Hybrid Control Systems and Comprehensive Gröbner Bases

Krister Forsman
Department of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

Email: krister@isy.liu.se

1994-02-25, revised 1994-04-15

Technical reports from the Automatic Control group in Linköping are available as UNIX-compressed Postscript files by anonymous ftp at the address 130.236.24.1 (joakim.isy.liu.se).
Hybrid Control Systems and Comprehensive Gröbner Bases

Krister Forsman
Department of Electrical Engineering, Linköping University
S-581 83 Linköping, Sweden
Email: krister@isy.liu.se

Abstract. A class of hybrid systems that can be modeled by polynomial differential equations is investigated. By a hybrid system we mean a system which mixes continuous and discrete variables. All variables in the system have the same time scale, though. The polynomial models have the advantage that constructive methods are available for system analysis. One such method that occurs naturally in this context is so called comprehensive Gröbner bases.

Keywords: hybrid systems, modeling, symbolic computation

1 A Class of Hybrid Systems

Hybrid systems (hs) have attracted a lot of interest lately, most importantly because they occur in many industrial applications. Several frameworks have been suggested for modeling and control of different kinds of hs [1, 2, 14, 16, 18, 19, 20, 21, 22, 23].

In this paper we consider dynamical systems that relate some variables

\[ u = (u_1, \ldots, u_m), \quad x = (x_1, \ldots, x_n), \quad y = (y_1, \ldots, y_p) \]

The variables \( x \) are latent variables, that are not possible to measure whereas \( u, y \) are both possible to measure. An a priori partition of the external variables in inputs and outputs may be unnatural in many cases (see [27]), but in this paper the variables named \( u \) are not only inputs, but, as we will see, they are characterized by their values.

All variables are functions of (ordinary) time, i.e. \( \mathbb{R} \) or \( \mathbb{Z} \). We don’t mix continuous and discrete time in the same system. The dynamic variables are of two types:

- **continuous** variables, that take values in \( \mathbb{R} \)
- **discrete** variables, that take values in a finite set e.g. \( \{0,1\} \).

The systems that we are interested in are described by logical conditions involving differential equations. The focus of our attention is on constructive methods for analysis and we will let this aspect influence the modeling. The control design problem is not addressed.

The first class of hybrid systems we are considering is such that the dynamical variables are partitioned in a particularly simple way: the discrete variables are a subset of \( u \) and all latent variables \( x_i \) are continuous.

**Example 1.1** Consider an electric circuit consisting of passive components and a number of switches, e.g. the one in figure 1. If we pick some of the currents or voltages in the circuit
to be the output $y$ we may wish to determine exactly how the positions of the switches affect the dynamics of the $y$-variables, i.e. we do not wish a description that goes via the latent variables. In the circuit of figure 1 we could ask how the differential equation relating $v$ and $y$ depends on the positions of the switches $u_1$ and $u_2$.

![Figure 1: An electric circuit.](image)

**Example 1.2** Another, more theoretical example, is given by:

If $u_1(t) = 0$ then
\[
\dot{x}_1(t) = u_2(t) \\
u_1(t) = 1 \\
\dot{x}_1(t) = -x_1(t)
\]

If $u_2(t) = 0$ then
\[
\dot{x}_2(t) = x_1(t) \\
u_2(t) = 1 \\
\dot{x}_2(t) = -x_2(t)
\]

$y(t) = x_2(t)$

In the next few sections we will answer the question “What is the relation between the dynamical variables $u_1, u_2$ and $y$?” for this particular example.

We assume that the discrete variables constitute a subset of $u$, say
\[
\overline{u} := (u_1, \ldots, u_M), \quad M \leq m
\]

and that each of $u_1, \ldots, u_M$ takes values in the finite field $F_q := \{0, \ldots, q-1\}$ (multiplication and addition modulo $q$), where $q$ is a prime. As explained in [10] there is no loss of generality in assuming that the number of discrete values is a prime and that it is equal for all $u_i$, $i = 1, \ldots, M$. A general system in this class of his can now be written

If $\overline{u}(t) = (j_1, \ldots, j_M)$ then
\[
\dot{x}(t) = f^j(x(t), u(t)), \quad j \in F_q^M \\
y(t) = h(x(t), u(t))
\]

in the continuous time case, and

If $\overline{u}(t) = (j_1, \ldots, j_M)$ then
\[
x(t + 1) = f^j(x(t), u(t)), \quad j \in F_q^M \\
y(t) = h(x(t), u(t))
\]
in discrete time. The multi-index \( j \) ranges over all of \( F_q^M \) and for each \( j \)
\[
f^j : \mathbb{R}^{n+m} \to \mathbb{R}^n, \quad h : \mathbb{R}^{n+m} \to \mathbb{R}
\]
are some, possibly nonlinear, functions.

For the problems addressed in this paper there is only a small difference between continuous and discrete time systems, so we will mostly deal with differential equations.

Example 1.2 is particularly simple in that \( u_1 \) only affects \( x_1 \) and \( u_2 \) only \( x_2 \): the first component of \( f^{(j_1,j_2)} \) only depends on \( j_1 \) and the second one only on \( j_2 \).

Some problems we will address are:

- Can we eliminate the latent variables \( x_1, x_2 \)?
- Is there an algorithm for such elimination?
- Can we estimate the latent variables from measurements of the input and output variables (observability)?

For polynomial continuous and discrete time systems these questions are well understood and have been given complete and satisfactory answers, see e.g. [5, 6, 8, 7, 11, 13, 25].

Before continuing the treatment of these questions for the particular class of hs defined, let us give a short comparison with previous work on hs and the difference between the present and other ones.

Firstly, in some of the previous work a large part of the difficult is that the time-scale is non-classical [18, 19]. E.g. time may be a partially ordered set or we can have several simultaneous time scales. Secondly, the type of systems discussed here is quite simple. It is simpler than all those discussed in [1, 2, 14, 16, 20, 21]. In fact, in the terminology of [1] we are studying a special case of a type A hybrid system. Of course, simpler systems allow for more powerful analysis tools.

For systems with a well defined physical background, bond graphs provide a powerful modeling tool. Lately the bond graph formalism has been extended to a class of hybrid systems [22, 23].

The computational tool in this paper is a special kind of Gröbner bases. It should be noted that Gröbner bases have been used earlier in connection with hybrid systems [18, 19], but to do multivalued logic, not in order to determine external behaviors.

We start by looking at the state elimination problem. Of course one solution is to do the elimination for each special case \( \bar{u} \in F_q^M \). This is possible to do algorithmically e.g. if all \( f^j \) are polynomial [5, 8, 13]. However, this solution is unappealing in several ways: it could lead to a combinatorial explosion (we have to make \( q^M \) computations) and the structure of the system will probably not show very well. It would be nicer if we could do it directly on the equations defining the system, e.g. in example 1.2:

\[
\begin{align*}
\dot{x}_1 &= \begin{cases} 
-x_1, & u_1 = 0 \\
 0, & u_1 = 1 
\end{cases} \\
\dot{x}_2 &= \begin{cases} 
-x_2, & u_2 = 0 \\
 0, & u_2 = 1 
\end{cases} \\
y &= x_2
\end{align*}
\]  

The system (5) looks "discontinuous" in the sense that we used braces when writing down its equations. Of course, it is continuous in \( u \), the "discontinuity" consists in \( u \) taking values in a finite set. The system can be modeled as a polynomial nonlinear system.
2 Algebrization

In this section and the next we describe methods for modeling and computing with a class of hs. The mathematical tools used come from commutative algebra and algebraic geometry. Knowledge about this kind of mathematics is not very widespread among engineers, but unfortunately there is not place enough here to recall the basics of this branch. Instead we refer to the excellent introductory textbook by Cox et al. [3]. We will use concepts such as: ideals, prime ideals, Gröbner bases, the Zariski topology.

Example 2.1 Let us go back to the system in example 1.2 for a while. Since $u_1, u_2$ only take the values 0 and 1 it is not difficult to see that an equivalent set of equations is

$$
\begin{align*}
\dot{x}_1 &= x_2 - u_1 x_1 - u_1 x_1 \\
\dot{x}_2 &= x_1 - u_2 x_1 - u_2 x_2
\end{align*}
$$

If we expand these equations we get

$$
\begin{align*}
\dot{x}_1 &= u_2 - u_1 u_2 - u_1 x_1 \\
\dot{x}_2 &= x_1 - u_2 x_1 - u_2 x_2
\end{align*}
$$

i.e. an ordinary state space description of a polynomial system.

This simple trick works for ternary etc input variables as well, by Lagrange interpolation, q.v. e.g [4]. Let

$$
\lambda_{i,\nu}(u) = \prod_{\substack{\mu \in F_q \setminus \nu \\
\mu \neq \nu}} \frac{(u_i - \mu)}{(\nu - \mu)}
$$

Then

$$
\lambda_{i,\nu}(u) = \begin{cases} 
1, & \text{if } u_i = \nu \\
0, & \text{if } u_i \neq \nu
\end{cases}
$$

Using this, the polynomial system corresponding to (2) will be

$$
\sum_{j \in F_q^m} \left\{ \left( \prod_{i=1}^{m} \lambda_{i,j_i}(u) \right) \left( \dot{x} - f^j(x, u) \right) \right\} = 0
$$

Example 2.2 If we have that

$$
(u = 0 \Rightarrow \dot{x}_1 = -x_1), \quad (u = 1 \Rightarrow \dot{x}_1 = -2x_1), \quad (u = 2 \Rightarrow \dot{x}_1 = x_2)
$$

then we can write this as

$$
\frac{1}{2} \left( 2 - u \right) \left( 1 - u \right) \left( \dot{x}_1 + x_1 \right) + u \left( 2 - u \right) \left( \dot{x}_1 + 2x_1 \right) + \frac{1}{2} u \left( u - 1 \right) \left( \dot{x}_1 - x_2 \right) = 0
$$

Let us now formalize the algebrization\* procedure that translated equations (5) into equations (6) a little to understand how it works and what its limitations are. We are trying

\*By algebrization we simply mean “making the system polynomial”. 

to define an operator $L$ from logical propositions $P$ involving the dynamical variables to polynomials in these variables. Hereby we have the convention that

$$P \leftrightarrow [L(P) = 0]$$

Thus

$$[P_1 \lor P_2] \leftrightarrow [L(P_1)L(P_2) = 0] \quad (13)$$

But there is one large problem with the $L$-operator: What is $L(P)$?

This is important, because $[P_1 \rightarrow P_2] \leftrightarrow [\neg P_1 \lor P_2]$. It is not possible to extend $L$ to this case, unless there is a polynomial equation representing $\neg P_1$.

This is only possible if the variables involved in $P_1$ take a finite number of values, because in this case all algebraic sets are both Zariski-open and Zariski-closed!

**Example 2.3** If $u \in F_3$ we can take $L(\neg |u = 1|) := u \cdot (u - 2)$.

In our case we have a set of clauses of the type

$$P_1 \rightarrow P_2, \quad P_3 \rightarrow P_4, \ldots \quad (14)$$

where $L(\neg P_{2i-1})$ are well defined, and the $P_{2i-1}$ are mutually exclusive, i.e.

$$\text{Xor}(P_1, P_3, P_5, \ldots) = 1 \quad (15)$$

So instead of

$$L(\neg P_1)L(P_2) = 0, \quad L(\neg P_3)L(P_4) = 0, \ldots \quad (16)$$

we can write

$$L(\neg P_1)L(P_2) + L(\neg P_3)L(P_4) + \ldots = 0 \quad (17)$$

since only one of $L(\neg P_1), L(\neg P_3)$, will be nonzero at a time.

It’s highly probable that the above discussion can be made more rigorous and succinct using model theory [24].

The following theorem concludes the discussion about algebrization in showing that for $f^j$ linear in $x$ we always get algebrizations that are linear in $x$.

**Theorem 2.1** Consider systems of the type (2) in which all $f^j, j \in F_q^M$ are affine functions of $x$. For such systems the algebrization procedure captured by formula (10) results in equations of the type

$$\dot{x}_i = a_i(u, x) + b_i(u)$$

where $a_i$ are polynomials in $u$ and linear in $x$, and $b_i$ is a polynomial in $u$.

**Proof.** An alternative formulation is that the coefficient of $\dot{x}$ in (10) does not depend on $u$. Now, it can be proved in a straightforward manner that

$$\sum_{\nu} \lambda_{i,\nu}(u) = 1 \quad (18)$$

for every $i$. A conceptually more appealing proof is obtained by just noticing that $P = \sum \lambda_{i,\nu}(u)$ is a polynomial in $u_i$ that takes the value 1 for all $u_i \in F_q$. Since there is a one-to-one correspondence between polynomials of degree $< q$ and polynomial functions $F_q \rightarrow F_q$, the polynomial $P$ has to be identically equal to 1. Compare with the analog of the Hilbert Nullstellensatz in finite rings [9].
3 State Elimination

After the algebrization we could proceed as suggested in e.g. [7, 8], i.e. use Gröbner bases (gb) to eliminate the latent variables $x_1, x_2$. However, there are some more or less interesting additional aspects. Let us first very briefly describe the approach in [8], though.

The basic idea is to differentiate the output w.r.t. time and then replace every $\dot{x}_i$ that occurs with the corresponding rhs in the state equation. For input variables no such substitution is possible, of course. This is formally captured by Lie-derivatives.

For a discrete time system one proceeds in an analogous way, taking time-shifts of $y$ and eliminating all occurrences of $x_i(t + 1)$ using the state equations.

An additional problem for systems of the type (2) is: When we take Lie-derivatives of the output map $h$ (in this case $x_2$), what are the time-derivatives of the discrete variables? If we consider the inputs as parameters, i.e. they don’t vary with time, these derivatives are zero.

**Example 3.1** (Continuation of example 1.2) Taking Lie-derivatives we get

\[ y_0 = h_0 = x_2, \quad y_1 = h_1 = x_1 - u_2 x_1 - u_2 x_2 \tag{19} \]
\[ y_2 = h_2 = u_2 u_1 x_1 - u_1 x_1 + u_2 x_2 \tag{20} \]

where we used the notation

\[ y_i := \frac{d^i}{dt^i} y(t) \]

In the rest of the paper we will treat systems with only one output i.e. $p = 0$, so there is no harm in using subindices to denote time-derivatives of $y$.

Above we used the Fermat-relations $u_i^2 \equiv u_i$ to simplify $h_2$. They represent the fact that for every prime $q$

\[ \forall x \in F_q : \quad x^q = x \tag{21} \]

known as Fermat’s (little) theorem [15]. The Fermat relations are fundamental in algebraic modeling of discrete event systems; see e.g. [9, 10].

If we allow the inputs to vary, which is more natural in this application, we run into some formal problems, since this implies that the output will not be continuously differentiable, so one might object to our attempts of finding a differential equation for $y$. However, this situation arises already for ordinary linear systems if we allow the input to change step-wise, which is the normal assumption in sampled systems theory. In the case the input is not constant should look for weak solutions of the input-output equation, i.e. allow distributions etc. For further aspects of this problem we refer to chapter 1 of [17] and its references.

These problems entirely disappear if we consider discrete time systems, of course.

Note that in our example all the Lie-derivatives are linear in the $x_i$, since $h, f_1, f_2$ are. The next step is now to find a dependency relation between the polynomials representing $y_0, y_1, y_2$, i.e. we wish to find a nontrivial polynomial $P$ with coefficients in $k(u_1, u_2)$, such that

\[ P(y_0, y_1, y_2) = 0 \tag{22} \]

The existence of such a polynomial is guaranteed by the following theorem:
Theorem 3.1 Let $K$ be any field. Any $n$ rational functions $h_0, \ldots, h_n \in K(x_1, \ldots, x_n)$ are algebraically dependent over $K$.

Proof. This follows immediately from the fact that the transcendence degree of $K(x_1, \ldots, x_n)$ over $K$ is $\leq n$ with equality iff $x_1, \ldots, x_n$ are algebraically dependent over $K$. □

To actually determine the dependency relation one could use elimination theory, e.g., Gröbner bases, to eliminate the $x$-variables in the polynomial ideal

$$\mathcal{O}_n := \langle y_0 - h_0(x,u), \ldots, y_n - h_n(x,u) \rangle$$

in the ring $k[y_0, \ldots, y_n, x, u]$, i.e., $y_0, \ldots, y_n$ are variables and $h_0, \ldots, h_n$ are polynomials in $x, u$. In order to take into account that $u_1, \ldots, u_M \in F_q$ we add the Fermat-relations to $\mathcal{O}_n$:

$$\mathcal{O}_n' := \mathcal{O}_n + \langle u_1^q - u_1, \ldots, u_M^q - u_M \rangle$$

If the $h_i$ are linear in $x$, as they are in example 1.2, the polynomial $P$ will be linear in $y$, so it is a little overkill to use ideal theory and Gröbner bases in that case, we could just use linear algebra. But there is an interesting additional problem here, more interesting than the first one: Do we get the same thing if we assign values to $u_1, u_2$ first and then eliminate $x$ as if we do it in the other order?

Example 3.2 Let us continue example 1.2 to explain the complications that may arise. If we compute a gb for the ideal

$$\mathcal{O}_2 = \langle y_0 - x_2, y_1 - x_1 + (x_2 + x_1)u_2, y_2 - u_2x_2 - u_2u_1x_1 + u_1x_1 \rangle$$

w.r.t. the lexicographic term ordering $x_1 > x_2 > y_2 > y_1 > y_0$ we get

$$G_1 := \{y_1 - x_1 + (x_1 + y_0)u_2, x_2 - y_0, y_2 + u_1y_1 + (u_1u_2 - u_2)y_0 \}$$

after applying the Fermat-relations. Note that all computations should be made in characteristic zero, even though the $u_i$ take values in $F_q$.

Now, according to $G_1$ a candidate input-output-equation is

$$p_1 := y_2 + u_1y_1 - (u_2 - u_1u_2)y_0$$

But if we put $u_1 = u_2 = 1$ in the original equations we get

$$y_1 + y_0 = 0$$

whereas the same substitution in $p_1$ renders

$$y_2 + y_1 = 0$$

Luckily the first element of $G_1$ is $y_1 + y_0$ under this substitution. The point is that the situation could be even worse: $G_1$ might not be a gb any more, after substitution! □

It seems that ordinary Gröbner bases are not informative enough when it comes to solving the latent variable elimination problem for this class of hs. Instead we will have to turn our attention to a more powerful tool: comprehensive Gröbner bases.
4 Specializations and Comprehensive Gröbner Bases

To understand what a comprehensive Gröbner basis is, we have to make a deviation on specializations. Recall that a $k$-specialization is a ring-homomorphism

$$\varphi : k[U_1, \ldots, U_m, X_1, \ldots, X_n] \rightarrow k[X_1, \ldots, X_n]$$

defined (generated) by $\varphi(U_i) = \eta_i \in k$ for some $\eta_i$. An example: Let

$$\varphi : \mathbb{Q}[U_1, X_1, X_2] \rightarrow \mathbb{Q}[X_1, X_2]$$

be given by $\varphi(U_1) = 3$. Then

$$\varphi(X_1^2 - U_1^2 + 7U_1X_2) = X_1^2 + 21X_2 - 9$$

For a more general, and rigorous, definition see [26]. So in a sense “specialization” just means “assigning values to some of the variables”!

**Definition 4.1** [26] Consider an ideal

$$I \subset k[U_1, \ldots, U_m, X_1, \ldots, X_n] =: R$$

and a fixed term-ordering on $R$. A finite subset $G \subseteq I$ is a comprehensive Gröbner basis (cgb) for $I$ iff $\varphi(G)$ is a gb for $\varphi(I)$ for all $k$-specializations $\varphi : R \rightarrow k[X_1, \ldots, X_n]$. \hfill $\square$

Again, a more general definition appears in [26].

Note that $G$ is required to be finite, as opposed to usual gb. This means that the existence of cgb does not follow from the existence of gb, but in [26] it is proved that every ideal in $R$ has a cgb, and an algorithm for computing cgb is described. Implementations of the cgb algorithm have been made e.g. in the computer algebra systems Axiom and SAC-2. It is also showed that the concept is not trivial, i.e. there are gb that are not cgb. A sufficient condition for a gb to be a cgb is derived in [12].

What we need for the application above is cgb, so that the gb is preserved under every specialization of the discrete variables. Our main result is

If $G$ is a comprehensive Gröbner basis for the ideal $\mathfrak{O}_n$ then for every assignment

$$\varphi : \bar{u} \rightarrow F^M_q$$

of $u$, the set $\varphi(G)$ contains equations telling

1. What equations the $y$ satisfy.
2. How the latent variables $x$ are influenced by $y$

In other words the cgb $G$ contains all relevant information about the system’s behavior.

However there are some rather interesting questions remaining: How should we “decode” the information contained in a cgb? How should we translate polynomials into propositions, i.e. what is the inverse of the $L$-operator?
Let us return to example 1.2 for the last time.

**Example 4.1** Compute a gb w.r.t. the lexicographic term ordering

\[ x_1 > x_2 > u_2 > u_1 > y_2 > y_1 > y_0 \]

i.e. consider \( u \) as variables instead of elements in the coefficient field. We also include the Fermat relations in the ideal:

\[ \mathcal{O}'_2 := \mathcal{O}_2 + \langle u_1^2 - u_1, u_2^2 - u_2 \rangle \]  

A gb for \( \mathcal{O}'_2 \) is

\[ G_2 := \{ u_2 x_1 - x_1 + y_0 u_2 + y_1, (x_1 - y_1)(y_2 - y_0), (y_1 + y_0)(x_1 - y_1), x_2 - y_0, \\
    y_2 + u_1 y_1 + u_2 u_1 y_0 - u_2 y_0, u_2(y_2 - y_0), u_2(y_1 + y_0), u_1(y_2 + y_1), \\
    (y_1 + y_0)(u_1 y_1 + y_2), y_2(y_2 + y_1) \} \]

plus Fermat-relations. Note that due to the Fermat relation, the ideal is not prime. Now we see that \( x_2 \) is always observable, \( x_1 \) is observable iff \( u_1 \) is zero, etc.

It is not guaranteed that \( G_2 \) is a cgb for \( \mathcal{O}'_2 \) (the algorithm in [26] hasn’t been implemented yet by the author), but clearly it contains more information than \( G_1 \) in (26).

\[ \Box \]

### 5 Conclusions and Extensions

We have considered the latent variable elimination problem for a class of hs. The first step in our approach is to translate logical conditions to polynomial equations. The discrete variables are assumed to be input variables. When eliminating the latent variables we get some specialization problems, which naturally lead to comprehensive gb.

The main message is: If we model hs as polynomial systems we have access to *algorithmic* tools for e.g. observability, and comprehensive Gröbner bases enter quite naturally.

The next step in this preliminary study will be to use the techniques described on an application example. It is also tempting to try and generalize the tools presented here to a wider class of hybrid systems, most importantly such with logical conditions involving latent variables.

### Acknowledgement

This work was financially supported by the Swedish Council for Technical Research (TFR).

### References


