Computer Science Journal of Moldova, vol.**16**, no.1(46), 2008

# Gröbner Bases for Nonlinear DAE Systems of Analog Circuits

Silke J. Spang

**Abstract**

Systems of differential equations play an important role in modelling and analysis of many complex systems e. g. in electronics and mechanics. The following article is concerned with a symbolic analysis approach for reduction of the differential index of nonlinear differential algebraic equation (DAE) systems, which occur in the modelling and simulation of analog circuits.

## 1 Introduction

Systems of differential equations play an important role in modelling and analysis of many complex systems e. g. in electronics and mechanics. For example, the simple oscillator circuit of figure 1, which is part of nearly all analog electronic devices, yields the following DAE

$$C1\left(V_1'(t) - V_2'(t)\right) - I_{L1}(t) = 0 \tag{1.1}$$

$$\frac{V_2(t)}{R1} + C1(V_2'(t) - V_1'(t)) = 0 \tag{1.2}$$

$$V_1(t) + L1 \cdot I_{L1}'(t) = 0 \tag{1.3}$$

where $I_{L1}$ denotes the current through the inductor $L1$ and $V_i$ the voltage between the node $i$ and the ground.

Unlike ordinary differential equation systems (short ODE), proper DAE systems are subject to hidden constraints. These constraints are not explicitely stated in the system of equations, but they constrain the solution within a certain manifold. For instance, in the above DAE

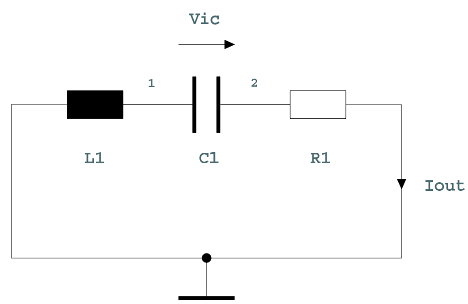


Figure 1. Analog oscillator circuit

there is no possibility to compute an explicit formula for $V_2'$ which does not depend on $V_1'$ and vice versa using algebraic deformations only. Deriving the whole system, we obtain:

$$C1\left(V_1'(t) - V_2'(t)\right) - I_{L1}(t) = 0 \tag{1}$$
$$\frac{V_2(t)}{R1} + C1(V_2'(t) - V_1'(t)) = 0 \tag{2}$$
$$V_1(t) + L1 \cdot I_{L1}'(t) = 0 \tag{3}$$
$$C1\left(V_1''(t) - V_2''(t)\right) - I_{L1}'(t) = 0 \tag{4}$$
$$\frac{V_2'(t)}{R1} + C1(V_2''(t) - V_1'(t)) = 0 \tag{5}$$
$$V_1'(t) + L1 \cdot I_{L1}''(t) = 0 \tag{6}$$

Adding (4) and (5) we get

$$\frac{V_2'(t)}{R1} - I_{L1}'(t) = 0, \tag{7}$$

multiplying (7) with $L1$ and adding it to (3) we end up with

$$\frac{V_2'(t)}{R1} \cdot L1 - V_1(t) = 0. \tag{8}$$

In the same way we get another equation for $V_1'(t)$ and get the following equation:

$$V_1'(t) = -\frac{R1}{L1} \cdot V_1(t) + \frac{1}{C1} \cdot I_{L1}(t) \tag{9}$$

So the system can be reformulated as an ODE of the following form:

$$V_1'(t) = -\frac{R1}{L1} \cdot V_1(t) + \frac{1}{C1} \cdot I_{L1}(t) \tag{2.1}$$

$$V_2'(t) = -\frac{R1}{L1} \cdot V_1(t) \tag{2.2}$$

$$I_{L1}'(t) = \frac{V_1(t)}{L1} \tag{2.3}$$

In general hidden constraints for such systems can be handled using methods from commutative algebra.

The treatment of linear DAE systems using algebraic methods is straight-forward, but this is not the case for nonlinear terms, e. g. DAEs containing exponential functions. Here some further development of computational methods is necessary to match the needs of this equations which arise from nonlinear circuits.

We describe a method for the detection of such hidden constraints and reformulate the DAE in an ODE like manner. In section 2 some background theory of differential systems and their differential index is explained and an algebraic framework by meanings of rings, ideals and Gröbner bases for the properties of local solvability and being formally integrable are given. The ring of all differential equations up to order $q$ w. r. t. the independent variable $t$ will be reinterpreted as the polynomial ring $A^{(q)}$. In section 3 the computational development during the work with SINGULAR and *Mathematica* on this subfield is described. We will see how such problems can be tackled using polynomial systems in SINGULAR. Systems containing the latter give rise to electrical circuits describing the behavior of transistors and diodes.

Section 4 expands our view to some new classes of functions. We will get some feeling how to tackle exponential functions, sines and cosines and in particular square-roots in an algebraic and polynomial frame. Systems containing exponential functions may give rise to electrical circuits describing the behavior of transistors and diodes. We will embed all these classes in a new ring $D$ for which we define derivative map $\varphi_D$. To this end will see in section 5 that our gained theory applied to some DAE systems good solutions in Analog Insydes. After our preprocessing the last example we detected some equations which gave sufficient constraints to compute the solution with the nonlinear DAE-Solver of *Mathematica*. Concluding with section six we will give some outlook for further development.

I would like to thank the Analog Insydes Team especially Dr. Alexander Dreyer for many fruitful discussions and their support with problems in Analog Insydes. I also want to thank the Fraunhofer Institute for Industrial Mathematics especially my department System Analysis, Prognosis and Control with the department chief Dr. Patrick Lang for giving my the opportunity to work this field. Last I want to thank my advisor Prof. Dr. Gerhard Pfister for being a good friend and advisor which always has an ear for me and my problems.

## 2 Basics and mathematical background

In this section we will present some algebraic and analytic basics which shall help to understand the next sections.

Suppose a DAE $(F)$ of order $q$ is given. We introduce the **differential index** (cf. [4]) of $(F)$ to be $r$ if a minimum of $r+1$ geometric differentiations of $(F)$ is required until no new constraint is found. Note that this index definition is one out of a group of indices measuring the difficulty of solving DAE systems (cf. [6]).

As already mentioned above, proper DAE systems yield additional constraints to the solution, which are not stated explicitely in terms of equations.

**Example 1**
*Consider the following system (cf. [6]) with functions $x_i$ in the independent variable $t$*

$$x_1' + x_1 = 0 \tag{3.1}$$
$$x_2 x_2' - x_3 = 0 \tag{3.2}$$
$$x_1^2 + x_2^2 - 1 = 0. \tag{3.3}$$

*This system admits a hidden constraint* $x_1^2 - x_3 = 0$ *which appears after a differentiation of equation* (3.3) *and the elimination* $x_1'$ *and* $x_2 \cdot x_2'$ *by using the equations* (3.1) *and* (3.2). *The above turns out to have differential index two as after two steps there occur no more hidden constraints. In this case the system can be transformed to an ODE.*

Systems of high index are algebraically underdetermined as they have a gap of constraints which only appear after differentiation. These hidden constraints may slow down numerical computations, or make them even impossible. Systems of lower indices have less of these hidden equations and it turns out to be desirable to transform a higher indexed system into one of a lower index. Among the approaches to decrease the differential index is the theory of **locally solvable** and **involutive** systems. (cf. [5] ) Here we prolongate and project the given DAE until no new constraint can be found.

We define the **prolongation** and the **projection** of $q$-th order systems (cf. [3] chapter 2, [4] chapter 2-3), where the prolongation coincides with differentiation and the projection with the elimination of the highest order part.

**Definition 2**
*Let* $f_1, .., f_n \in C^m(T, t)$ *be* $m$*-differentiable functions in the time* $t \in T \subset K$ *for an interval* $T$ *in a field* $K$. *If* $f_i^{(j)} = \frac{d^j f_i}{d^j t}$ *denotes the* $j$*-th derivation of* $f_i$*, then we denote the space of all* ***differential algebraic equations up to order*** $q$ *of* $f = (f_1, \ldots, f_n)$ *over* $K$ *by* $A^{(q)} = K[f^{(q)}, f^{(q-1)}, \ldots, f', f, t]$.

Now we are able to give a formal definition of projection and prolongation in terms of ring maps and elimination.

**Definition 3**
*Let* $D_t : A^{(q)} \to A^{(q+1)}$ *be a formal differentiation, i.e.:*

- $D_t(p \cdot q) = D_t(p) \cdot q + p \cdot D_t(q)$ *(chain rule)*
- $D_t(p + q) = D_t(p) + D_t(q)$
- $D_t(f_i^{(j)}) = f_i^{(j+1)}$ *for all* $f_i^{(j)} \in A^{(q)}$.

- *$D_t(a) = 0$ for all $a \in K$.*

*The field $Const(A) = \{a \in A^{(0)} : D_t(a) = 0\}$ is called the* ***field of constants****. Note that $K$ is a subfield of $Const(A)$, but they need not be equal.*

Now a DAE system $(F)$ can be transformed into an ideal $I$ of $A^{(q)}$. Recall that an ideal is a subset which is invariant under addition and scalar multiplication and is denoted by $I \trianglelefteq A$. Note that the solutions of $(F)$ do not coincide with the solutions of $I$. Of course, every solution of $(F)$ corresponds to a solution in $I$ but not vice versa. This is because algebraically the derivative of an $f_i$ is another variable and we have no a priori knowledge about their analytical relationship. The map $D_t$ has its natural extension for ideals $I = \langle g_1, \dots g_r\rangle \trianglelefteq A^q$ given by

$$D_t(I) = \langle D_t(g_1), \dots, D_t(g_r)\rangle.$$

**Definition 4**
*Let $I$ be an ideal in $A^{(q)}$:*

- *The* ***algebraic prolongation*** *of $I$ is defined to be*

$$\mathcal{P}(I) = \langle I, D_t(I)\rangle \trianglelefteq A^{q+1}.$$

- *The* ***algebraic projection*** *of $I$ is given by*

$$\mathcal{E}(I) = I \cap A^{q-1}.$$

The next definition gives some properties of systems which have a low index.

**Definition 5**
*Let $I \trianglelefteq A^{(q)}$ be an ideal then*

1. *$I$ is called* ***locally solvable*** *if*

$$\mathcal{E} \circ \mathcal{P}(I) = I.$$

*2. I is called **formally integrable** if for all $k \geq 0$*

$$\mathcal{E} \circ \mathcal{P}(\mathcal{P}^k(I)) = \mathcal{P}^k(I).$$

We try to use Gröbner basis methods to obtain such forms. First of all we define orderings and Gröbner bases (cf. [2]).

**Definition 6 (Ordering)**
*Let $A = K[x_1, \ldots, x_n]$ be an affine $K$-algebra. A total ordering $>$ on the set of monomials of $A$.*

$$Mon(A) = \{x_1^{a_1} \cdot x_2^{a_2} \cdots x_n^{a_n} : a_i \in \mathbb{N}\}$$

*is an antisymmetric binary relation*

$$> (x^\alpha, x^\beta) = \begin{cases} 1 & \alpha >_{\mathbb{N}^n} \beta \\ 0 & \alpha = \beta \\ -1 & \beta >_{\mathbb{N}^n} \alpha \end{cases}$$

*for an ordering $>_{\mathbb{N}^n}$ of $\mathbb{N}^n$. Additionally*

$$> (x^\alpha, x^\beta) = > (x^\gamma \cdot x^\alpha, x^\gamma \cdot x^\beta)$$

*holds. For simplicity we write:*

- $x^\alpha > x^\beta$ *if* $> (x^\alpha, x^\beta) = 1$
- $x^\alpha = x^\beta$ *if* $> (x^\alpha, x^\beta) = 0$
- $x^\alpha < x^\beta$ *if* $> (x^\alpha, x^\beta) = -1$.

*An ordering is a **well- or global ordering** if $x^\alpha > 1$ for all $\alpha \in \mathbb{N}^n \setminus \{(0, \ldots, 0)\}$, a **local ordering** if every $x^\alpha < 1$ and **mixed ordering** otherwise.*

**Definition 7 (leading monomial and leading ideal)**
*Let $A$ be an affine $K$-algebra and $>$ an ordering on $Mon(A)$, then:*

1. *For every polynomial*

$$f = f_1 x^{\alpha_1} + f_2 x^{\alpha_2} + \ldots + f_m x^{\alpha_m} \in A \setminus \{0\}$$

*with* $f_1 \neq 0$ *and* $x^{\alpha_1} > x^{\alpha_2} > \ldots > x^{\alpha_m}$ *let* $\mathrm{LM}(f) = x^{\alpha_1}$ *denote the* ***leading monomial****, which is the biggest monomial w. r. t.* $>$ *in* $f$*.*

2. *For any ideal* $I \trianglelefteq A$ *let* $\mathrm{L}(I) = \langle \mathrm{LM}(f) : f \in I \rangle$ *denote the* ***leading ideal*** *of* $I$*.*

Now we are able to define a **Gröbner basis** as a so called “fine form“ of an ideal $I$, see [2, Def. 1.6.1].

**Definition 8 (Gröbner basis)**
*Let* $I = \langle f_1, \ldots, f_r \rangle \trianglelefteq A$ *be any ideal. A* ***standard basis*** *is a representation* $\langle g_1, \ldots, g_m \rangle$ *of* $I$ *such that the equality* $\mathrm{L}(I) = \langle \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_m) \rangle$ *holds. If the underlying ordering* $>$ *is a global one then we call a standard basis just* ***Gröbner basis****.*

To represent ideals on computers, we can use Gröbner bases. This form is suitable for computations as it provides a reduction to the monomial case. Note that computing with monomials is only a combinatorial problem. The so-called **normal form** w. r. t. to a set $\{g_1, .., g_m\}$ is defined as follows:

**Definition 9 (Normal form, standard representation)**
*Let* $\mathcal{G}$ *denote the set of all finite subsets* $G \subset A$*. A map*

$$\mathrm{NF} : A \times \mathcal{G} \to A, \ (f, G) \mapsto \mathrm{NF}(f|G)$$

*is called* ***normal form*** *on* $A$ *if* $\mathrm{NF}(0|G) = 0$ *for all* $G \in \mathcal{G}$ *and for all* $f \in R$ *and* $G \in \mathcal{G}$*:*

1. $\mathrm{NF}(f|G) \neq 0 \implies \mathrm{LM}(\mathrm{NF}(f|G)) \notin \mathrm{L}(G)$*.*

2. *If* $G = \{g_1, \ldots, g_m\}$*, then* $r := f - \mathrm{NF}(f|G)$ *has a* ***standard representation*** *w. r. t.* $G$*, that is, either it holds* $r = 0$*, or*

$$r = \sum_{i=1}^{r} a_i \cdot g_i, \ a_i \in A,$$

*satisfying* $\mathrm{LM}(r) \geq \mathrm{LM}(a_i \cdot g_i)$*, such that* $a_i \cdot g_i \neq 0$*, for all* $i$*.*

Most of the classical problems of ideal theory, e. g. ideal membership, variable elimination, equality of two ideals, etc. can be easily solved using Gröbner bases. To eliminate variables, we use **elimination orderings**.

Simply expressed, we can view them as a separator that makes everything we want to eliminate larger than the elements we want to keep. The best elimination orderings for fast Gröbner basis computations are the so called block orderings (cf. [2] Example 1.2.8.(3) ).

Because of the elimination property of Gröbner bases it seems advisable to use the Gröbner basis theory to obtain a good formulation for a given DAE system. We finish this section proving the following lemma.

**Lemma 10**
*Let $I \trianglelefteq A^{(q)}$ be a linear locally solvable DAE. Then $I$ is formally integrable too.*

PROOF
We have to show the $\mathcal{E} \circ \mathcal{P}(\mathcal{P}^k(I)) = \mathcal{P}^k(I)$ for all $k$. As the prolongation of a linear DAE is again linear we conclude that every $\mathcal{P}^k(I)$ is linear. Thus it suffices to show that $\mathcal{E} \circ \mathcal{P}(\mathcal{P}(I)) = \mathcal{P}(I)$. As $I$ is linear all polynomials in $I$ are of degree one. Hence $D_t(I)$ is simply a substitution of variables. So let $I$ be written as Gröbner basis w. r. t. a block elimination ordering $>$ on the ring variables satisfying

$$\{f^{(0]}\} < \{f'\} < \{f^{(2)}\} < \ldots < \{f^{(q)}\}.$$

Then the Gröbner basis $G$ of $I$ can be written in block form:

$$G = \{\underbrace{f_{q1}, \ldots, f_{qn_q}}_{order\ q}, \ldots, \underbrace{f_{11}, \ldots, f_{1,n_1}}_{order\ 1}, \underbrace{f_{01}, \ldots, f_{0n_0}}_{order\ 0}\}$$

Let $F_j = \{f_{j1}, \ldots, f_{jn_j}\}$. Now $D_t(G)$ is again a Gröbner basis as all variables are substituted. In fact

- $D_t(G) = \{D_t(F_q), D_t(F_{q-1}), \ldots, D_t(F_0)\}$
- $D_t^2(G) = \{D_t^2(F_q), D_t^2(F_{q-1}), \ldots, D_t^2(F_0)\}$.

Now

$$\begin{aligned}\mathcal{E}\circ\mathcal{P}(\mathcal{P}(I)) &= \mathcal{E}\circ\mathcal{P}(I+D_t(I))) \\ &= \mathcal{E}(I+D_t(I)+D_t(I+D_t(I))) \\ &= \mathcal{E}(I+D_t(I)+D_t^2(I)) \\ &= I+D_t(I)+\mathcal{E}(D_t^2(I)) = \mathcal{P}(I)+\mathcal{E}(D_t^2(I))\end{aligned}$$

As $P(I)\subseteq\mathcal{E}\circ\mathcal{P}(\mathcal{P}(I))$ always holds, it suffices to show the inclusion $\mathcal{E}\circ\mathcal{P}(\mathcal{P}(I))\subseteq\mathcal{P}(I)$. This reduces to $\mathcal{E}(D_t^2(I))\subseteq\mathcal{P}(I)$. Now

$$\begin{aligned}\mathcal{E}(D_t^2(I)) &= \langle D_t^2(F_{q-1}),\ldots,D_t^2(F_1),D_t^2(F_0)\rangle \\ &= D_t(\langle D_t(F_{q-1}),\ldots,D_t(F_1),D_t(F_0)\rangle) \\ &= D_t(\mathcal{E}(D_t(I)))\subseteq D_t(\mathcal{E}\circ\mathcal{P}(I)) \\ &= D_t(I)\subseteq\mathcal{P}(I)\end{aligned}$$

This proves our claim.

# 3 A computational approach

In the following section a computational approach for interacting the *Mathematica*-based tool Analog Insydes [1] with SINGULAR is explained. One of the main difficulties is to construct a communication bridge between both systems that come from different mathematical application domains. SINGULAR is well optimised for polynomials and Gröbner bases, while Analog Insydes is used for modelling and mixed numeric/symbolic approximation of analog circuits.

## 3.1 Differentiation and Prolongation

A natural way to implement differentiation is dealing with word rewriting systems. The derivative of a variable is simply represented by another variable. This rewriting process is obtained by the definition of new variables $df_i$ for $f_i'$, $ddf_i$ for $f_i''$ etc. Then differentiation is obtained by a left shift to the formal derivative. So, to obtain a correct differentiation we simply introduce a map $\varphi$ defining the derivative of every function. The core of the whole differentiation of pure polynomials in the $A^{(q)}$ is the product rule (see Algorithm 1).

**Algorithm 1 PROC** productrule (*poly* $f$, *map* $\varphi$)

**Require:** a polynomial (resp. monomial) $f \in A^q$ and the derivative map $\varphi$ of variables
**Ensure:** a polynomial $df$ which is $f' \in A^{(q+1)}$
  **if** $(\deg f = 0)$ **then**
    **return** 0;
  **if** $(\deg f = 1)$ **then**
    **return** $\varphi(f)$;
  **else**
    pick a prime factor $p$ of $f$;
    $g := \frac{f}{p}$
    $df := g \cdot \varphi(p) +$**productrule**$(g, \varphi) \cdot p$
  **return** $df$

Using this underlying core it is possible to obtain a procedure for the differentiation of an ideal. This procedure is called **derivideal**. It gets an ideal $I$ as input and the definition of the derivative map $\varphi$, the output is $D_t(I)$, the derivative of $I$. The prolongation is simply defined by the procedure **Prolongation** which takes the ideal *dae* and a natural number $functionanz$ as arguments where the latter denotes the number of involved functions and is just to generate the derivative map automatically. The following example which is derived from Example 1 shows how the procedure **Prolongation** works.

**Example 11**

```
> ring r=0,(dx(1..3),x(1..3),t),dp;
> ideal dae=dx(1)+x(1),x(2)*dx(2)-x(3),
  x(1)^2+x(2)^2-1;
//x(1),x(2),x(3) are the 3 functions
> def difring=Prolongation(dae,3);
> setring difring;
> dae;
dae[1]=dx(1)+x(1)
dae[2]=dx(2)*x(2)-x(3)
dae[3]=x(1)^2+x(2)^2-1
```

```
//ddx(i) is x(i)''
> prol;
prol[1]=dx(1)+x(1)
prol[2]=dx(2)*x(3)+x(2)*x(3)-dx(2)
prol[3]=x(2)^2+x(3)-1
prol[4]=dx(2)*x(2)-x(3)
prol[5]=x(1)^2-dx(2)*x(2)
prol[6]=ddx(2)+dx(2)^3+dx(2)^2*x(2)
       -dx(2)*dx(3)-dx(3)*x(2)
prol[7]=ddx(1)+dx(1)
```

The ideal **prol** is the Gröbner basis of the prolongation from Definition 4. The equation $x_1^2 - x_3 = 0$ (cf. Example 1) can be easily derived from **prol[4]** and **prol[5]**. The ring *difring* is ordered by a block ordering admitting $\{t, x_1, \ldots dx_3\} < \{ddx_1, ddx_2, ddx_3\}$. Hence, we see that prol[1..5] is the elimination of the highest derivatives in *prol*. So we see that after defining the prolongation with Gröbner bases it is an easy task to compute the elimination.

### 3.2 Computing locally solvable systems

**Algorithm 2 PROC** LocallySolvableDAE(ideal dae, int n)

**Require:** a DAE dae of $q$-th order in $A^{(q)}$, $N$ the number of functions
**Ensure:** a DAE locs of $q$-th order which is locally solvable and the differential index of dae

```
int difindex = 0;
ideal locs = dae;
ideal buffer = 0;
while buffer ≠ locs do
   buffer = locs;
   locs = InvolutionStep(locs);
   difindex = difindex + 1;
return (locs, difindex);
```

In the previous section we saw that the computation of $\mathcal{E} \circ \mathcal{P}(I)$ for every $I$ can be implemented easily. This is done in the auxilliary procedure **InvolutionStep**. The natural way to extend this to a procedure which returns a locally solvable system is described in Algorithm 2.

To finish the computations of Example 1 we see the following example:

**Example 12**

```
> ring r=0,(dx(1..3),x(1..3),t),dp;
> ideal dae=dx(1)+x(1),x(2)*dx(2)-x(3),
            x(1)^2+x(2)^2-1;
> LocallySolvableDAE(dae,3);
[1]:
   _[1]=dx(3)+2*x(3)
   _[2]=dx(1)+x(1)
   _[3]=dx(2)*x(3)+x(2)*x(3)-dx(2)
   _[4]=x(2)^2+x(3)-1
   _[5]=dx(2)*x(2)-x(3)
   _[6]=x(1)^2-x(3)
[2]:
   2
```

Here we see that the differential index of our system is 2 and the equality $x_1^2 - x_3 = 0$ (c.f Example 1) appears as the sixth in the result. The advantage is that we can now derive every hidden constraint in the original DAE from the resulting one. The next two sections will deal with some extensions to new functions that may be included in the nonlinear systems like exponential functions, sines, cosines and squareroots.

# 4 Integration of more function types

This section describes the main ideas how to extend the algorithmic approach from the last section to the case of rings including more function classes like exponential functions, sines, cosines and squareroots as extensions of $A^{(q)}$. These allow us to formalize the concept of DAE systems including the latter.

## 4.1 Special extension rings for Exponential functions

First, let us consider systems with exponential functions. Such systems occur in simple analog circuit consisting of transistors and diodes. The special diode equations are called Shockley equations. They explain the connection between current and voltage. The Shockley diode equation is

$$I = I_S \cdot (e^{q \cdot \frac{V_D}{k \cdot T}} - 1),$$

where $I$ is the diode current, $I_S$ is a scalar factor called the saturation current, $q$ - the elementary charge, $V_D$ is the voltage across the diode, $k$ - the Boltzmann constant and $T$ - the temperature.

The extension ring $B^{(q)}$ is defined as follows:

**Definition 13**
*Let $n_e$ be a natural number and $arg = arg_1, \ldots, arg_{n_e}$ be a list of polynomials in $A^{(q)}$. Then we define $e_i := e^{arg_i}$. Now the ring of special exponential DAEs of q-th order is denoted by*

$$B^{(q)}_{arg} := A^{(q)}[e_1, \ldots, e_{n_e}] = K[f^{(q)}, \ldots, f', f, t, e].$$

*If there is no confusion about $arg$, we simply write $B^{(q)}$ instead of $B^{(q)}_{arg}$.*

Note that for every list $arg$ there is another ring. So the prolongation and of course the index and the local solvability depend on $arg$. With the additional definition

$$D_t(e_i) = D_t(arg_i) \cdot e_i$$

we get our natural extensions of the above discussed theory. In the next subsection the programs defined in the previous section will be extended for these special rings.

## 4.2 A solution to the computational task of exponential functions

As written in Section 3 the core of prolongation is how to define the derivative map. To this end we have to get a method to get the exponents of the $e_i$ and to define the derivatives. As we already have an algorithm representing a derivative map to compute derivatives in $A^{(q)}$, call it $\varphi_{A^{(q)}}$ and **DerivPoly**$(\cdot, \varphi)$, we can describe the derivative map for $B^{(q)}$.

**Definition 14**
*Let $x$ be one of the variables in $B^{(q)}$. We define the derivative map $\varphi_{B^{(q)}}$ as follows*

$$\begin{aligned} \varphi_{B^{(q)}} : & B^{(q)} \to B^{(q+1)} \\ & x \mapsto \begin{cases} \varphi_{A^{(q)}}(x) & \text{if } x \in A^{(q)} \\ \mathbf{DerivPoly}(arg_i, \varphi_{A^{(q)}}) \cdot e_i & \text{if } x = e_i \end{cases} \end{aligned}$$

Now the prolongation can be extended to exponential functions easily if we know their number and $arg$.

### 4.3 How to expand to sines and cosines

As sine and cosine depend on each other by means of their derivatives we extend our ring $B^{(q)}_{arg}$ simultaneously with both sine and cosine on arguments in $B^{(q)}_{arg}$ as follows:

**Definition 15**
*Let $n_{trig}$ be any natural number and $trig_{arg} = trig_{arg_1}, \ldots, trig_{arg_{n_{trig}}}$ a list of polynomials in $B^{(q)}_{e_{arg}}$. We define $s_i := \sin(trig_{arg_i})$ and $c_i := \cos(trig_{arg_i})$ as the ring of special trigonometric functions of $q$-th order*

$$\begin{aligned} C^{(q)}_{trig_{arg}, e_{arg}} := & B^{(q)}_{e_{arg}}[s_1, \ldots, s_{n_{trig}}, c_1, \ldots, c_{n_{trig}}] = \\ & = K[f^{(q)}, \ldots, f', f, t, e, s, c]. \end{aligned}$$

*If there is no confusion about $trig_{arg}$ and $e_{arg}$ we simply write $C^{(q)}$ instead of $C^{(q)}_{trig, earg}$.*

To extend our theory of locally solvable systems we have to extract our derivative map.

**Definition 16**
*Let $x$ be one of the variables in $C^{(q)}$. We define the derivative map $\varphi_{C^{(q)}}$ as follows*

$$\begin{aligned}\varphi_{C^{(q)}} : & C^{(q)} \to C^{(q+1)} \\ & x \mapsto \begin{cases} \varphi_{C^{(q)}}(x) & \text{if } x \in B^{(q)} \\ \mathbf{DerivPoly}(trig_{arg_i}, \varphi_{B^{(q)}}) \cdot c_i & \text{if } x = s_i \\ \mathbf{DerivPoly}(trig_{arg_i}, \varphi_{B^{(q)}}) \cdot (-s_i) & \text{if } x = c_i \end{cases}\end{aligned}$$

As a last extension we present squareroots.

## 4.4 Extension to square-roots

As we consider polynomials and no rational functions we need to adjoint both, the squareroot $\sqrt{\cdot}$ and its multiplicative inverse $\frac{1}{\sqrt{\cdot}}$ to the ring $C^{(q)}$. This yields the following definition:

**Definition 17**
*Let $C_{e_{arg}, trig_{arg}}$ as in Definition 15 and let $sqrt_{arg} = sqrt_{arg_1}, \ldots, sqrt_{arg_{n_{sqrt}}} \in C^{(q)}$ be the list of the arguments $n_{sqrt}$ of the squareroot functions $\sqrt{\cdot}_i$. We define for the list $mf := e_{arg}, trig_{arg}, sqrt_{arg}$ the ring of special functions including exponential functions, sines, cosines and squareroots as follows:*

$$D_{mf} = C^{(q)}_{trig_{arg}, e_{arg}}[\sqrt{\cdot}_1, \ldots, \sqrt{\cdot}_{n_{sqrt}}, \frac{1}{\sqrt{\cdot}_1}, \ldots, \frac{1}{\sqrt{\cdot}_{n_{sqrt}}}].$$

Additionally extracting the derivative map for the new ring $D$ with

$$\begin{aligned}\varphi_D(\sqrt{\cdot}_i) &:= \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot \frac{1}{2} \cdot \frac{1}{\sqrt{\cdot}_i} \\ \varphi_D(\frac{1}{\sqrt{\cdot}_i}) &:= \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot \frac{-1}{2} \cdot \left(\frac{1}{\sqrt{\cdot}_i}\right)^3\end{aligned}$$

we get the desired prolongation.

## 4.5 The structure of $D$

$D$ fits to all algebraic combinations that are possible with exponential functions, sines, cosines and square-roots. Unfortunately, we have no a priori knowledge about their analytical and geometric relationship. The following subsection tries to fill this gap. First of all let us recall the geometric relationship between sines and cosines, that is

$$\sin(x)^2 + \cos(x)^2 = 1 \ \forall x.$$

Therefore, we have naturally $s_i^2 + c_i^2 = 1$ for all $i \in \{1, \dots, n_{trig}\}$. Because of the multiplicative relation between the square-roots and their reciprocal we have additionally the conditions $\sqrt{\cdot}_i \cdot \frac{1}{\sqrt{\cdot}} = 1$ for every $i$. Hence we get the following definition:

**Definition 18**
*Let $D_{mf}$ be defined as above (Definition 17), then we define the ideal $I_0$ to be*

$$I_0 = \langle s_i^2 + c_i^2 - 1, \sqrt{\cdot}_j \cdot \frac{1}{\sqrt{\cdot}_j} - 1, \sqrt{\cdot}_j^2 - sqrt_{arg_j} \ : \\ i \in \{1, \dots, n_{trig}\}, j \in \{1, \dots, n_{sqrt}\}\rangle$$

The ideal $I_0$ represents every algebraic combination which is trivially zero for this special classes. The main question is: does this suffice in general? Hence we want to show that $I_0$ is already locally solvable. Therefore we consider the following lemma.

**Lemma 19**
*Let $D$ and $I_0$ be defined as above, then $I_0$ is formally integrable, especially for the theory of locally solvable set in $D$ it is sufficient to compute the normal forms w. r. t. $I_0$ after every prolongation step.*

Proof
We will show that $D_t(I_0) \subset I_0$. Therefore we show that $D_t(g) \in I_0$ for every generator $g$ of $I_0$. We have the following cases:

1. $g = s_i^2 + c_i^2 - 1$ for some suitable $i$. Then

$$\begin{aligned}
D_t(g) &= D_t(s_i^2 + c_i^2 - 1) \\
&= D_t(s_i^2) + D_t(c_i^2) - D_t(1) \\
&= 2 \cdot s_i \cdot D_t(s_i) + 2 \cdot c_i \cdot D_t(c_i) - 0 \\
&= 2 \cdot \mathbf{DerivPoly}(trig_{arg_i}, \varphi_{B^{(q)}}) \cdot s_i \cdot c_i - \\
&\quad - 2 \cdot \mathbf{DerivPoly}(trig_{arg_i}, \varphi_{B^{(q)}}) \cdot c_i \cdot s_i \\
&= 0 \in I_0
\end{aligned}$$

2. $g = \sqrt{\cdot}_j \cdot \frac{1}{\sqrt{\cdot}_j} - 1$ for a suitable $j$. Then

$$\begin{aligned}
D_t(g) &= D_t(\sqrt{\cdot}_j \cdot \frac{1}{\sqrt{\cdot}_j} - 1) \\
&= D_t(\sqrt{\cdot}_j \cdot \frac{1}{\sqrt{\cdot}_j}) - D_t(1) \\
&= D_t(\sqrt{\cdot}_j) \cdot \frac{1}{\sqrt{\cdot}_j} + \sqrt{\cdot}_j \cdot D_t(\frac{1}{\sqrt{\cdot}_j}) - 0 \\
&= \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot \frac{1}{2} \cdot \frac{1}{\sqrt{\cdot}_i} \cdot \frac{1}{\sqrt{\cdot}_j} + \\
&\quad + \sqrt{\cdot}_j \cdot \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot \frac{-1}{2} \cdot \left(\frac{1}{\sqrt{\cdot}_i}\right)^3 \\
&= \frac{1}{2} \cdot \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot [\left(\frac{1}{\sqrt{\cdot}_i}\right)^2 - \\
&\quad - (\sqrt{\cdot}_i \left(\frac{1}{\sqrt{\cdot}_i}\right)) \left(\frac{1}{\sqrt{\cdot}_i}\right)^2] \\
&= \frac{1}{2} \cdot \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot \left(\frac{1}{\sqrt{\cdot}_i}\right)^2 \cdot [1 - (\sqrt{\cdot}_i \left(\frac{1}{\sqrt{\cdot}_i}\right))] \\
&= -\frac{1}{2} \cdot \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot \left(\frac{1}{\sqrt{\cdot}_i}\right)^2 \cdot g \in I_0
\end{aligned}$$

3. $g = \sqrt{\cdot}_j^2 - sqrt_{arg_j}$ for a suitable $j$. Then

$$\begin{aligned}
D_t(g) &= D_t(\sqrt{\cdot}_j^2 - sqrt_{arg_j}) \\
&= D_t(\sqrt{\cdot}_j^2) - D_t(sqrt_{arg_j}) \\
&= 2 \cdot \sqrt{\cdot}_j \cdot D_t(\sqrt{\cdot}_j) - \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \\
&= 2 \cdot \sqrt{\cdot}_j \cdot \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot \frac{1}{2} \cdot \frac{1}{\sqrt{\cdot}_i} - \\
&\quad - \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \\
&= \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot [2 \cdot \frac{1}{2} \cdot \sqrt{\cdot}_j \cdot \frac{1}{\sqrt{\cdot}_i} - 1] \\
&= \mathbf{DerivPoly}(sqrt_{arg_i}, \varphi_{C^{(q)}}) \cdot g \in I_0
\end{aligned}$$

Hence $\mathcal{P}(I_0) = \langle I_0, D_t(I_0) \rangle = I_0$ and thus $\mathcal{P}^k(I_0) = I_0$ for all $k \in \mathbb{N}$ and especially $\mathcal{E} \circ \mathcal{P}(\mathcal{P}^k(I_0)) = I_0 = \mathcal{P}^k(I_0)$ for all $k$ and thus $I_0$ is formally integrable.

Algorithm 3 extends Algorithm 2 following Lemma 19.

**Algorithm 3 PROC** LocallySolvableDAE(ideal dae,ideal $I_0$)

**Require:** a DAE dae of $q$-th order in $D^{(q)}$ and the ideal $I_0$ from Definition 18
**Ensure:** a DAE locs of $q$-th order which is locally solvable and the differential index of dae

```
int difindex = 0;
ideal locs = dae;
I_0=groebner(I_0);
ideal buffer = 0;
while buffer ≠ locs do
   buffer = locs;
   locs = NF(InvolutionStep(locs),I_0);
   difindex = difindex + 1;
return (locs, difindex);
```

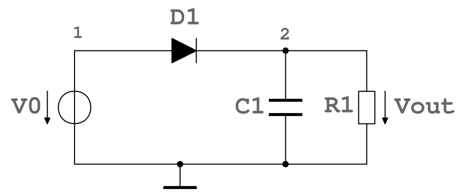


Figure 2. Analog rectifier circuit

# 5 Computational examples and outlook

To finish this article with the last section we will see how the gained theory applies to the following two examples. This section will be concluded with an outlook for further development.

## 5.1 Analog rectifier circuit

**Example 20**
*We have given the analog circuit in figure 2. This circuit contains both nonlinear as well as dynamic components, namely the diode $D1$ and the capacitor $C1$. Given numerical element values and a custom input voltage waveform $V_0 = V_{in}(t)$ we shall compute the transient response $V_{\mathrm{out}}(t)$ across the load resistor $R1$. The model parameters $I_S$ (saturation current) and $V_T = k\,T/q$ (thermal voltage) are given as $I_S = 1\,\mathrm{pA}$ and $V_T = 26\,\mathrm{mV}$. The values of the circuit elements are assumed to be $R1 = 100\,\Omega$ and $C1 = 100\,\mathrm{nF}$. This yields the following DAE system $(F)$:*

$$Ib_{ACdD1}(t) + Ib_{V0}(t) = 0 \tag{4.1}$$

$$-Ib_{ACdD1}(t) + \frac{V_{n2}(t)}{100} + \frac{V'_{n2}(t)}{10^7} = 0 \tag{4.2}$$

$$\frac{e^{\frac{V_{n1}(t)-V_{n2}(t)}{V_T}} - 1 + V_{n2}(t) - V_{n1}(t)}{10^{12}} = Ib_{ACdD1}(t) \tag{4.3}$$

*and the input condition $V_{n1}(t) = V_{in}(t)$.*
*The three equalities are transferred from Analog Insydes to* SIN-

GULAR, *which continues with computations in the polynomial ring* $\mathbb{Q}[V'_{n1}, V'_{n2}, Ib'_{V0}, Ib'_{ACdD1}, V_{n1}, V_{n2}, Ib_{V0}, Ib_{ACdD1}, t, e_1]$.

*Now the procedure* ***LocallySolvableDAE*** *returns the following system to* Mathematica

$$\begin{aligned}
Ib_{ACdD1}(t) + Ib_{V0}(t) &= 0 && (5.1)\\
Ib'_{ACdD1}(t) + Ib'_{V0}(t) &= 0 && (5.2)\\
1 + 10^{12} \cdot Ib_{ACdD1}(t) + V_{n2}(t) - e_1 &= V_{n1}(t) && (5.3)\\
10^5 \cdot (100 \cdot Ib_{ACdD1}(t) - V_{n2}(t)) &= V'_{n2}(t) && (5.4)\\
(e_1 + 1)(-10^5 \cdot (100 \cdot Ib_{ACdD1}(t) - & \\
-V_{n2}(t)) + V'_{n1}(t)) &= 10^{12} \cdot V_T \cdot Ib'_{ACdD1}(t) && (5.5)
\end{aligned}$$

*where* $e_1 = e^{\frac{V_{n1}(t) - V_{n2}(t)}{V_T}}$. *Using normal forms and our knowledge about* $V_{n1}$ *the system can be written as*

$$\begin{aligned}
V'_{n1}(t) &= V'_{in}(t) && (6.1)\\
V'_{n2}(t) &= f(t) && (6.2)\\
Ib'_{ACdD1}(t) &= -\frac{(e_1 + 1)(f(t) + V_{n2}(t)) - V'_{in}(t)}{10^{12} \cdot V_T} && (6.3)\\
Ib'_{V0}(t) &= -Ib'_{ACdD1}(t) && (6.4)\\
V_{n1}(t) &= V_{in}(t) && (6.5)\\
Ib_{ACdD1}(t) &= -Ib_{V0}(t) && (6.6)
\end{aligned}$$

*where* $f(t) = 10^5 \cdot (100 \cdot Ib_{ACdD1}(t) - V_{n2}(t))$. *This constrains the equations of an explicit ODE formulation in the variables* $V_{n1}, V_{n2}$ *and* $Ib_{ACdD1}$. *This is because* $V_{in}(t)$, *and hence* $V'_{in}(t)$, *have to be explicitly provided by the user. Therefore, the first three equations give formulas for* $V'_{n1}, V'_{n2}$ *and* $Ib'_{ACdD1}$, *which do not depend on unknown derivatives.*

## 5.2 A system including sines and cosines

**Example 21**

$$Ib_{ABdLC}(t) + Ib_{V0}(t) = 0 \tag{7.1}$$
$$D_{IdLC}(t) = Ib'_{ABdLC}(t) \tag{7.2}$$
$$-16\cos(Ib_{ABdLC}(t)) + D_{IdLC}(t) = 4 \tag{7.3}$$

*Although the system contains all necessary conditions, it can hardly be solved numerically without preprocessing. Processing the equation system above with the approach of locally solvable sets described earlier, we find the following equations:*

$$16\cos(Ib_{ABdLC}(t)) + 4 = D_{IdLC}(t) \tag{8.1}$$
$$Ib_{ABdLC}(t) + Ib_{V0}(t) = 0 \tag{8.2}$$
$$16\cos(Ib_{ABdLC}(t)) + 4 = Ib'_{ABdLC}(t) \tag{8.3}$$
$$Ib_{ABdLC}(t) + Ib'_{V0}(t) + 4 = 0 \tag{8.4}$$
$$64(\sin(Ib_{ABdLC}(t)) + 2\sin(2Ib_{ABdLC}(t))) + D'_{IdLC}(t) = 0 \tag{8.5}$$

*If we look at the system, we see that constraint (8.4) is redundant, as it is implicitly given by (8.2) and (8.3). If we additionally cut constraint (8.1) which is implicitly given in constraint (8.5), we get the following system ($F$):*

$$Ib_{ABdLC}(t) + Ib_{V0}(t) = 0 \tag{9.1}$$
$$16\cos(Ib_{ABdLC}(t)) + 4 = Ib'_{ABdLC}(t) \tag{9.2}$$
$$64(\sin(Ib_{ABdLC}(t)) + 2\sin(2Ib_{ABdLC}(t))) = D'_{IdLC}(t) \tag{9.3}$$

*With the initial conditions $D_{IdLC}(0) = 1, Ib_{ABdLC}(0) = 0$ and $Ib_{V0}(0) = 0$ the solution to the system computed by Analog Insydes can be seen in figure 3.*

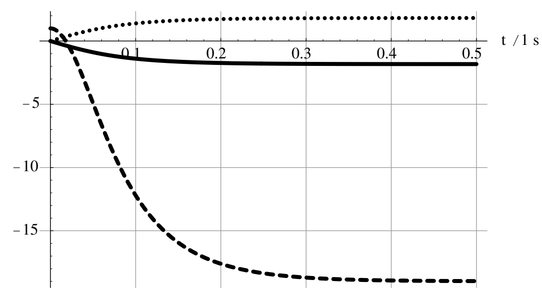


Figure 3. Time integration of 9.1–9.3, where $Ib_{ABdLC}$ $(\cdots)$, $D_{IdLC}$ (– –), and $Ib_{V0}$ (—).

## 5.3 Conclusion and outlook

We have embedded an important class of nonlinear DAE systems into a polynomial frame. This enables us to apply the theory of commutative algebra and Gröbner bases for modelling problems arising from analog circuit analysis. Therefore, we recalled some algebraic basics. We introduced algorithmic procedures for transforming DAEs to systems which are as close as possible to ODEs. After discussing polynomial nonlinear DAEs our approach was extended to systems containing exponential terms. This is an improvement of the known theory of local solvability and formal integrability (cf. [5], [4], [3]). This enables the analysis of important nonlinear components like diodes and transistors.

In further developments, because integrating further function types in Singular would only entail unnecessary work, we have decided to move the prolongation implementation to *Mathematica*. On one hand, this allows us to consider a larger spectrum of DAEs, and on the other hand, the prolongation process may use the specialized *Mathematica* differentiation functions. To transform between Mathematica and Singular representation, we define a mapping between *Mathematica* DAEs and Singular ideals.

The initial results of my research are very promising, more practical applications will be tackled in the future by using more sophisticated approaches. This will be published in a forthcoming article.

## References

[1] J. Broz, A. Dreyer, T. Halfmann, E. Hennig, M. Thole, and T. Wichmann. *Analog Insydes – Release 2.1.1 Manual*. Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, Kaiserslautern, Germany, 2007.

[2] G.-M. Greuel and G. Pfister. *A* **Singular** *introduction to commutative algebra*. Springer-Verlag, Berlin, 2002. With contributions by Olaf Bachmann, Christoph Lossen and Hans Schönemann, With 1 CD-ROM (Windows, Macintosh, and UNIX).

[3] M. Hausdorf. Geometrisch-algebraische Vervollständigung allgemeiner Systeme von Differentialgleichungen. *Diplomarbeit*, 2000.

[4] G. J. Reid, P. Lin, and A. D. Wittkopf. Differential Elimination-Completion Algorithms for DAE and PDAE. *Studies in Applied Mathematics*, 106(1):1–45, 2001.

[5] W. M. Seiler. Indices and solvability for general systems of differential equations. In *Computer algebra in scientific computing—CASC'99 (Munich)*, pages 365–385. Springer, Berlin, 1999.

[6] T. Wichmann. *Symbolische Reduktionsverfahren für nichtlineare DAE-Systeme*. Shaker Verlag, Aachen, Germany, 2004.

Silke J. Spang,
Received January 21, 2008

Fraunhofer Institute for Industrial Mathematics (ITWM)
Department System Analysis, Prognosis and Control
Kaiserslautern, Germany
E–mail: *silke.spang@itwm.fraunhofer.de*