "Dead-beat control with robustness," *Int. J. Control*, vol. 43, pp. 1427-1440, 1986.

[180] Y. Zhao and H. Kimura, "Multivariable dead-beat control with robustness," *Int. J. Control*, vol. 47, pp. 229-255, 1988.

[181] Y. Zhao and H. Kimura, "Two-degrees-of-freedom dead-beat control system with robustness," *Int. J. Control*, vol. 48, pp. 303-315, 1988.

[182] Y. Zhao and H. Kimura, "Two-degrees-of-freedom dead-beat control with robustness: multivariable case," *Int. J. Control*, vol. 49, pp. 667-679, 1989.

[183] Y. Zhao and H. Kimura, "Optimization of control input for dead-beat control systems," *Int. J. Control*, vol. 59, pp. 1107-1117, 1994.

[184] Q. M. Zhu and S. A. Billings, "Identification of polynomial and rational NARMAX models," in *proc. 10th IFAC Symp. on Identification and System Parameter Estimation*, Copenhagen, pp. 295-300, 1994.

[185] Q.-M. Zhu and K. Warwick, "Dead-beat controller for a class of nonlinear systems," *Contr. Theory Adv. Technology*, vol. 7, pp. 101-116, 1991.

Index