DIFFERENTIAL GEOMETRIC APPROACH AND APPLICATION OF COMPUTER ALGEBRA

Kurt Schlacher
Department for Automatic Control and Control Systems Technology, Johannes Kepler University Linz, Austria

Keywords: nonlinear control, differential geometry, computer algebra, algebraic methods, formal integrability, equivalence problems, accessibility, observability

Contents

Summary

Applications of computer algebra based methods in control are relatively new in contrast to numerical methods, although several systems are available since 40 years. In the field of nonlinear control these methods are indispensable, because they liberate the control engineer from laborious symbolic calculations like symbolic differentiation, simplification of equations or elimination of variables. Many problems in control can be reduced to a special problem, also called equivalence problem, whenever a given dynamic system can be transformed by a certain change of coordinates into a given canonical form. In contrast to linear and time invariant systems, where linear algebra suffices to handle this type of problem, one has to deal with over determined partial differential equations in the case of nonlinear systems. Computer algebra systems are efficient tools for the test, whether such systems of partial differential equations admit a solution. But the application of computer algebra methods requires an adequate representation of the dynamic system. The main idea is the separation of the description of a set of differential equations from the description of its solutions, because only the first part is accessible for computer algebra systems. Since jet manifolds are the suitable tool, we give a short introduction to this theory before its application to the equivalence problem is presented. The power of computer algebra methods is demonstrated by some

selected problems of nonlinear control, like the accessibility or observability problem or the input to state linearization problem. Apart from these standard problems, we discuss also applications of the proposed methods to implicit systems.

1. Introduction

The idea of doing algebraic calculations on computers is not new, from 1960 numerous computer algebra systems have appeared on the market which show that one can go beyond purely numerical computations. Interestingly enough, there was only a little number of applications in control until the middle of the eighties, when nonlinear control and differential geometry were brought together. The success of this approach was accompanied by the development of new and powerful computer algebra systems, which are now available on nearly any computer platform. Besides simple examples, the new methods in geometric control require the performance of straightforward but very tedious symbolic computations, and computer algebra systems seemed to be the right tool to overcome this problem. After a short period of enthusiasm, the disappointment prevailed since it seemed that computer algebra systems were not capable to solve problems more complex than text book examples. The main reason for this failure was the belief that symbolic algorithms should be developed in the same way as numeric ones and that is sufficient to transfer formulas and algorithms, of course very complex ones, simply from a text book into the algebra system.

Let us take a short look at the basic algorithms for accessibility or observability, etc. for nonlinear systems, which can be found in textbooks, from a computational point of view. These algorithms require symbolic differentiation, the determination of the rank of matrices, and the solution of special kinds of partial differential equations. Symbolic differentiation can be performed in a straightforward manner, but the determination of the rank of a matrix requires the test, whether a symbolic expression equals zero. Unfortunately, the latter problem is a hard one in symbolic computation. If this problem is not treated thoroughly, then an algorithm may fail to handle the problem correctly. The symbolic determination of the solution of sets of ordinary or even partial differential equations is an even more complex problem. It is a very interesting field of research, but also far beyond this chapter.

The goal of this chapter is to show that many problems in geometric control can often be reduced to a test, whether a system of overdetermined algebraic, ordinary differential or partial differential equations admits a solution. From a computational point of view this check requires two basic operations, the differentiation of symbolic expressions and the elimination of variables from a set of equations. The idea is to separate the check for the existence of a solution strictly from the determination of the solution itself. The check can often be performed efficiently by algebra systems, whereas the determination of the solutions requires more advanced methods. To get an impression of this approach, we consider the control system

$$\dot{x}^{\alpha} = x_1^{\alpha} = f^{\alpha}\left(x,u\right), \qquad \alpha = 1,\ldots,n \tag{1}$$

of n ordinary differential equations with the state $x \in \mathbb{R}^n$ and the single input $u \in \mathbb{R}.$ the symbols $\dot{x}^\alpha$ or x_1^α denote the derivative of the dependent variable x^α with respect to the independent variable $t \in \mathbb{R}.$ Given the system

$$\bar{x}_1^\alpha = \bar{f}^\alpha\left(\bar{x},\bar{u}\right) \quad , \quad \alpha = 1,\dots,n, \tag{2}$$

we pose the problem, whether there exists a coordinate transformation

$$\bar{x}^\alpha = \varphi^\alpha\left(x\right) \quad , \quad \bar{u} = \psi\left(x,u\right), \tag{3}$$

such that any solution of (1) is mapped to a solution of (2) and vice versa. From (1),(2),(3) it follows that the transformation (3) meets the following set of partial differential equations of first order

$$\sum_{\beta=1}^{n} \partial_\beta \varphi^\alpha\left(x\right) f^\beta\left(x,u\right) = \bar{f}^\alpha\left(\varphi\left(x\right),\psi\left(x,u\right)\right) \quad , \quad \partial_\beta = \frac{\partial}{\partial x^\beta} \tag{4}$$

for $\alpha = 1,\dots,n.$ A good strategy is to tackle this problem in two steps. The goal of the first step is to derive necessary and locally sufficient conditions for the solvability of (4), whereas the determination of the functions φ^α, ψ is postponed to the second step. As mentioned above, this chapter is focused on the first step, where computer algebra systems turned out to be most useful currently. Roughly speaking, we present algorithms which allow us to check the local solvability of the problem, but we are not able to construct the symbolic solution in general. However, also computer algebra methods for ordinary and partial differential equations are making fast progress and they have already reached a level higher than that of even well trained engineers.

This chapter is organized as follows. First we start with some remarks on symbolic computation and summarize the most important mathematical facts for a computer algebra adequate description of control problems, presented later on. The main idea of the approach presented here is the separation of the description of a set of differential equations from the description of its solutions, because computer algebra systems can handle only the equations efficiently in contrast to the problem of the determination of their solutions. If additionally these equations are of a polynomial type, then one can use powerful algebraic tools like Groebner bases or algebraic elimination theory. The following section is devoted to the equivalence problem of two dynamic systems, where we discuss the first system can be converted to solutions of the second one? The usefulness of this problem is evident, since one can reduce seemingly new problems in many cases to solved ones. The price, one has to pay, is that these transformations are solutions of systems of over determined partial differential equations. But this problem has been discussed in detail in the previous section. We demonstrate the presented approach with the help of the accessibility, the observability, and the input to state linearization problem. Apart from these standard problems, we discuss also applications to implicit control systems, where computer algebra methods are even more important than for explicit systems. Finally, we bring this chapter to an end with some concluding remarks.

2. Remarks on Symbolic Computation

Computer algebra systems have historically evolved in several stages. MACSYMA, SCRATCHPAD, REDUCE, and MUMATH, available since the late 1960s, belong to the first generation. The second generation with MAPLE and MATHEMATICA made computer algebra applications popular in applied mathematics and engineering. Currently, the third generation with AXIOM, MAGMA or MUPAD is on the market. Of course, this enumeration cannot be complete, since many specialized products are available in addition.

When engineers start to use computer algebra systems, they have to face the problem that these systems are not simply a replacement for doing symbolic calculations by hand, since they require new techniques for being used efficiently. E.g., let us look at simple operations like addition, multiplication, etc. Performing these operations with floating point numbers, one may assume that their execution time is independent of the data. That is not at all true for symbolic computation, where the data are symbolic expressions and the execution time heavily depends on the size of the data. Additionally, there is no canonical representation of the data and the user must take care of it. Factors of 10 to 1000 for the execution time can easily be gained or lost, since the execution time does not depend on the number of operations only but also on the size of the data and on their representation. For numerical computations one prefers algorithms with a small number of operations. Since in symbolic computation also the size of data should be kept small, often algorithms are used that are avoided in numerical problems. A simple example is the check, whether k vectors

$$v_j = \sum_{i=1}^{n} v_j^i e_i\,,\quad v_j^i \in \mathbb{R}\,,\quad k \le n\,,\quad \operatorname{span}\left\{e_i\right\} = \mathbb{R}^n$$

are linearly independent. The test, whether rank $\left[v_j^i\right] < k$ is met, gives the right answer. It can be performed by Gaussian elimination in a straightforward manner. An alternative approach is based on the Grassmann algebra $\wedge(\mathbb{R}^n)$ of $\mathbb{R}^n$ with the Grassmann or exterior product $\wedge$, which is implemented in any advanced computer algebra system. The test takes the simple form $\bigwedge_{j=1}^{k} v_j = 0$, and its performance with symbolic data is in most cases higher than the Gaussian elimination. (Roughly speaking, the Grassmann product $\wedge$ is the generalization of the cross product $\times$ from $\mathbb{R}^3$ to $\mathbb{R}^n, n \ge 1.$)

A difficult problem in symbolic computation is the test whether two expressions are equal in some mathematical sense. E.g., let e denote the symbolic expression

$$e = 2\sin^2(x) = \cos(2x)$$

and let us perform the test $e \overset{?}{=} 1$ on a computer algebra system. Without use of trigonometric simplifiers the answer will be $e \ne 1,$ since the system compares the pattern of the expressions, which are obviously different. Of course, we get the correct answer applying the suitable trigonometric simplifier. Therefore, the user has to select the correct simplifiers that ought to be applied. Fortunately, problems of this type can be reduced to an ideal membership problem, well known in algebra and algebraic geometry. Let $f^\alpha \in \mathbb{K}\left[x^1, \ldots, x^p\right],\ \alpha = 1, \ldots, q$ denote q polynomials in the unknown

$x^i,\ i=1,\dots,p$ over a field $\mathbb{K}$. We pose the problem, whether a given polynomial g is contained in the ideal generated by f^α, or equivalently, whether one can find polynomials λ_α such that

$$g=\sum_{\alpha=1}^{q}\lambda_\alpha f^\alpha$$

is met. This problem can be solved reliably with Groebner bases. Let us restate the problem from above. We set

$$z^1=\sin(x),\quad z^2=\cos(x),\quad z^3=\cos(2x)$$

$$f^1=\left(z^1\right)^2+\left(z^2\right)^2-1,\quad f^2=z^3-2\left(z^2\right)^2+1$$

and see that $\lambda_1=2,\lambda_2=1$ solve the equation

$$e-1=2\left(z^1\right)^2+z^3-1=\lambda_1 f^1+\lambda_2 f^2.$$

Obviously, $f^1=f^2=0$ implies $e=1$. We will not develop this approach further, although many developers of computer algebra packages have to use these algorithms. Since Groebner bases are implemented in any advanced computer algebra system, as well as many powerful algorithms based on Groebner bases, the user can fall back to many powerful simplifiers such that the simplification of “general” expressions with respect to many predefined rules can be performed in a straightforward manner. Nevertheless, one should be aware of the fact that the application of simplifiers is time consuming, and the use of many simplifiers can slow down the programs significantly.

Many problems in symbolic computing, like the simplification of general expressions, are NP-complete. According to our knowledge, the user assistance is the best help to make these difficult problems manageable. E.g., programs should be written in a way that rules for simplifiers can be added or removed by the user, because the decision tree must be cut as early as possible. Nevertheless, the new developments show that the capability of algebra systems in this field has reached a level which is often much higher than the level of even well trained engineers. At present, there are only a few packages available for problems in nonlinear control, but the situation improves fast. The reader will find packages in the WWW, which can deal with many problems in geometric control for explicit systems.

The main idea for the application of computer algebra methods to control problems is based on the possibility to represent dynamic systems as algebraic objects. We present an approach, where dynamic systems describe submanifolds of special manifolds, called jet manifolds, such that standard methods of algebraic computing become available. Therefore, we summarize some mathematical facts concerning regular manifolds, jet manifolds and the application of this theory to special sets of partial differential equations in the following section.

3. Some Mathematical Facts

Computer algebra systems process symbolic, especially algebraic expressions, and roughly speaking they have no knowledge concerning dynamic systems or differential equations. Therefore, one has to convert a control problem to an algebraic problem in such a way that it can be treated by symbolic methods. Jet manifolds are the suitable mathematical tool to express differential equations in an algebraic manner. Before we start with this topic, we summarize some useful notation. To avoid mathematical subtleties, we assume from now on that all functions and maps under consideration are smooth. The reader may consult textbook for further information about the presented topics.

We use the concept of bundles throughout this chapter. A bundle is a triple $(\mathcal{E},\pi,\mathcal{B})$ with the total manifold $\mathcal{E}$, the base manifold $\mathcal{B}$ and the surjective submersion $\pi:\mathcal{E}\to\mathcal{B}$. For each point $x\in\mathcal{B}$, the subset $\pi^{-1}(x)=\mathcal{E}_x$ is called the fiber over *x*. All fibers are diffeomorphic to the so called typical fiber. In the finite dimensional case with $\dim\mathcal{E}=p+q,\dim\mathcal{B}=p$ we can introduce adapted coordinates $\left(x^i,u^\alpha\right)$ at least locally, with the independent coordinates $x^i, i=1,\dots,p$ and the dependent ones $u^\alpha,\alpha=1,\dots,q$. Since one always has to choose coordinates to represent a problem in a computer algebra system, we will exclusively take adapted coordinates from now on. Often we will write $\mathcal{E}$ instead of $(\mathcal{E},\pi,\mathcal{B})$, whenever the projection π and the base manifold $\mathcal{B}$ follow from the context. A section σ of $\mathcal{E}$ is a map $\sigma:\mathcal{B}\to\mathcal{E}$ such that $\pi\circ\sigma=\mathrm{id}_{\mathcal{B}}$ is met, where $\mathrm{id}_{\mathcal{B}}$ denotes the identity map on $\mathcal{B}$. We do not require that a section σ exists globally and write for the set of all sections $\Gamma(\mathcal{E})$. From now on, we use Latin indices for the independent and Greek indices for the dependent variables.

Let $\mathcal{M}$ be a smooth *m*-dimensional manifold, then the tangent bundle $\mathcal{T}(\mathcal{M})$ and the cotangent bundle $\mathcal{T}^*(\mathcal{M})$ are well known examples of bundles in engineering. Using local coordinates, we write

$$v^i(x)\partial_i\in\Gamma\left(\mathcal{T}(\mathcal{M})\right),\qquad \omega_i(x)\mathrm{d}x^i\in\Gamma\left(\mathcal{T}^*(\mathcal{M})\right),\qquad i=1,\dots,m$$

for sections of $\mathcal{T}(\mathcal{M}),\mathcal{T}^*(\mathcal{M})$, where we applied already the Einstein convention for sums to keep the formulas short and readable. Furthermore, we have $v^i,\omega_i\in C^\infty(\mathcal{M})$ with the set $C^\infty(\mathcal{M})$ of smooth functions $\mathcal{M}\to\mathbb{R}$. From these bundles one derives further bundles, like the exterior *k*-form bundle $\wedge_k^*(\mathcal{M})$ or other tensor bundles. We denote the exterior algebra over $\mathcal{M}$ by $\wedge^*(\mathcal{M})$, $\mathrm{d}:\wedge_k^*(\mathcal{M})\to\wedge_{k+1}^*(\mathcal{M})$ is the exterior derivative and $\mathrm{i}:\mathcal{T}(\mathcal{M})\times\wedge_{k+1}^*(\mathcal{M})\to\wedge_k^*(\mathcal{M})$ is the interior product written as $\mathrm{i}_X(\omega)$ with $X\in\mathcal{T}(\mathcal{M})$ and $\omega\in\wedge_{\kappa+1}^*(\mathcal{M})$. The symbol $\wedge$ denotes the exterior product of Grassmann product of the exterior algebra $\wedge^*(\mathcal{M})$. The Lie derivative of $\omega \in \wedge^*(\mathcal{M})$ along the field $f \in \Gamma(\mathcal{T}(\mathcal{M}))$ is written as $f(\omega)$, whereas for the Lie derivative of a vector field $g \in \Gamma(\mathcal{T}(\mathcal{M}))$ along f the bracket notation $[f,g]$ is used. The *k*-fold repeated Lie bracket is denoted by $\mathrm{ad}_f^k(g)$ with $\mathrm{ad}_f(g) = [f.g]$ and $\mathrm{ad}_f^{k+1}(g) = \left[f, \mathrm{ad}_f^k(g)\right]$. One can define also the exterior product $\wedge$ of tangent vectors, in this case we denote the corresponding *k*-vector bundle by $\wedge_k(\mathcal{M})$ and the algebra by $\wedge(\mathcal{M})$.

In order to apply computer algebra methods to control problems we represent dynamic systems as submanifolds of special manifolds, called jet manifolds. Particularly, if these submanifolds are described by polynomial equations, one can use algebraic tools like Groebner bases for their investigations. Therefore, we give now a short introduction to jet manifolds, followed by a discussion of submanifolds. We discuss the problem of formal integrability and show its connection to the Theorem of Frobenius, which is well known in non linear control.

3.1 Jet Manifolds

Let f be a smooth section of the bundle $(\mathcal{E},\pi,\mathcal{B})$. The k^{th} order partial derivatives of f will be denoted by

$$\frac{\partial^k}{\partial_1^{j_1}\cdots\partial_p^{j_p}} f^\alpha = \partial_j f^\alpha = f_j^\alpha, \quad \partial_i = \frac{\partial}{\partial x^i}$$

with $J = j_1,\dots,j_p$, and $k = \#J = \sum_{i=1}^p j_i$. J is nothing else than an ordered multi-index. The special index $J = j_1,\dots,j_p$, $j_i = \delta_{ik}$ will be denoted by 1_k and $J+1_k$ is a shortcut for $j_i + \delta_{ik}$ with the Kronecker symbol δ_{ik}. Let (x^i, u^α) be adapted coordinates and f be a smooth section of $\mathcal{E}$, then we can extend f to a map $j(f) = j^1(f) : x \to \left(x^i, f^\alpha(x), \partial_i f^\alpha(x)\right)$, the first jet of f. One can show that the set of all first jets of sections $\Gamma(\mathcal{E})$ is contained in a differentiable manifold, denoted by $J^1(\mathcal{E})$, whose structure is derived from this set. An adapted coordinate system of $\mathcal{E}$ induces an adapted system on $J^1(\mathcal{E})$, which is denoted by $\left(x^i, u^\alpha, u_{1_i}^\alpha\right)$ with the pq new coordinates $u_{1_i}^\alpha$. The manifold $J^1(\mathcal{E})$ has two natural projections, $\pi^1 : J^1(\mathcal{E}) \to \mathcal{B}$ and $\pi_0^1 : J^1(\mathcal{E}) \to \mathcal{E}$ with $\pi\left(j^1(f(x))\right) = x$ and $\pi_0^1\left(j^1(f(x))\right) = f(x)$, which correspond to the bundles $\left(J^1(\mathcal{E}), \pi, \mathcal{B}\right)$ and $\left(J^1(\mathcal{E}), \pi_0^1, \mathcal{E}\right)$. It is worth mentioning that a section f of these bundles is not necessarily the first jet $j^1(\sigma)$ of a section σ of since those sections must meet the relations $\partial_i f^\alpha - f^\alpha_{1_i} = 0$. Analogously to the first jet of a section f, we define the n^{th}-jet $j^n(f)$ of f by $j^n(f) = \left(x^i, f^\alpha(x), \partial_J f^\alpha(x)\right), \#J \;\; = 1,\ldots n$. The n^{th}-jet manifold $J^n(\mathcal{E})$ of $\mathcal{E}$ may be considered as a container for n^{th}-jets of sections of $\mathcal{E}$. Furthermore, an adapted coordinate system of $\mathcal{E}$ induces an adapted system on $J^n(\mathcal{E})$ with $\left(x^i, u^{(n)}\right)$ and $u^{(n)} = u^\alpha_J, a = 1,\ldots,q, \# J = 0,\ldots,n$. The natural projections and the corresponding bundles are given by

$$\begin{aligned} \pi^n &: J^n(\mathcal{E}) \to \mathcal{B}, && \left(J^n(\mathcal{E}), \pi^n, \mathcal{B}\right) \\ \pi^n_m &: J^n(\mathcal{E}) \to J^m(\mathcal{E}), && \left(J^n(\mathcal{E}), \pi^n_m, J^m(\mathcal{E})\right) \end{aligned} \tag{5}$$

for $m = 1,\ldots,n-1$ with $\pi\left(j^1(f(x))\right) = x$ and $\pi^n_m\left(j^n(f(x))\right) = j^m(f(x))$. To simplify certain formulas later on, we set $J^0(\mathcal{E}) = \mathcal{E}, J^1(\mathcal{E}) = J(\mathcal{E})$.

-
-
-

Bibliography

W.M. Boothby,1986, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, Inc.,Orlando, USA, [A mathematical introduction to the basics of differential geometry]

D. Cox and J. Little and D. O'Shea, 1998, *Using Algebraic Geometry*, Springer, [An introduction to algebraic geometry, algebraic computing and Groebner bases]

A. Isidori, 1995, *Nonlinear Control Systems*, Springer,London, UK,[The first standard textbook on geometric control and its applications]

A. Kugi and K. Schlacher and R. Novak, 1999,Software *Package*: *Non-linear Affine-Input Control Systems*,Maple-Applications Center, Engineering, Control, http://www.maplesoft.com,[A computer algebra package for the geometric control of nonlinear systems]

H. Nijmeijer and A. van der Schaft, 1990, *Nonlinear Dynamical Control Systems*,Springer,New York, [A standard textbook on geometric control and its applications including mechanical nonlinear control systems

J.F. Pommaret,1978, *Systems of Partial Differential Equations and Lie Pseudogroups*,Gordon and Breach Science Publishers, New York, USA,[A textbook on geometric and algebraic methods for partial differential equations]

S. Sastry, 1999, *Nonlinear Systems Analysis, Stability and Control,* Springer,New York, USA, [A standard textbook which gives an overview on nonlinear systems, Lyapunov theory, differential geometry and their applications]

D.J. Saunders, 1989, *The Geometry of Jet Bundles*, Cambridge University Press, Cambridge, UK, [A comprehensive introduction to jet manifolds and jet bundles]

M. Spivak,1979,*Differential Geometry*, Vol. 1 to 5,Publish or Perish, Inc., Houston, Texas,[A standard reference consisting of five books which gives a comprehensive overview on differential geometry]

J. zur Gathen and J. Gerhard, 1999, *Modern Computer Algebra*, Cambridge University Press,Cambridge, UK, [An introduction to computer algebra and symbolic computing]

Biographical Sketch

Prof. Kurt Schlacher was born in Graz, Austria, on the 16*th* of August in the year 1955. 1973 he started to study electrical engineering at the Technical University of Graz, and finished 1979 with the diploma degree cum laude. In the year 1980 he did his obligatory national service. In the year 1981 he joined the department of automatic control at the Technical University of Graz, where he received his Ph.D. cum laude in the year 1984 and his habilitation for automatic control in the year 1990. In 1992 he moved to Linz at the Johannes Kepler University, Austria, where he got the position of a full professor for Automatic Control that he holds presently. Apart from several academic positions, he serves as an Associate Editor of the IEEE Transactions on Control Systems Technology. He is also member of the scientific committees of the following journals: IFAC International Journal of Automation Austria, Automatisierungstechnik (Oldenbourg-Verlag, Germany), as well as member of the IFAC Technical Committees on Control Design and Mechatronics. Since 2002 he is member of the IFAC council and of EUCA council. Furthermore, he is head of the Christian Doppler Laboratory for Automatic Control of Mechatronic Systems in Steel Industries. His main interests are modeling and control of nonlinear systems with respect to industrial applications applying differential geometric and computer algebra based methods. He is author of more than 80 publications published in national and international proceedings and journals, as well as co-author of the book Digitale Regelkreise (Oldenbourg-Verlag) together with Prof. Hofer and Prof. Gausch.