Dead beat controllability of polynomial systems: symbolic computation approaches*

Dragan Nešić[†] and Iven M. Y. Mareels

Department of Systems Engineering and

Cooperative Research Centre for

Robust and Adaptive Systems

RSISE

ANU

Canberra, ACT 0200

AUSTRALIA

Keywords: polynomial systems, dead beat, controllability

*The authors wish to acknowledge the funding of the activities of the Cooperative Research Centre for Robust and Adaptive Systems by the Australian Government under the Cooperative Research Centres Program.

[†]The author to whom all correspondence should be addressed. e-mail address: dragan.nesic@anu.edu.au, tel. - int + 61 + 6 + 2492456, fax. - int + 61 + 6 + 2798088

Abstract

State and output dead beat controllability tests for a very large class of polynomial systems with rational coefficients may be based on the QEPCAD symbolic computation program. The method is unified for a very large class of systems and can handle one or two sided control constraints. Families of minimum time state/output dead beat controllers are obtained. The computational complexity of the test is prohibitive for general polynomial systems but by constraining the structure of the system, we may beat the curse of complexity. A computationally less expensive algebraic test for output dead beat controllability for a class of odd polynomial systems is presented. Necessary and sufficient conditions are given. They are still very difficult to check. Therefore, a number of easier-to-check sufficient conditions are also provided. The latter are based on the Gröbner basis method and QEPCAD. It is shown on a subclass of odd polynomial systems how it is possible to further reduce the computational complexity by exploiting the structure of the system.

1 Introduction

In the last 35 years, we have witnessed great advances in dead beat (DB) control theory for linear systems. Starting with Kalman's state space approach and his most elegant solution to the minimum time control of linear discrete time systems, the problem of DB control has attracted the attention of many researchers [23]. In the linear framework, almost all aspects of the problem have been addressed, most of them now being solved. Indeed, many different design approaches, robustness of DB controllers, trade off between the magnitude of control signals and the DB time, case studies, etc. have been reported in the literature.

On the other hand, nonlinear DB control has not received as much attention. The computational complexity caused by nonlinearities forces us to tackle special classes of non linear systems rather than develop a general approach to DB control. Most papers in the literature address only the existence problem [10, 11, 12, 17, 24, 25] whereas just a few designs have been reported (see e.g. [24, 25]). Obviously, all systems for which we know how to design a DB controller have a special structure which reduces the computational complexity, which is inherent in the general problem, in a substantial way.

In this paper, we restrict our considerations to classes of discrete time non linear systems which allow both a state space and output representation in a polynomial format. More precisely, we consider sub classes of the following class of polynomial systems:

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

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. Other assumptions on the structure of the system are of technical nature and are introduced later. The motivation for the analysis of polynomial systems comes, for example, from identification techniques of block oriented models [18] that yield input output identified models of the form:

$$y(k+1) = F(y(k), \dots, y(k-s), u(k), \dots, u(k-t))$$

where $y(k)$ and $u(k)$ are output and input of the system and F is a polynomial function in all arguments. A comprehensive study of realisation theory for this class of systems is given in [27].

Several techniques have been used to test certain controllability properties of non linear discrete time systems. They range from Lie algebraic techniques in [22] to more “linear-like” techniques in [10, 11, 12, 28]. Also, the controllability properties and classes of systems that have been investigated in the literature differ considerably and there does not seem to be a unifying approach which would be applicable to a large class of systems. Some results on output dead beat control of non linear systems can be found in [1]. However, the authors took the predictive control approach in [1] and the definition of output dead beat control is different from ours.

The results of this paper are in the spirit of [10, 11, 12, 28] but are very closely related to results in [24, 25, 26]. In [24] we considered the simplest class of polynomial systems: scalar polynomial systems. The results gave us a lot of insight into the properties of scalar systems but unfortunately could not be generalised to higher order systems. In [25] results on output dead beat controllability for a class of second order odd polynomial systems were presented. In the last part of this paper we generalise results in [25] to systems of arbitrary order. We illustrated how the Gröbner basis method can be used to find invariant sets of a variety in [26]. In this paper we build on the results of [26] to obtain a most comprehensive symbolic computation approach to state/output dead beat control for more general classes of polynomial systems than the ones considered in [24, 25, 26].

We present here several state and output DB controllability tests for a very large class of systems (1). The method naturally leads to a design method for families of state/output minimum time DB controllers. The tests exploit the symbolic computation package called QEPCAD and are applicable to MIMO polynomial systems, allowing also control constraints, such as bounded or positive controls. QEPCAD can be used to compute the sets of states for (1) that can be transferred to the origin in one, two, etc. time steps and this information is used to test dead beat controllability.

The applicability of the method is limited by the computational requirements. Although the computational complexity may be formidable in general, the complexity curse may not be an issue when considering systems of specific structure, such as bilinear or generalised Hammerstein, and solutions to non trivial examples are presented. Moreover, by exploiting the structure of the considered system we may be able to use other algorithms to further reduce the computational requirements.

In the second part of the paper we concentrate on the output DB controllability properties for the so called odd polynomial systems and illustrate how it is possible to use the Gröbner basis method to test it. The method that we use illustrates how it is possible to determine invariant sets, which are important to describe output DB controllability properties. The controllability tests that we propose still suffer from computational inefficiency although the test is less computationally demanding than the one based on QEPCAD. It is our opinion that the lack of efficiency of the tests come from the structure of the considered systems and it can be regarded as an intrinsic property of polynomial systems. A number of easier-to-check necessary conditions and sufficient conditions for output DB controllability are then presented.

Finally, a minimum time DB algorithm for a subclass of odd polynomial systems is presented. The output DB controllability test is more explicit and easier to check in this case. However, this sub class of odd systems does not appear to be very large. The paper illustrates the trade offs between the generality of the proposed tests and their feasibility.

We emphasize that the proposed algorithms may be very successfully modified by exploiting the structure of the considered class of polynomial systems (see Example 1). Moreover, it seems that the design of minimum time DB controllers for (1) would always require the use of QEPCAD or a similar algorithm. Consequently, any computational improvements in the DB controller design would hinge on the advances in this field of computer science/mathematics.

The paper is organised as follows. In Section 2 we list some results from algebraic geometry, and give some definitions. Section 3 is dedicated to state and output DB controllability tests, which are based on the QEPCAD algorithm. We demonstrate how a family of DB controllers can be designed using the method. Then, in Section 4 we consider a subclass of systems considered in Section 3, called odd systems, and develop another approach for output DB controllability. The controllability test is based on the Gröbner basis method and QEPCAD but it is in general less computationally demanding than the one based on QEPCAD in the previous Section. In Section 5 we further specialise the structure of odd polynomial systems and obtain stronger conclusions on output DB controllability which eventually lead to a much simpler design of a minimum time output DB controller. Our results are summarised in the last Section.

2 Preliminaries

We use [9] as a main reference for most of the results from algebraic geometry that are given below. We use standard definitions of rings and fields. The sets of integers, real and rational numbers are respectively denoted as Z, $\Re$ and Q. $\Re^n$ is a set of all n-tuples of elements of $\Re$, where n is a non negative integer. The ring of polynomials in n variables over the real field $\Re$ is denoted as $\Re[x_1, x_2, \ldots, x_n]$. Let $f_1, f_2, \ldots, f_s$ be polynomials in $\Re[x_1, x_2, \ldots, x_n]$. Then we define

$$V(f_1, f_2, \ldots, f_s) = \{(a_1, a_2, \ldots, a_n) \in \Re^n : f_i(a_1, a_2, \ldots, a_n) = 0 \text{ for all } 1 \leq \text{i} \leq \text{s}\}.$$

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. A subset $I \subset \Re[x_1, x_2, \ldots, x_n]$ is an ideal if it contains the zero polynomial, is closed under addition and multiplication by another polynomial in $\Re[x_1, x_2, \ldots, x_n]$ [9].

Let $f_1, f_2, \ldots, f_s$ be polynomials in $\Re[x_1, x_2, \ldots, x_n]$. Then the set

$$\langle f_1, \ldots, f_s \rangle = \{\sum_{i=1}^{s} h_i f_i : h_1, \ldots, h_s \in \Re[x_1, \ldots, x_n]\}$$

is called the ideal generated by $f_1, f_2, \ldots, f_s$.

Definition 1 A real variety $V \subset \Re^n$ 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 $V_2 = V$ [9]. □

It is a known fact [9] that any real variety V can be written as a finite union $V = V_1 \cup V_2 \cup \ldots \cup V_m$, where each V_i is an irreducible variety. Another standard result in algebraic geometry is that 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 $\Re^n$ must stabilise. That is, there exist a positive integer N such that $V_N = V_{N+1} = \ldots$ ($I_N = I_{N+1} = \ldots$). For the relationship between ideals and varieties, or the so called algebra-geometry dictionary, see Chapter 4 of [9].

Definition 2 Fix a monomial order (see [9]). 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 $f_i \in I$. $\square$

For the properties and applications of Gröbner bases see [9]. Care should be taken when using Gröbner bases since most results hold only over an algebraically closed field. Given a set of polynomials $f_1, \ldots, f_n$, we denote their Gröbner basis as $G = Gbasis[f_1, \ldots, f_n]$.

The logical operators "AND" and "OR" are respectively denoted as $\wedge$ and $\vee$. Given two sets $A, B \in \Re^n$, with $B \subset A$, we denote the complement of B with respect to A as $A - B$. We denote the composition of function f in (1) as $f_{u(1)} \circ f_{u(0)}(x(0)) = f(f(x(0), u(0)), u(1))$. A sequence of controls is denoted as $\mathcal{U} = \{u(0), u(1), \ldots\}$ and its truncation to a sequence of length $p+1$ is denoted as $\mathcal{U}_p = \{u(0), u(1), \ldots, u(p)\}$. For longer sequences of controls $\mathcal{U}_p$ we use the notation $f^{\mathcal{U}_p}(x(0)) = f_{u(p)} \circ \ldots \circ f_{u(0)}(x(0))$. The state of system (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 $x(p+1, x(0), \mathcal{U}_p) = f^{\mathcal{U}_p}(x(0))$.

Definition 3 The system (1) is state dead beat controllable if for any initial state $x(0) \in \Re^n$ there exists a control sequence $\mathcal{U}$ and a positive integer ν such that $x(p+1, x(0), \mathcal{U}_p) = 0$, $\forall p \geq \nu$, where $\mathcal{U}_p$ represents the truncation of the sequence $\mathcal{U}$. $\square$

Definition 4 The system (1) is output dead beat controllable if for any initial state $x(0) \in \Re^n$ there exists a control sequence $\mathcal{U}$ and a positive integer ν such that $h(x(p+1, x(0), \mathcal{U}_p)) = 0$, $\forall p \geq \nu$, where $\mathcal{U}_p$ represents the truncation of the sequence $\mathcal{U}$. $\square$

3 QEPCAD based DB controllability tests

We show here that an algorithmic approach to deciding state/output DB controllability problem of a large class of systems (1) is possible. A symbolic computation package called QEPCAD is instrumental in automating this approach.

It is a well known fact that given the equation $a_2(t)u^2 + a_1(t)u + a_0(t) = 0$ in two parameters u and t, there exists a real solution for the variable u if and only if the discriminant $a_1(t)^2 - 4a_2(t)a_0(t) \geq 0$. Hence, we have a condition on the parameter t alone, which guarantees the existence of a real solution u. The Sturm Theorem [21] establishes a similar result for any univariate polynomial $f(u)$ to have a real root. Tarski's Theorem further generalised this idea [21]:

Theorem 1 *Let φ be a finite set of polynomial equations, inequations and inequalities of the form $F(t_1, \ldots, t_r, x_1, \ldots, x_n) = 0, G(t_1, \ldots, t_r, x_1, \ldots, x_n) \neq 0, H(t_1, \ldots, t_r, x_1, \ldots, x_n) > 0$ where $F, G, H \in Z[t_1, \ldots, t_r, x_1, \ldots, x_n]$. 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.* $\square$

Although theoretically very important, Tarski's method is highly impractical for computation of the conditions ψ_j in the above Theorem.

It was not until 1973 that a more practical method for computing ψ_j was found. CAD (Cylindrical Algebraic Decomposition) [5, 6] was first discovered by Collins in 1973 and since then a number of improvements have been reported in the literature. QEPCAD[1] (Quantifier Elimination by Partial Cylindrical Algebraic Decomposition) [7] is the name of a software program which implements CAD. It represents a part of a quantifier elimination procedure for real closed fields and is based on the SACLIB package which was developed by G. Collins and a number of other researchers (for a more detailed description of the algorithm see [5, 6, 7]). QEPCAD is probably the only method for computing ψ_j in Tarski's Theorem, which has a software implementation.

QEPCAD was found to be useful in motion planning [6], bang-bang control [15] and we show below that it can also be used in deciding state and output DB controllability of a large class of polynomial systems given by:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)), \\ y(k) &= h(x(k)) \end{aligned} \tag{2}$$

where $x(k) \in \Re^n$, $y(k) \in \Re$ and $u(k) \in \Re$ are respectively the state, the output and the input of the system (2) at time k. The vector $f(x, u) = (f_1(x, u) \ldots f_n(x, u))^T$ is such that $f_i(x, u) \in Q[x, u] = Q[x_1, x_2, \ldots, x_n, u]$ and $h \in Q[x_1, \ldots, x_n]$, which means that we assume

[1]QEPCAD is still not available in a commercial computation package. It is still being perfected.

f_i and h have rational coefficients. We assume for in this sub section that $f(0,0) = 0$, but do not require the same assumption in the remainder of the paper.

The class of systems (2) represents a very large subclass of (1). In fact, in practise we always deal with (2) since any irrational coefficients are approximated with a desired accuracy by rational coefficients. We denote the set of states that can be transferred to the origin in $k+1$ time steps as:

$$S_k = \{x : \exists u(0), \ldots, u(k) \in \Re, \text{ such that } f_{u(k)} \circ \ldots f_{u(0)}(x) = 0\}$$

3.1 State dead beat controllability

It is obvious that QEPCAD can be used to decide whether a set of polynomial equations has a common real root. Moreover, we can obtain conditions on coefficients of these polynomials which guarantee the existence of a real root. As a result, we can compute the sets $S_0,\ S_1,\ \ldots$ using QEPCAD. First, we find the composition $f_{u(k)} \circ \ldots f_{u(0)}(x)$. Then, we apply QEPCAD to obtain $S_k, k = 0, 1, 2, \ldots$ by considering the following decision problem

$$\text{find } x \in \Re^n \text{ such that } \exists u(0), \ldots, u(k) \in \Re, \text{ which yields } f_{u(k)} \circ \ldots f_{u(0)}(x) = 0$$

In other words, QEPCAD is used to project the variety $V(f_{u(k)} \circ \ldots f_{u(0)}(x)) \subset \Re^{n+k+1}$ onto the space $\Re^n$ where the vector x lives. It is obvious that $S_k \subseteq S_{k+1}, \forall k$ and we have a chain $S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots$.

The following theorem follows easily from the above construction:

Theorem 2 *Suppose that there exists an integer N such that $S_N = S_{N+1}$. The system (2) is state DB controllable if and only if $S_N = \Re^n$.* $\Box$

Comment 1 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 [24]). However, even when the chain does not terminate, obtaining a characterisation of the sets S_k is important in its own right and may be used in the design of control laws, such as minimum time DB controllers.

Comment 2 The approach is very computationally expensive for general polynomial systems (2). The bounds on the computation time for the original algorithm can be found in [5] and the improvements are discussed using some examples in [6]. The computation time increases rapidly with the increase of the number of variables and total degrees of the input polynomials. Hence, we need more time to compute the set S_{k+1} than the set S_k. However, the examples that are presented below show that for moderate total degrees of polynomials and low order systems this approach may yield satisfactory answers.

Comment 3 It is not difficult to include bounds on controls in the QEPCAD based state DB 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. However, instead of checking the controllability on $\Re^n$ we may need to work on a bounded subset of the state space $\mathcal{B} \subset \Re^n$, that is we check whether $S_N \supseteq \mathcal{B}$. Indeed, even in the case of linear systems with bounded controls the chain $S_0 \subset S_1 \subset \ldots$ may not terminate for DB controllable systems [28]. In other words, there may be no uniform bound on the DB time. This generalises the approach of Desoer and Wing for minimum time DB control of linear systems with bounded controls [28] and Evans' controllability of linear systems with positive controls [12].

Comment 4 It is easy to see that a family of state DB controllers can be designed using this approach. Indeed, assume that the system under consideration is state DB controllable, that is $S_N = \Re^n$. The sets $S_k, k = 0, 1, \ldots, N$ are defined by:

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

where $t_{i,j} \in \Re[x_1, \ldots, x_n]$ are polynomials and

$$m_{i,j}^k \in \{\leq, \geq, >, <, =, \neq\},\quad j = 1, \ldots, r_i^k,\; i = 1, \ldots, p^k,\; k = 0, 1, \ldots, N.$$

For example, the defining expression for S_k may have form $((x_1^2 + x_2 = 0) \bigwedge (x_2 \geq 0)) \bigvee (x_1 - x_2 + 1 < 0)$. From the definition of sets S_k it follows that $\forall x \in S_{k+1}, \forall k = 0, \ldots, N-1,\ \exists u(x) \in \Re$ such that $\bigvee_{i=1}^{p^k} (\bigwedge_{j=1}^{r_i^k} t_{i,j}^k(f(x, u(x))) \;\; m_{i,j}^k \;\; 0)$. Hence, once we have obtained the defining expressions for the sets S_k we can easily find controllers which are such that they map S_{k+1} to S_k, $\forall k = 0, 1, \ldots, N-1$. Notice that the control $u(x)$, which transfers the state x from S_{k+1} to S_k, may not be unique. In this way we obtain a family of all minimum time DB controllers, which are by construction discontinuous and nonlinear state feedback controllers.

From Theorem 2 we have the following state DB controllability test.

TEST 1: 0. Input: f(x,u)

1. Let $k = 0$. Find the set S_0 using QEPCAD by considering $\exists u(0) \in \Re, f(x, u(0)) = 0$.

2. $k = k + 1$

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

$$\exists u(i) \in \Re, i = 0, 1, \ldots, k, \; f_{u(k)} \circ \ldots f_{u(0)}(x) = 0.$$

Compare whether $S_k = S_{k-1}$ (using QEPCAD in general). If not, go to 2. If yes, go to 4.

4. If $S_k = \Re^n$, the system is state DB controllable. If $S_k \neq \Re^n$, the systems is not state DB controllable.

It is important to notice that an infinite loop may occur in the above algorithm if $S_{k+1} \neq S_k, \forall k$.

3.2 Output dead beat controllability

A very similar procedure can be used to deal with output DB controllability and control of (2). The only difference between the two approaches is that in the case of state DB control the target set T, to which we want to control the state of the system, is the origin whereas here we need to compute it. As a result, the output DB controllability test can be split into two parts. Each part of the algorithm is based on the use of QEPCAD and the obtained controllability test is in general more expensive than TEST 1.

PART 1: The computation of the target set T is done below. It is obvious that any state in the variety $T_0 = \{x : h(x) = 0\}$ potentially belongs to the target set. However, we are interested in the subset of T_0 for which there exists a control u which keeps the state in T_0. In other words, we compute

$$T_1 = \{x : \exists u(0) \in \Re \text{ such that } h \circ f_{u(0)}(x) = 0\} \cap T_0$$

The set T_2 is defined as follows:

$$T_2 = \{x : \exists u(0), u(1) \in \Re \text{ such that } h \circ f_{u(1)} \circ f_{u(0)}(x) = 0\} \cap T_1$$

We continue the same procedure and if we obtain $T_L = T_{L+1}$, we define the set $T_L = T$ as the target set. Notice that we have $T_{k+1} \subset T_k, \forall k$ and in general the chain $T_0 \supset T_1 \supset \ldots$ may not terminate. On the target set T we have that the output is zero and moreover for any initial state in T we can find a sequence of controls which keeps the state in T for all future time steps (see Definition 4). The target set is given by

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

where $m_{i,j} \in \{\leq, \geq, <, >, =, \neq\}$.

PART 2: If an initial state is in T, the output is zero and it can be kept at zero for all future times. So we can denote this set as $S_0^O = T$. The set of states that are not in T but that can be transferred in one step to T is denoted:

$$S_1^O = \{x : \exists u(0) \in \Re \text{ such that } \bigvee_{i=1}^{P} (\bigwedge_{j=1}^{R_i} t_{i,j} \circ f_{u(0)}(x) \;\; m_{i,j} \;\; 0)\}.$$

and it can be obtained using QEPCAD. Similarly, by using QEPCAD we find the set

$$S_2^O = \{x : \exists u(0), u(1) \in \Re \text{ such that } \bigvee_{i=1}^{P} (\bigwedge_{j=1}^{R_i} t_{i,j} \circ f_{u(1)} \circ f_{u(0)}(x) \;\; m_{i,j} \;\; 0)\}, etc.$$

Then we can state the following

Theorem 3 *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 DB controllable if and only if $S_L^O = \Re^n$.*
□

The proof of the Theorem is obvious.

Comment 5 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 easily compute the following subset of the target set:

$$T^* = \{x : h(x) = 0 \text{ and } \exists u \in \Re \; x = f(x,u)\}$$

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

The following output DB controllability test is obtained from the above given procedures.

TEST 2

1. (a) Let $k = 0$ and $T_0 = V(h)$.

   (b) $k = k + 1$

   (c) Find composition $h \circ f_{u(k)} \circ \ldots \circ f_{u(0)}(x)$ and find the set T_k using QEPCAD:

   $T_k = \{x \in \Re^n : \exists u(i) \in \Re, i = 0, 1, \ldots, k \text{ such that } h \circ f_{u(k)} \circ \ldots \circ f_{u(0)}(x) = 0\} \cap T_{k-1}$

   If $T_k = T_{k-1}$, we define $T = T_k$ and go to 2. If $T_k \neq T_{k-1}$, go to 1.(b).

2. We have that $T = \{x : \bigvee_{i=1}^{P}(\bigwedge_{j=1}^{R_i} t_{i,j}(x) \;\; m_{i,j} \;\; 0)\}$.

   (a) Let $k = 0$ and compute $S_0^O \subset \Re^n$ using QEPCAD:

   $$S_0^O = \{x \in \Re^n : \exists u(0) \in \Re \text{ such that } \bigvee_{i=1}^{P}(\bigwedge_{j=1}^{R_i} t_{i,j} \circ f_{u(0)} \;\; m_{i,j} \;\; 0)\}$$

   (b) $k = k + 1$

   (c) Compute the set S_k^O using QEPCAD:

   $S_k^O = \{x \in \Re^n : \exists u(s) \in \Re, s = 0, 1, \ldots, k, \text{ such that } \bigvee_{i=1}^{P}(\bigwedge_{j=1}^{R_i} t_{i,j} \circ f_{u(k)} \circ \ldots \circ f_{u(0)}(x) \;\; m_{i,j} \;\; 0)\}$

   If $S_k^O = S_{k-1}^O$ go to 2.(d). If $S_k^O \neq S_{k-1}^O$, go to 2.(b).

   (d) If $S_k^O = \Re^n$, the system is output DB controllable and vice versa.

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

The outlined QEPCAD based approach can be regarded as a straight forward and unified approach to state/output DB controllability and control of polynomial systems (2). However, the main hindrance to its implementation is the obvious computational complexity of the problem. It is possible to reduce the complexity of the problem by either requiring less information about S_k (not a complete description) or constraining the structure of the system (2). Although it is plausible in certain situations to require less information about sets S_k, 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. Sections 4 and 5 show how constraining the structure of (2) may reduce the computational complexity of the controllability test or even be used to obtain finitely computable conditions for controllability.

We present below several nontrivial examples, solved using QEPCAD.

Example 1 Consider the scalar polynomial system:

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

The set S_0 can be computed by using QEPCAD. We actually compute for which states $x \in \Re$ there exists $u \in \Re$ such that $xu^6 + (x+1)u^3 - 2u^2 + 3xu + 2x = 0$. QEPCAD computed the set S_0 in 1.2 sec[2]:

$$\begin{aligned} S_0 \quad = \quad & \{x : 4123953x^7 + 13719780x^6 + 7007148x^5 - 2009664x^4 + 382968x^3 + \\ & 901620x^2 - 130208x - 1728 \leq 0 \ \vee \ x \geq 0\} \end{aligned}$$

The input formula for the set S_1 is the composition $f_{u(1)} \circ f_{u(0)}$. 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 $\Re$ 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 the set S_1 after more than 9 hours.

Notice that using results of [24] we could decide on DB controllability after computing the set S_0 only, which took 1.2 seconds to compute. This shows that instead of using straight forward computation of all S_k's, that is proposed, we sometimes may require less information to conclude on DB controllability. It is our opinion 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.

It is important to emphasize that QEPCAD is an interactive program and solving non trivial problems requires a detailed knowledge of its operation. For instance, in this case it

[2]Examples 1, 2 and 3 are computed using a DECstation 5000/240 with a 40 MHz R3400 risc-processor, by G. Collins and C. Brown.

is not too difficult to see that the interval $]-\infty,-3] \subset S_0$ and let us compute which states can be transferred to this interval in one step by using $xu^6+(x+1)u^3-2u^2+3xu+2x < -3$. It was obtained that this is true for any $x \in \Re$. Hence, $S_1 = \Re$. 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) the answer could be obtained using QEPCAD. Although this case-by-case approach is not plausible to use in general, we believe that for certain classes of systems it may be successfully imbedded in the controllability test. □

Example 2 Consider the generalised Hammerstein system:

$$\begin{pmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & -2 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u^2(k) \tag{4}$$

By using the procedure outlined in TEST 1 we obtain:

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

The computation time for sets S_0, S_1 and S_3 is respectively 0.34 sec, 0.517 sec and 133 sec. A minimum time state DB (feedback) controller is given below:

$$u(x) = \text{any real root } u \text{ to} \begin{cases} (x_1+u=0) \wedge (-x_2-2x_3+u^2=0) & \text{,if } x \in S_0 \\ (x_3-(x_1+u)^2)=0) \wedge (-x_2-2x_3+u^2=0) & \text{,if } x \in S_1 - S_0 \\ (2(-x_2-2x_3+u^2)+x_3 \geq 0) \wedge & \text{,if } x \in \Re^3 - S_1 \\ (2x_3(-x_2-2x_3+u^2)+(-x_2-2x_3+u^2)^2 & \\ -6(x_1+u)^2(-x_2-2x_3+u^2)+x_3^2- & \\ 2x_3(x_1+u)^2+(x_1+u)^4=0) & \end{cases} \tag{5}$$

Obviously, 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 $x \in \Re^3 - S_1$. 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, we obtain different minimum time state dead beat controllers.

Notice that the given controller globally stabilises the system in this case. It is not difficult to see that as $||x|| \to 0$ then the required control in (5) $|u| \to 0$. As a result, $\forall \varepsilon > 0$

we can find much smaller $\delta > 0$ such that if $||x(0)|| < \delta$ then $||x(1)|| < \varepsilon$ and $||x(2)|| < \varepsilon$ which implies $||x(k)|| < \varepsilon, \forall k$ since $x(k) = 0, k \geq 2$. $\square$

Example 3 Consider the third order bilinear systems:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_3(k) \\ x_3(k+1) &= x_1(k) + x_3(k) - (x_1(k) + x_2(k) - x_3(k))u(k) \end{aligned} \tag{6}$$

The computed sets S_k are given below:

$$\begin{aligned} S_0 &= \{x : x_2 = 0 \ \wedge \ x_3 = 0\} \\ S_1 &= \{x : x_3 = 0 \ \wedge \ x_3 - x_2 - x_1 = 0\} \\ S_2 &= \{x : x_3 - x_2 - x_1 \neq 0 \ \wedge \ x_3 + x_2 \neq 0\} \\ S_3 &= \{x : (x_2 - x_1 \neq 0 \ \wedge \ 2x_3 + x_1 \neq 0) \ \vee \ x_3 - x_2 - x_1 \neq 0\} \\ S_4 &= \Re^3 \end{aligned} \tag{7}$$

And hence the system is state DB controllable. Notice that we could conclude on DB controllability using the results in [10, 17] 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 modulo 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. We may use the computationally less complex test presented in the next section to check state DB controllability of this class of systems.

Example 4 Check whether the system:

$$\begin{aligned} x_1(k+1) &= x_2(k) + u(k) \\ x_2(k+1) &= -x_1(k) + u^2(k) \\ y(k) &= x_2(k) \end{aligned} \tag{8}$$

is output DB 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 $\Re^2$ that can be transferred to T in one step:

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

Similarly, we have that

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

and therefore the system is output DB controllable. $\Box$

4 Invariant sets and output dead beat controllability: the Gröbner basis method

The methodology in the previous section can be applied to a large class of polynomial systems but the computational requirements may be formidable. If the structure of the polynomial system is appropriately constrained, we may obtain computationally less expensive controllability tests. The purpose of this and the following sections is to show two situations when this is possible. We illustrate the tradeoff between the generality of the proposed methods and the computational requirements.

The systems that we consider in this section are given by (2) and the following two assumptions.

Assumption 1 Consider the composition

$$h \circ f_u(x) = h(f(x,u)) = a_m(x)u^m + \ldots + a_0(x) \tag{9}$$

m in the equation (9) is an odd integer. $\Box$

Assumption 2 $\forall x \in V(h), \ \exists u \in \Re$ such that $h \circ f_u(x) = 0$. $\Box$

Systems (2) with Assumptions 1 and 2 are referred to hereafter as **odd systems**. By using Assumption 1 we restrict our consideration to systems whose output $y(k+1)$ is affected by $u(k)$ (one time delay from input to output is present). However, generalisation of our results to systems of arbitrary time delay is straightforward . Assumption 2 is technical and there are systems of interest that do not satisfy it. However, it simplifies the consideration of output DB controllability. It implies that the target set T (see the previous Section) is defined as $T = V(h)$. Assumption 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 [3, 20, 18, 4].

Definition 5 The target set $T = V(h)$ is denoted here as V_O and is called the zero output variety. $V_C = V(a_m)$ is referred to as the critical variety. □

Definition 6 An invariant set $V_{I_j} \subseteq V_C$ is such that $\forall x \in V_{I_j},\ \forall u \in \Re, f(x,u) \in V_{I_j}$. The union of all invariant sets is called the maximal invariant set $V_I = \cup_j V_{I_j}$. □

We show below 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 DB controllability of odd polynomial systems. The set of states from which it may not be possible to zero the output is contained in the critical variety V_C (see equation (9)). 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. Before we state the theorem notice that the following compositions can be regarded as polynomials in $u(0), \ldots, u(k)$ whose coefficients are polynomials in x:

$$\begin{aligned}
a_m \circ f_{u(0)} &= \sum_{i=0}^{m_1} b_i(x) u(0)^i \\
a_m \circ f_{u(1)} \circ f_{u(0)} &= \sum_{i_1=0, i_2=0}^{m_2, p_2} b^2_{i_1,i_2}(x) u(0)^{i_1} u(1)^{i_2} \\
\ldots & \quad \ldots \\
a_m \circ f_{u(k)} \circ \ldots \circ f_{u(0)} &= \sum_{i_1=0,\ldots,i_{k+1}=0}^{m_{k+1}, p_{k+1}, \ldots, l_{k+1}} b^{k+1}_{i_1,\ldots,i_{k+1}}(x) u(0)^{i_1} \ldots u(k)^{i_{k+1}}
\end{aligned} \tag{10}$$

Theorem 4 *The maximal invariant set $V_I \subseteq V_C$ may be computed by the following algorithm:*

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

1. *Initialise:*

   Input: $a_m(x),\ f(x,u)$; $G_0 = \langle a_m \rangle$; $k = 0$

2. *Iterate:* $k = k + 1$

3. *Compute* $a_m \circ f_{u(k-1)} \circ \ldots \circ f_{u(0)}(x)$.

4. Compute the Gröbner basis G_k:

$$G_k = Gbasis[a_m, b_0^1, \ldots, b_{m_1}^1, b_{00}^2, \ldots, b_{m_2,p_2}^2, \ldots, b_{m_{k+1},p_{k+1},\ldots,l_{k+1}}^{k+1}]$$

where the polynomials $b_{i_1,\ldots,i_{s+1}}^{s+1} \in Q[x], s = 1, \ldots, k$ are defined in (10).

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

The algorithm stops after a finite number of iterations.

Proof: Notice that $V_I \subseteq V_C$. The set of all critical states is defined by the ideal $I_1 = \langle a_m \rangle$. Consider now 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 $a_m \circ f_{u(0)}(x) = 0, \forall u(0) \in \Re$. Notice that the composition of two polynomials is a polynomial and therefore we have $a_m \circ f_{u(0)}(x) = b_{m_1}^1(x)u(0)^{m_1} + \ldots + b_1^1(x)u(0) + b_0^1(x)$. This polynomial is identically equal to zero for all $u(0)$ if and only if the polynomials $b_i^1(x) = 0, \forall i = 0, 1, \ldots, m_1$. Therefore, the points that are mapped to V_C in the first step, regardless of the control action taken, are defined by the ideal $I_2 = \langle a_m, b_{m_1}^1, \ldots, b_0^1 \rangle$. 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 $a_m \circ f_{u(1)} \circ f_{u(0)}(x) = b_{m_2 p_2}^2(x)u(0)^{m_2}u(1)^{p_2} + \ldots + b_{00}^2(x)$ is a polynomial in all its arguments and is identically equal to zero $\forall u(0), u(1) \in \Re$ 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 $I_3 = \langle a_n, b_{m_1}^1, \ldots, b_0^1, b_{00}^2, \ldots, b_{m_2 p_2}^2 \rangle$, 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. Two sets of polynomials define the same ideal if and only if their reduced Gröbner basis is the same [9]. In step i we need to compute the Gröbner basis of I_i and compare it to the reduced Gröbner basis in the previous step. The chain of ideals necessarily has got finite length, say N. Since points 4 and 5 of the algorithm compute the Gröbner basis of a set of polynomials, we conclude from [9, pg. 89] that the algorithm stated in Theorem 4 terminates after a finite number of iterations. Q.E.D.

Comment 6 We emphasize that the algorithm given in Theorem 4 can be used to find an invariant set of any variety defined by $V(f_1, \ldots, f_c), f_i \in Q[x_1, \ldots, x_n]$. 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 $V(f_1) = V(0) = \Re^n$ is invariant. However, in this section we are interested only in the invariant sets of V_C since they can be used to characterise output DB controllability of odd systems.

We need the following definition:

Definition 7 The trivial invariant set $V_T \subseteq V_I$ is such that for any initial state $x(0) \in V_T$ there exists a sequence of controls $u(0), u(1), \ldots$ which transfers the initial state $x(0)$ to the zero output variety V_O in finite time. □

Comment 7 The trivial invariant set can be computed using the QEPCAD algorithm. 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. We can compute (using QEPCAD) the subset of V_I from which we can reach the zero output variety in one step:

$$S_1^T = \{x \in \Re^n : \exists u(0) \in \Re,\ f_1 = 0, \ldots, f_s = 0, f_1 \circ f_{u(0)}(x) = 0, \ldots, f_s \circ f_{u(0)}(x) = 0\}$$

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 trivial invariant set V_T and the maximal invariant set V_I determine output DB controllability of odd systems for which Assumption 2 holds. It is obvious that the following is true:

Theorem 5 *An odd polynomial system is output DB controllable if and only if $V_I = V_T$.* □

From Theorems 4 and 5 it is easy to deduce the following output DB controllability test for odd polynomial systems.

TEST 3:

1. Check whether Assumptions 1 and 2 are satisfied. Assumption 2 is checked using QEPCAD by first computing the set

$$h^* = \{x \in \Re^n : \exists u(0) \in \Re, \quad h = 0, \ h \circ f_{u(0)}(x) = 0\}$$

   and then comparing whether $h = h^*$.

2. Compute defining equations for V_I using the procedure from Theorem 4 given below:

   (a) Compute $h \circ f_u(x)$ and let $k = 0, G_0 = \langle a_m \rangle$.

   (b) k=k+1

   (c) Compute $a_m \circ f_{u(k)} \circ \ldots \circ f_{u(0)}(x)$.

   (d) Find the Gröbner basis $G_k = Gbasis[a_m, b_0^1, \ldots, b_{m_1}^1, ldots, b_{m_{k+1}, p_{k+1}, \ldots, l_{k+1}}^{k+1}$. where polynomials $b_{i_1, i_2, \ldots, i_s}^{s+1}$, $s = 1, 2, \ldots, k$ are defined in (10).

   (e) If $G_k = G_{k-1}$ go to step 3. If not, go to (b).

3. Find (using QEPCAD) the set of common real solutions for the set of defining polynomials for G_k. This set of solutions defines V_I. 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 7). 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 8 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 and all the deficiencies of this method were discussed in the previous section. We remark that each of the sets S_k^T is finitely computable [5, 6] but in general the trivial invariant set is not finitely computable.

Notice that in the steps 1 and 3 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 2) that the computation requirements are less hindering since few compositions of functions are required and the total degrees of the input polynomials are small. Furthermore, the number of variables in the input polynomials for step 1 is $n+1$ and for step 3 is n, whereas in step 4 it increases and may be much larger than the order of the system n.

We emphasize that the class of odd systems is inherently simpler than the systems with rational coefficients (2) considered in the previous section since QEPCAD only needs 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 9 We can use the above given procedure to check output DB 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 $V_O = V(h - y^*)$.

The following two corollaries are easy consequences of Theorem 5 and they may help us to decide on output DB controllability without resorting to QEPCAD.

Corollary 1 Assume $V_I \neq \emptyset$. Then, the odd system is output DB controllable only if $V_O \cap V_I \neq \emptyset$. □

Notice that using the above given Corollary we may only conclude that the system is not output DB controllable.

Corollary 2 The odd system is output DB controllable if $V_I = \emptyset$. □

We now present a practical example of an odd system and illustrate our test for output DB controllability.

Example 5 Modelling and identification of a column type grain dryer is studied respectively in [19] and [20]. The mathematical model of the sub system, which relates the uppermost measured temperature $y_1 = y$ and the productivity of the exhaust grain mechanism $u_1 = u$, was identified in [20] and is given by:

$$\begin{aligned} y(k+1) &= 1.6389y(k) - 0.4397y(k-1) - 0.1803y(k-2) \\ &\quad -0.0082u(k)y(k) - 0.0042u(k-1)y(k-1) - 0.0074u(k-2)y(k-2) \\ &\quad +0.0019u(k) - 0.0041u(k-1) + 0.0021u(k-2) \end{aligned} \tag{11}$$

which is a so called BARMA (bilinear ARMA) model. We can check output DB controllability of this sub system using the methodology developed in this Section. For this purpose we introduce state variables:

$$x_1(k) = y(k)$$

$$
\begin{aligned}
x_2(k) &= -0.4397y(k-1) - 0.1803y(k-2) - 0.0042u(k-1)y(k-1) \\
&\quad -0.0074u(k-2)y(k-2) - 0.0041u(k-1) + 0.0021u(k-2) \\
x_3(k) &= -0.1803y(k-1) - 0.0074u(k-1)y(k-1) + 0.0021u(k-1)
\end{aligned}
$$

and we obtain a non homogeneous bilinear system:

$$
\begin{aligned}
x(k+1) &= \begin{pmatrix} 1.6389 & 1 & 0 \\ -0.4397 & 0 & 1 \\ -0.1803 & 0 & 0 \end{pmatrix} x(k) + u(k) \begin{pmatrix} -0.0082 & 0 & 0 \\ -0.0042 & 0 & 0 \\ -0.0074 & 0 & 0 \end{pmatrix} x(k) \\
&\quad + \begin{pmatrix} 0.0019 \\ -0.0041 \\ 0.0021 \end{pmatrix} u(k) \\
y(k) &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x(k)
\end{aligned} \tag{12}
$$

Now we use the output DB controllability test of this Section:

STEP 1: We first check Assumptions 1 and 2. Consider the expression

$$
y(k+1) = 1.6389x_1(k) + x_2(k) + (0.0019 - 0.0082x_1(k))u(k).
$$

Assumption 1 is satisfied since 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 $V_O = V(x_1 - y^*)$. We have that $\forall x(0) \in V_O$ there exists control $u(0)$ which yields $x(1) \in V_O$ if $y^* \neq 19/82$. Therefore, Assumption 2 is also satisfied for all set points $y^* \in \Re - \{19/82\}$ and we can apply the methods from this Section. Thus, it is assumed that we want to control the temperature y to a set point $y^* \neq 19/82$. We have that $a_m(x) = 0.0019 - 0.0082x_1$.

STEP 2: In order to compute the maximal invariant set, we compute the compositions:

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

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 that the ideals that define varieties V_C, V_1 and V_2 are respectively:

$$
G_0 = Gbasis[19 - 82x_1] = \langle 19 - 82x_1 \rangle
$$

$$
\begin{aligned}
G_1 &= Gbasis[19 - 82x_1, 19 - 134x_1 - 82x_2, 1558 - 6724x_1] \\
&= \langle -19 + 82x_1, 247 + 1681x_2 \rangle \\
G_2 &= Gbasis[19 - 82x_1, 19 - 134x_1 - 82x_2, 1558 - 6724x_1, 19 - 184x_1 \\
&\quad -134x_2 - 82x_3, 14430x_1 + 816, -5514x_1 + 12776, 1558 + 11020x_1 \\
&\quad +6720x_2] = \langle 1 \rangle
\end{aligned}
$$

It follows that $G_3 = G_2 = \langle 1 \rangle$ and hence $V_2 = V(G_3) = \emptyset$. As a result, we have $V_I = \emptyset$ and we do not have to do last two steps of the test. Indeed, according to Corollary 2 the system (12) is output DB controllable to any set point $y^* \in \Re - \{19/82\}$. $\square$

5 Reachable sets and output dead beat controllability

In this Section we further specialise the structure of odd polynomial systems with the aim of obtaining a simpler and finitely computable output DB controllability test. As a bonus, a minimum time output DB controller is obtained using the method. The results in this section generalises [25].

The class of systems that we consider in this Section are given below:

$$y(k+1) = F(y(k), u(k-1), u(k-2), \ldots, u(k-n+1), u(k)) \tag{13}$$

where y and u are respectively the output and input of the system and $F[y, v_1, v_2, \ldots, v_{n-1}, u] \in Q[y, v_1, v_2, \ldots, v_{n-1}, u]$. 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:

$$
\begin{aligned}
x_1(k+1) &= F(x_1(k), x_2(k), \ldots, x_n(k), u(k)) \\
x_2(k+1) &= u(k) \\
x_3(k+1) &= x_2(k) \\
\cdots & \quad \cdots \\
x_n(k+1) &= x_{n-1}(k) \\
y(k) &= x_1(k)
\end{aligned} \tag{14}
$$

We still assume that Assumptions 1 and 2 are satisfied for systems (14). The special forms of $f(x(k), u(k))$ and $h(x(k))$ for systems (14) are obvious.

The notion of *strongly invariant sets* plays a more important role in deciding output DB controllability for systems (14) than the invariant sets introduced in the previous Section.

Definition 8 A set $W_{I_j} \subseteq V_C$ 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 $\mathcal{U}_t = \{u(0), u(1), \ldots, u(t)\}$ which yields $x(t+1) = f^{\mathcal{U}_{t+1}}(x(0)) = x(0)$. The union of all strongly invariant sets $W_I = \cup_j W_{I_j}$ is called the maximal strongly invariant set. $\square$

Definition 9 Polynomials of special form are given by:

$$f_s = x_1 - \sum_{i_2 i_3 \ldots i_n} b_{i_2 i_3 \ldots i_n} x_2^{i_2} x_3^{i_3} \ldots x_n^{i_n}, \ b_{i_2 i_3 \ldots i_n} \in \Re, \forall i_2, i_3, \ldots i_n$$

and varieties $V_s = V(f_s)$ are called varieties of special form. $\square$

Notice that there may be only finitely many varieties of special form that are contained in the critical variety $V_C = V(a_m)$. We denote this number as B. Also, polynomials of special form and the varieties of special form are *irreducible.*

Definition 10 The p-step reachable set $V_r^p(x_0)$ from an initial state $x(0)$ is given by:

$$V_r^p(x(0)) = \{\zeta : \zeta = f_{u(p-1)} \circ \ldots \circ f_{u(0)}(x(0)), u(i) \in \Re, \ \forall i = 0, \ldots, p-1\} \ \square \tag{15}$$

Consider the polynomial:

$$\begin{aligned} f_r(x(n-1), x(0)) &= x_1(n-1) - F_{x_n(n-1)} \circ F_{x_{n-1}(n-1)} \circ \ldots F_{x_3(n-1)} \circ F_{x_2(n-1)}(x(0)) \\ &= x_1(n-1) - \sum_{i_1 i_2 \ldots i_n} B_{i_2 i_3 \ldots i_n}(x(0)) x_2^{i_2}(n-1) x_3^{i_3}(n-1) \ldots x_n^{i_n}(n-1) \end{aligned} \tag{16}$$

This set 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 $\forall x(0) \in \Re^n$ and we can conclude the following [9]:

1. $V_r^{n-1}(x(0))$ is an irreducible variety, $\forall x(0) \in \Re^n$
2. $dim V_r^{n-1}(x(0)) = n-1, \ \forall x(0) \in V_C$
3. $x_1(n-1) - F_{x_n(n-1)} \circ F_{x_{n-1}(n-1)} \circ \ldots F_{x_3(n-1)} \circ F_{x_2(n-1)}(x(0))$ is an irreducible polynomial $\forall x(0) \in \Re^n$

The special structure of the system (14) yields the particular structure of the set $V_r^{n-1}(x(0))$ which can be used to simplify the controllability tests. Indeed, we can prove the following Lemma.

Lemma 1 *The maximal strongly invariant set $W_I \subseteq V_C$ can be decomposed into a finite union of the varieties of special form $W_I = V_{s1} \cup V_{s2} \cup \ldots \cup V_{sL}, L \leq B$ $\square$.*

Proof of Lemma 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 $B_{i_2 i_3 \ldots i_n}(x(0))$ 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 (16) 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} [9, 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 $V_{s1} \rightarrow V_{sk}$.

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 1, we can prove the following Lemmas.

Lemma 2 *Every invariant set must contain a strongly invariant set.* $\square$

Proof: 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 previous Lemma, we have that $V_{s1} \rightarrow V_{si},\ i = 1, 2, \ldots, B$. However, since we assumed that there are no strongly invariant sets in V_I, we must have that $i \neq 1$. Therefore, $V_{s1} \rightarrow V_{si},\ i = 2, \ldots, B$, and we can assume $i = 2$. Using the same argument we have that $V_{s2} \rightarrow V_{si},\ i = 3, \ldots, B$ and we can assume that $i = 3$, etc. After $B - 1$ steps we obtain that $V_{sB} \rightarrow V_{si},\ i = 1, 2, \ldots, B$ 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.

We can prove the following two Lemmas using very similar arguments.

Lemma 3 *Every state in $V_C - V_I$ can be transferred to $\Re^n - V_C$ in finite time.* $\square$

Lemma 4 *Every state in $V_I - W_I$ is transferred to a strongly invariant set W_I in finite time.* $\square$

We can combine these Lemmas 1, 2, 3 and 4, to obtain the following result:

Theorem 6 *The odd polynomial system (14) is output DB 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.* $\square$

Proof of Theorem 6:

Necessity: Suppose that there exists a variety of special form V_s contained i 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 not 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 1 and the same argument applies.

Sufficiency: We partition the whole state space $\Re^n = (V_C - V_I) \cup (V_I - W_I) \cup W_I \cup (\Re^n - V_C)$ and consider what happens on each of the subsets. If $x(0) \in \Re^n - V_C$ we can zero the output in one step. If $x(0) \in V_C - V_I$, according to Lemma 3, it follows that the initial state can be transferred to $\Re^n - V_C$ in finite time and consequently to V_O. Consider $x(0) \in V_I - W_I$. From Lemma 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 2 it follows that any state in V_I can be transferred to V_O in finite time. Because of Assumption 2 we conclude that the system is output DB controllable. Q.E.D.

We have considered what happens geometrically, whereas an algebraic test is needed to check the conditions of Theorem 6. From Lemma 1 and definition of strongly invariant sets, we can deduce the following method to check output DB controllability of systems (14).

TEST 4:

1. Check Assumptions 1 and 2. Assumption 2 is checked using QEPCAD.

2. Decompose the polynomial $a_m \in Q[x_1, \ldots, x_n]$ 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 in the Previous section. Denote the set of all polynomials f_{si} that yield invariant varieties as Σ_1^I. Obviously $\Sigma_1^I \subseteq \Sigma_1$, and find 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 $\Sigma_3 = \Sigma_2 - \Sigma_2^I$, etc.

(c) 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, $\Sigma_B^I = \emptyset$. Find 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. If this is true system is output DB controllable and vice versa.

Comment 10 It is very important to notice that the above given output DB controllability test stops after a finite number of operations. This was not the case with the systems considered in the previous two Sections 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 (14), however, guarantees that the above given test stops in finite time.

The following Corollaries may help us to reduce computations even more.

Corollary 3 *If $dimV_C = dimV(a_m) < n-1$ the system is output DB controllable.* $\square$

Proof: Since $dimV_r^{n-1}(x(0)) = n-1, \forall x(0) \imath \Re^n$, 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 [9]) in order to check the dimension of the variety V_C.

Corollary 4 *If V_C does not contain varieties of special form, that is a_m does not contain irreducible polynomials of special form, the system (14) is output DB controllable.* $\square$

Proof: From properties 1-2 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 5 *Suppose that there are B varieties of special form $V(f_{si})$ contained in V_C. The system (14) is output DB controllable if $V_O \cap V(f_{si}) \neq \emptyset$, $\forall i = 1, 2, \ldots, B$.* $\square$

An important property of the approach that we have taken is that it leads to the design of a minimum time output DB controller. The controller is presented in Figure 1. It is obvious that different control algorithms are performed for states that belong to different sets of the following partition $\Re^n = (\Re^n - V_C) \cup (V_C - V_I) \cup (V_I - W_I) \cup W_I$.

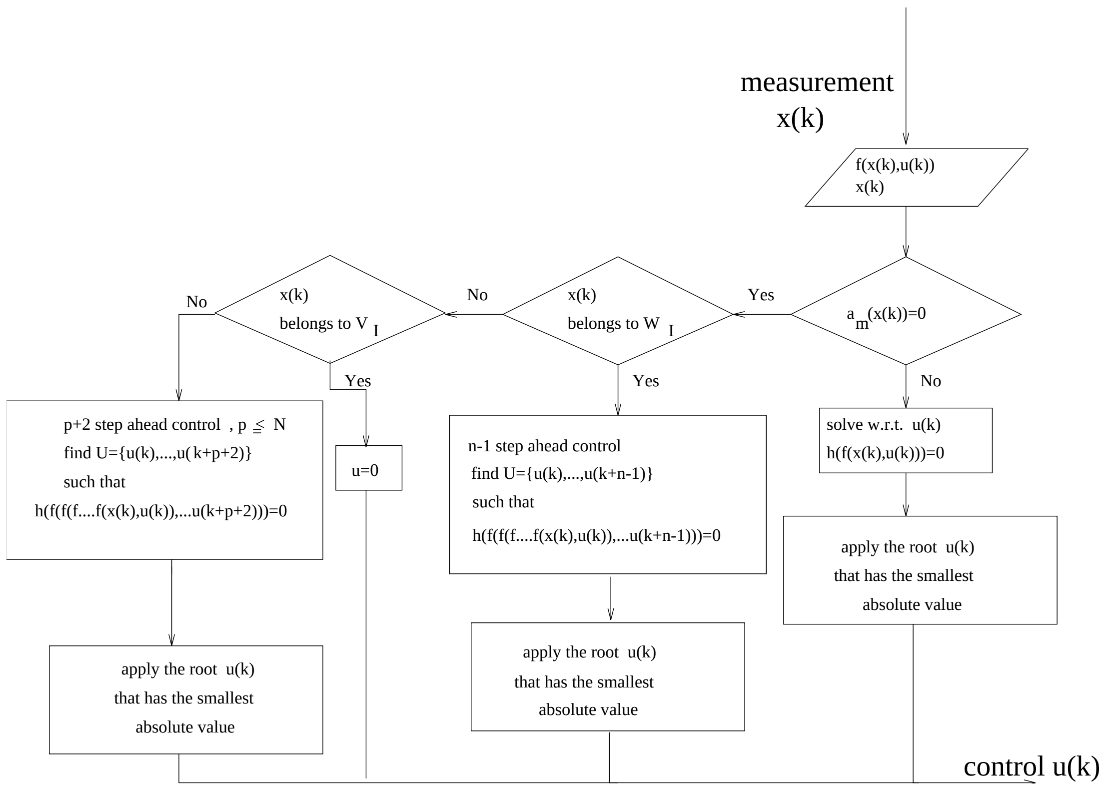

Figure 1: Output DB algorithm for a class of odd polynomial systems

Example 6 Consider the system:

$$\begin{aligned} y(k+1) &= (y^2(k) - 2y(k)u^2(k-1)u^2(k-2) - 3y(k) + u^4(k-1)u^4(k-2) \\ &\quad +3u^2(k-1)u^2(k-2) + 2)u(k)^3 + u(k)^2u(k-1)^2 - y(k) \\ &\quad +u(k-1)^2u(k-2)^2 + 3 \end{aligned} \tag{17}$$

Introduce 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:

$$\begin{aligned} x_1(k+1) &= (x_1^2(k) - 2x_1(k)x_2^2(k)x_3^2(k) - 3x_1(k) + x_2^4(k)x_3^4(k) + 3x_2^2(k)x_3^2(k) + 2)u(k)^3 \\ &\quad +u(k)^2x_2^2(k) - x_1(k) + x_2^2(k)x_3^2(k) + 3 \\ x_2(k+1) &= u(k) \\ x_3(k+1) &= x_2(k) \\ y(k) &= x_1(k) \end{aligned}$$

Step 1: It is easy to see that Assumptions 1 and 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:

$$\begin{aligned}
f_{s1} &= x_1 - x_2^2x_3^2 - 1 \\
f_{s1} \circ f_u(x) &= (x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2)u^3 - x_1 + x_2^2x_3^2 + 2 \\
G_0 &= \langle x_1 - x_2^2x_3^2 - 1 \rangle \\
G_1 &= Gbasis[x_1 - x_2^2x_3^2 - 1, (x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2), -x_1 + x_2^2x_3^2 + 2] \\
&= \langle 1 \rangle
\end{aligned}$$

and since $G_2 = \langle 1 \rangle$ it follows that V_{s1} is not invariant. Similarly, we have for variety V_{s2}:

$$\begin{aligned}
f_{s2} &= x_1 - x_2^2x_3^2 - 2 \\
f_{s2} \circ f_u(x) &= (x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2)u^3 - x_1 + x_2^2x_3^2 + 1 \\
G_0 &= \langle x_1 - x_2^2x_3^2 - 2 \rangle \\
G_1 &= Gbasis[x_1 - x_2^2x_3^2 - 2, (x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2), -x_1 + x_2^2x_3^2 + 1] \\
&= \langle 1 \rangle
\end{aligned}$$

Therefore, V_{s2} is not invariant. Consider now the variety $V(f_{s1} \cdot f_{s2})$. We obtain:

$$\begin{aligned}
f_{s1} \cdot f_{s2} &= (x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2) \\
(f_{s1} \cdot f_{s2}) \circ f_u(x) &= [(x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2)u^3 - x_1 + x_2^2x_3^2 + 2] \\
&\quad [(x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2)u^3 - x_1 + x_2^2x_3^2 + 1] \\
G_0 &= \langle (x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2) \rangle \\
G_1 &= Gbasis[(x_1 - x_2^2x_3^2 - 1)^2(x_1 - x_2^2x_3^2 - 2)^2, (x_1 - x_2^2x_3^2 - 1) \\
&\quad (x_1 - x_2^2x_3^2 - 2)^2, (x_1 - x_2^2x_3^2 - 1)^2(x_1 - x_2^2x_3^2 - 2), (x_1 - x_2^2x_3^2 - 1) \\
&\quad (x_1 - x_2^2x_3^2 - 2)] = \langle (x_1 - x_2^2x_3^2 - 1)(x_1 - x_2^2x_3^2 - 2) \rangle
\end{aligned}$$

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 $V_{s1} \to V_{s2} \to V_{s1} \to \ldots$.

Step 4: We do not need to used QEPCAD in this case. Indeed, since $x_2^2x_3^2 = -K, K = 1, 2$ 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 the systems is not output DB controllable. $\square$

6 Conclusion

It has been shown how some symbolic computation methods can be used to check state/output DB controllability of polynomial systems with rational coefficients. The QEPCAD algorithm provides a computational tool to DB control problems for a very large class of polynomial systems with rational coefficients with possible constraints on controls. The same method can easily be applied to MIMO polynomial systems. However, the computational complexity of the problem is in general prohibitive. As a result, the method is not feasible to use in general. Nevertheless, non trivial examples can be tackled for special sub classes of polynomial systems, such as bilinear or generalised Hammerstein systems. It was shown that for the so called odd polynomial systems the Gröbner basis method can be used together with QEPCAD to decide on output DB controllability. The computations are usually less expensive but even this case may still be infeasible. Finally, by further constraining the structure of odd systems, we derived finitely computable necessary and sufficient conditions for output DB controllability. We note here that a number of issues, such as stability of zero constrained dynamics and robustness of the obtained control laws, remain to be explored.

Acknowledgements: The authors wish to thank professor G.E.Collins and his student C. Brown for solving Examples 1, 2 and 3 in this manuscript using QEPCAD and for drawing to our attention several important references. Also, we would like to thank the anonymous reviewers for their constructive comments.

References

[1] G. Bastin, F. Jarachi and I. M. Y. Mareels, "Dead beat control of recursive nonlinear systems", in *Proc. 32nd Conf. Decis. Contr.*, San Antonio, 1993, pp. 2965-2971.

[2] R. Benedetti and J. J. Risler, *Real algebraic and semi-algebraic sets*, Hermann, 1990.

[3] S. A. Billings and M. B. Fadzil, "The practical identification of systems with nonlinearities," in *Proc. 7th IFAC/IFORS Symp. on Ident. Sys. Par. Estimation*, York, UK, 1985, pp. 155-160.

[4] S. A. Billings, M. B. Fadzil, J. L. Sulley and P. M. Johnson, "Identification of a nonlinear difference model of aa industrial diesel generator", *Mech. Syst. Sign. Processing*, vol. 2, pp. 59-76, 1988.

[5] G. E. Collins, "Quantifier elimination for real closed fields by cylindrical algebraic decomposition", *Lect. Notes Comp. Sci.*, vol. 33, pp. 134-183, 1975.

[6] 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.

[7] G. E. Collins and Hoon Hong, “Partial cylindrical algebraic decomposition for quantifier elimination,” *J. Symbolic Comp.*, vol. 12, pp. 299-138, 1991.

[8] G. E. Collins, Personal correspondence, 1995-96.

[9] D. Cox, J. Little and D. O’Shea, *Ideals, varieties and Algorithms*, Springer-Verlag, 1992.

[10] 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.

[11] M. E. Evans and D. N. P. Murthy, “Controllability of discrete time inhomogeneous bilinear systems”, *Automatica*, vol. 14, pp. 147-151, 1978.

[12] 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.

[13] S. T. Glad, “Output dead-beat control for nonlinear systems with one zero at infinity”, *Syst. Contr. Lett.*, vol. 9, pp. 249-255, 1987.

[14] 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, 1988.

[15] S. T. Glad, “An algebraic approach to bang-bang control”, in *Proc. 3rd Europ. Control Conf.*, Rome, Italy, 1995, pp. 2892-2895.

[16] 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.

[17] O. M. Grasselli, A. Isidori and F. Nicolo, “Dead-beat control of discrete-time bilinear systems”, *Int. J. Contr.*, vol. 32, pp. 31-39, 1980.

[18] R. Haber and H. Unbehauen, “Structure identification of nonlinear dynamic systems-a survey of input/output approaches”, *Automatica*, vol. 26, pp. 651-677, 1990.

[19] U. J. Jaaksoo, E. M. Talvis and J. M. Ummer, "Microprocessor-based grain dryer control," in *Prepr. 3rd IFAC/IFIP Symp. on Software for Computer Control*, Madrid, Spain, 1982, pp. 469-473.

[20] U. Kotta and U. Nurges, "Identification of input-output bilinear systems," in *Proc. 9th IFAC Triennial World Congress*, Budapest, Hungary, 1984, pp. 723-727.

[21] N. Jacobson, *Basic Algebra I*, W.H.Freeman and Company, 1974.

[22] B. Jakubczyk and E. D. Sontag, "Controllability of nonlinear discrete-time systems: a Lie algebraic approach", *SIAM J. Contr. Optim.*, vol. 28, pp. 1-33, 1990.

[23] J. O'Reilly, "The discrete linear time invariant time-optimal control problem-an overview", *Automatica*, vol. 17, pp. 363-370, 1981.

[24] 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.

[25] 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.

[26] D. Nešić and I. M. Y. Mareels, "Invariant sets and output dead beat controllability for odd polynomial systems: the Gröbner basis method", to appear in *Proc. 13th IFAC World Congress*, San Francisco, 1996.

[27] E. D. Sontag, *Polynomial response maps*, Springer-Verlag, Berlin, 1979.

[28] J. Wing and C. A. Desoer, "The multiple-input minimal time regulator problem (general theory)", *IEEE Trans. Automat. Contr.*, vol. 5, pp. 125-136, 1963.