Formal Computational Methods for Control Theory

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften
der Rheinisch-Westfälischen Technischen Hochschule Aachen zur
Erlangung des akademischen Grades eines Doktors der Naturwissenschaften
genehmigte Dissertation

vorgelegt von

Diplom-Mathematiker

Daniel Robertz

aus Aachen

Berichter: Universitätsprofessor Dr. Wilhelm Plesken
Universitätsprofessor Dr. Gerhard Jank

Tag der mündlichen Prüfung: 20.06.2006

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek
online verfügbar.

Contents

Chapter 1

Introduction

This thesis treats structural properties of control systems, e.g. controllability or parametrizability of their behavior, from an algebraic point of view. It is in the tradition of R. E. Kalman, who gave very important impetus to the structural analysis of control systems in the 1960s. Since then more and more algebraic methods have been applied and developed to gain insight into the behavior of control systems.

With this work I contribute to a few aspects of the links between algebra and control theory emphasizing formal methods and computational issues, which are also of independent interest. First I describe the main features of the employed method.

The control system is assumed to be modeled by a set of equations of a certain kind. In general, these equations are nonlinear. Then the algebraic approach which is pursued here considers a linearization of these equations. Now linear equations define a module over a ring which is chosen in accordance with the type of the given equations (e.g. ordinary or partial differential equations, difference equations, retarded differential equations, etc.). All consequences of the given equations are built into this module so that it represents the relations among the inspected quantities of the control system in an intrinsic way; in particular, two equivalent sets of linear equations give rise to the same module. The space of functions which is expected to contain solutions of the given equations is also assumed to be a module over the same ring; e.g. for differential equations the action on the function space is by differentiation, for difference equations by shifts, etc. Structural properties of the control system (e.g. the possibility to parametrize the solutions of the system in a formal way) are characterized by properties of the module which is defined by the given equations. However, these characterizations require a suitable interplay of the module defined by the equations and the module of admissible functions. In particular, the outlined approach is not applicable (at least in the full generality) to all kinds of function spaces. The most interesting properties of the module associated with the system equations can be dealt with by using homological algebra. This gives a unified approach to the structural properties and the possibilities to check them computationally. Among the structural properties of interest is the existence of so-called autonomous quantities, which cannot be influenced by a control. Whereas it is difficult in general to draw conclusions from the structural analysis of the linearized system about the properties of the given nonlinear one, sometimes the existence of such autonomous quantities for the original system can be confirmed or denied. If the linearized system is completely controllable, then the given nonlinear one has no autonomous quantities either. Conversely, if autonomous quantities are found for the linearized system, then one can try to lift them to the original system.

Among the developers of this algebraic approach are U. Oberst and E. Zerz who have been working mostly with equations with constant coefficients which give rise to commutative rings, and J.-F. Pommaret and A. Quadrat who were the first to tackle the non-commutative case.

The algebraic approach to control systems described above demands algorithms for symbolic computations at different steps of the strategy. If the given equations are nonlinear, an appropriate way to linearize them is required. The ring which is chosen in accordance with the type of the given equations needs to be dealt with constructively. Moreover, methods to compute with modules over these rings are fundamental in order to realize the constructions used from homological algebra.

Addressing the first issue, this thesis presents a method which results in a linearization of differential equations that is independent of any chosen trajectory. In particular, no solution of the set of equations is needed. This defines a generic linearization of the system (Chapter 3).

Secondly, Janet's algorithm is used for computing with rings and modules which arise in the present context. Given a finite generating set for a submodule of a free module of tuples over a commutative polynomial algebra, Janet's algorithm constructs another finite set, called Janet basis, which generates the same submodule, but consists of enough elements such that for any given tuple it can be decided whether it is an element of the submodule or not. More precisely, a Janet basis defines normal forms for the elements of the residue class module, enabling in this way effective computations in this module. As a tool for commutative algebra, Janet bases serve similar purposes as the more commonly known Gröbner bases do. Whereas Gröbner bases were introduced in the 1930s and Buchberger's algorithm, which computes Gröbner bases, was developed in the 1960s, Janet's theory from the 1920s even roots in earlier work of Méray and Riquier in the last decades of the 19th century. Janet's formal approach to systems of differential equations seemed forgotten for a long time, but was revived by J.-F. Pommaret.

In this thesis, Janet's algorithm is a keystone. In Chapter 2, I generalize Janet's algorithm from the well-known cases of commutative polynomial algebras and the Weyl algebra to Ore algebras which are of interest in the following chapters. For system theoretic applications Ore algebras are sufficiently general because the most important types of equations define Ore algebras, whose elements represent the operators which are involved in the equations.

The symbolic treatment of linear equations which arise from a generic linearization of differential equations as announced above requires another adaptation of Janet's algorithm. Since the generic linearization results in a system of linear equations with non-constant coefficients, these coefficients are subject to the original nonlinear equations. In particular, division by zero needs to be prevented when performing the arithmetics in the corresponding ring. In Chapter 3, I propose a straightforward way for dealing with these equations, which is presented in the framework of jet calculus and differential rings. The nonlinear differential equations are viewed as rewriting rules for elements of a differential ring. In general, it might be difficult to handle them effectively. Then linearization defines a module over this ring. Quite often one can determine the structure of this module without having a full command on the ring described by the nonlinear equations. Technically speaking, Janet's algorithm has to apply the given rules possibly to each coefficient in every step.

In Chapter 4, I present a very useful combinatorial tool, the generalized Hilbert series, which enumerates a vector space basis of a finitely presented module over an Ore algebra for which Janet's algorithm is applicable. The precision in which structural properties of the solution space of a linear system are represented by the module defined by the equations depends on the choice of the space of admissible functions. A faithful correspondence of homological conditions holds for function spaces which are injective cogenerators. For linear systems with constant coefficients some common function spaces are known to be injective cogenerators. In the case of ordinary or partial differential equations the generalized Hilbert series represents the freedom to choose a solution. For instance, formal power series solutions can be chosen through their Taylor coefficients; the coefficients which can be chosen arbitrarily are given by the generalized Hilbert series. For linear systems with non-constant coefficients very few injective cogenerators are known. In this case it is much more difficult to grasp the solutions. Still it makes sense to study the module which is defined by the equations. Janet's algorithm is applicable and provides the generalized Hilbert series. However, if the chosen space of functions is not an injective cogenerator, then it does not reflect all structural properties. In Chapter 4, I define an injective cogenerator for every Ore algebra which is relevant for the later chapters, but in general I can give no analytic interpretation of this module.

Applications of these methods are presented in Chapter 5. The decision about controllability and the detection of autonomous quantities of a given control system is demonstrated. The check of the structural properties needs a symbolic treatment of the system equations, so that the algebraic approach lends itself to a preprocessing for a numerical simulation of such systems. In particular, a reactor considered in chemical engineering is examined.

In Chapter 6 the possibility to represent solutions of a linear system of equations as the image of a certain operator is investigated more closely. The established theory about this way of parametrizing linear systems is recalled and common work with A. Quadrat in this context is presented. The notion of parametrization is extended to certain linear systems which are not completely controllable and a method to compute bases of free left modules of rank at least 2 over the Weyl algebras is described, which plays a crucial role for the construction of injective parametrizations.

A great portion of work was spent on implementations in Maple which enabled the present work. First of all, Janet's algorithm is available now for many rings through the packages `Involutive`, `Janet`, which I continued to develop, and `JanetOre`, which I initiated. The intentions of these packages are rather distinct (commutative polynomials, differential equations resp. Ore algebras) so that several applications benefit from these implementations. Moreover, I joined the `OreModules` project. This Maple package, originally developed by F. Chyzak and A. Quadrat, is intended to solve various system theoretic problems using the symbolic computational approach.

The final chapter demonstrates the Maple package `OreModules` on a stirred tank model. The package `JanetOre` is used to perform the necessary Janet basis computations. Chapter 7 is supposed to illustrate much of the theory mentioned above, but also gives examples of facilities of `OreModules` on system theoretic problems not elucidated in the previous chapters.

I would like to express my gratitude to several persons; first of all to Prof. W. Plesken for his supervision and support, secondly to Prof. J.-F. Pommaret and Dr. A. Quadrat who introduced us to the field of algebraic analysis. Moreover, during my visits of A. Quadrat at INRIA Sophia Antipolis I always enjoyed the very fruitful discussions.

Our cooperation with Prof. V. P. Gerdt and Dr. Y. A. Blinkov has also been very productive and successful in the realm of Janet and involutive bases; I would like to thank them, too.

Furthermore, I appreciate the support of my second supervisor Prof. G. Jank, in particular his control theory lecture.

I would like to thank the process systems engineering group of Prof. W. Marquardt (RWTH Aachen) for providing me with the reactor model of Example 5.3.3.

Finally, I thank all colleagues at Lehrstuhl B für Mathematik, RWTH Aachen, for a very nice atmosphere and especially Dr. M. Barakat for a lot of discussions.

Chapter 2

Janet's Algorithm

Janet's algorithm is named after Maurice Janet [Jan29] who developed it for the structural analysis of systems of (linear) partial differential equations. It allows to parametrize the formal power series solutions of a given system of (linear) partial differential equations by arranging a scheme which determines for each Taylor coefficient whether it can be chosen arbitrarily (in accordance with boundary conditions) to obtain a solution or whether it depends (linearly) on these choices. In particular it allows to determine the dimension of the vector space of the formal power series solutions. The formal approach to differential equations used in this context traces back to Ch. Méray [Mér80] and Ch. Riquier [Riq10]. After the subject had fallen into oblivion for many years, it was popularized again by J.-F. Pommaret [Pom94].

A linear system of partial differential equations with constant coefficients gives rise to a system of algebraic equations and vice versa by associating with a derivative the monomial formed by the variables used for differentiation. In this way, Janet's algorithm also becomes a tool for solving systems of polynomial equations. In fact, a Janet basis resulting from Janet's algorithm is a particular generating set for an ideal of a polynomial ring. Computations in the corresponding residue class ring can be performed using the Janet basis because it defines a normal form for the representatives of each residue class. The scheme for the Taylor coefficients, translated into the polynomial setting, accordingly provides combinatorial information about the set of common zeros of the ideal under consideration. In this context, Janet's algorithm can be viewed as a simultaneous generalization of both Euclid's algorithm (univariate polynomials) and Gaussian elimination (linear polynomials).

As a device for dealing with polynomial equations, Janet's algorithm is a very efficient alternative for Buchberger's algorithm computing Gröbner bases of polynomial ideals. The additional combinatorial features of Janet's algorithm however lead to a method of reducing a polynomial modulo other polynomials in a unique way. This combinatorial part has been generalized by V. P. Gerdt and Y. A. Blinkov in the framework of involutive divisions [GB98a], [GB98b]. On this basis they have been designing efficient algorithms to compute involutive bases (the analogues of Janet bases for more general involutive divisions).

In this chapter, Janet's algorithm is adapted to a certain class of Ore algebras, which are non-commutative polynomial rings. Ore algebras usually arise as algebras of linear operators, e.g. differential operators or difference operators. The most important examples for this thesis are the Weyl algebra and Ore algebras containing shift operators. Gröbner bases for algebras of solvable type were investigated by A. Kandri-Rody and V. Weispfenning in [KRW90]. Buchberger's algorithm to compute Gröbner bases has been generalized to Ore algebras by F. Chyzak (see [Chy98], [CS98], where it is also applied to the study of special functions and combinatorial sequences). Involutive divisions were studied for the Weyl algebra case by W. M. Seiler [HSS02]. Gröbner bases were also addressed in the framework of G-algebras by V. Levandovskyy [Lev05]. Very recently, the concept of involutive division was extended to non-commutative rings in the thesis [Eva06]. However, an adaptation of Janet's algorithm to Ore algebras does not need this generality, so that this thesis gives a presentation of it using only the necessary concepts. Restricted to Ore algebras, the exposition of Janet's algorithm is as efficient as for commutative polynomial rings.

A description of the fundamental ideas of Janet's algorithm was given in [PR05]. In this chapter we present more details and more precise descriptions of the sub-algorithms. However, the emphasis is put on the structural viewpoint. For efficiency issues we refer to the involutive basis algorithms [Ger05]. The first section explains the main combinatorial process in Janet's algorithm which accomplishes a specific sub-division of a certain set of monomials into cones of monomials. In Section 2.2, this method of decomposing monomial sets into disjoint cones is applied to the set of leading monomials of an ideal of a (commutative) polynomial ring. Janet reduction is defined and used in Janet's algorithm to construct a particular generating set of an ideal (or more generally a submodule of a finitely generated free module over a polynomial ring) which is called a Janet basis. Section 2.3 introduces the generalized Hilbert series and the Janet graph as combinatorial tools for the study of finitely generated modules over polynomial rings. After defining Ore algebras and recalling their most important properties in Section 2.4, Janet's algorithm is adapted to a certain class of Ore algebras in Section 2.5. Some implementations are discussed in Section 2.6. Finally, a simple tool for reducing the complexity for Janet basis computations is presented in Section 2.7.

Janet's algorithm is basic for the rest of this thesis. In particular, possibilities to extend it to differential rings will be discussed in Section 3.3, and it will be applied in the subsequent chapters.

2.1 Decomposition of Sets of Monomials into Disjoint Cones

The exposition of Janet's algorithm for commutative polynomial rings in this and the next section follows [PR05], but gives more details and more precise algorithms. We start by describing a method for partitioning a set of monomials into a finite set of "cones" of monomials. It will be applied in the next section to the set of leading monomials of a generating set for a submodule of a finitely generated free module over a polynomial ring. This combinatorial procedure steers the course of Janet's algorithm to find a particular generating set for this module which is called a Janet basis. The presented method is demonstrated in this section on an easy example of two-variate monomials and on an example involving three variables which shows the generic behavior of this procedure.

Let k be a field[1] and $R := k[x_1, \ldots, x_n]$ the (commutative) polynomial algebra over k. We define the set of *monomials* of R resp. in $\{x_1, \ldots, x_n\}$ by

$$\mathrm{Mon}(R) := \mathrm{Mon}(\{x_1, \ldots, x_n\}) := \{x^a \mid a \in (\mathbb{Z}_{\geq 0})^n\}, \qquad x^a := x_1^{a_1} \cdots x_n^{a_n},$$

which is the free commutative monoid on $x_1, \ldots, x_n$. The divisibility relation for monomials is defined by:

$$x^a \mid x^b \quad \Longleftrightarrow \quad b_i \geq a_i \quad \text{for all} \quad i = 1, \ldots, n \qquad (a, b \in (\mathbb{Z}_{\geq 0})^n).$$

Moreover, for each $\mu \subseteq \{x_1, \ldots, x_n\}$ we set

$$\mathrm{Mon}(\mu) := \{x^a \mid a \in (\mathbb{Z}_{\geq 0})^n;\ a_i = 0 \text{ for all } 1 \leq i \leq n \text{ such that } x_i \notin \mu\}.$$

Finally, for any set M we denote by $\mathcal{P}(M)$ the power set of M.

In this section we consider subsets S of $\mathrm{Mon}(R)$ which have the following property:

Definition 2.1.1. A set $S \subseteq \mathrm{Mon}(R)$ is said to be $\mathrm{Mon}(R)$-*multiple closed*, if

$$ms \in S \quad \text{for all } m \in \mathrm{Mon}(R),\ s \in S.$$

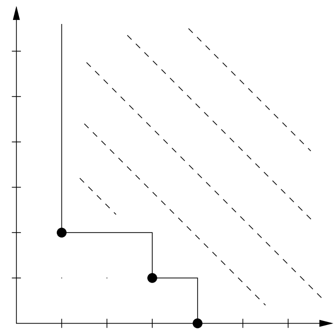
Every set $G \subseteq \mathrm{Mon}(R)$ satisfying

$$\mathrm{Mon}(R)G = \{mg \mid m \in \mathrm{Mon}(R), g \in G\} = S$$

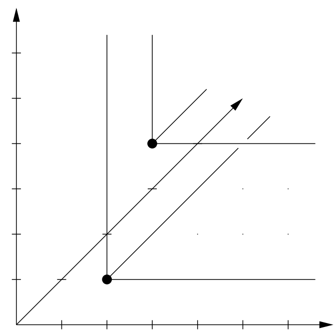
is called a *generating set* for S. By $[G]$ we denote the $\mathrm{Mon}(R)$-multiple closed set generated by G in $\mathrm{Mon}(R)$.

[1] For this section, the domain of coefficients of R is irrelevant. Hence, k could also be chosen to be, e.g., a commutative ring with 1.

Example 2.1.2. Let $R = k[x_1, x_2]$ and $G := \{x_1 x_2^2,\ x_1^3 x_2,\ x_1^4\}$. We consider the $\mathrm{Mon}(R)$-multiple closed set $S = [G]$ generated by G. If we visualize the monomial $x_1^i x_2^j$ as the point (i, j) in the positive quadrant of a x_1-x_2-coordinate system, then the set S of monomials can be viewed as the discrete set of points in the upper-right region in the following figure:



Example 2.1.3. Let $R = k[x_1, x_2, x_3]$ and $G := \{x_1 x_2,\ x_2^3 x_3\}$. The $\mathrm{Mon}(R)$-multiple closed set $S = [G]$ can be visualized in a similar way as in the previous example as the discrete set of points in the positive octant of the x_1-x_2-x_3-coordinate system shown in the next figure:



The proof of the following very important lemma is given along the lines of [Jan29].

Lemma 2.1.4. *Every* $\mathrm{Mon}(R)$*-multiple closed set has a finite generating set.*

Proof. By induction on n we show that every sequence F of monomials m_1, m_2, $\ldots$ in $\{x_1, \ldots, x_n\}$ with the property that no m_i is divisible by any previous m_j, $j < i$, is a finite sequence. The case of univariate monomials is clear. For the induction step, let $m_1 = x_1^{a_1} \cdots x_n^{a_n}$. For each $1 \leq j \leq n$ and $0 \leq d \leq a_j$ let $F^{(j,d)}$ be the subsequence of F consisting of all monomials for which the exponent of x_j equals d. The union of the elements of all $F^{(j,d)}$, $1 \leq j \leq n$, $0 \leq d \leq a_j$, already exhausts the sequence F: for every monomial m in F having a greater power of

x_j than m_1 there is a variable x_l whose exponent e in m is less than the exponent of x_l in m_1 because m_1 does not divide m. Therefore m is in the subsequence $F^{(l,e)}$. Now the induction hypothesis can be applied to each $F^{(j,d)}$ by ignoring the variable x_j. Altogether, F is thus a finite sequence of monomials. □

Corollary 2.1.5. *Every ascending sequence of* $\mathrm{Mon}(R)$*-multiple closed sets becomes stationary.*

Proposition 2.1.6. *Every* $\mathrm{Mon}(R)$*-multiple closed set has a unique minimal generating set.*

Proof. Any generating set G for a $\mathrm{Mon}(R)$-multiple closed set can be reduced to a minimal generating set by removing those elements g from G for which there exists $h \in G - \{g\}$ such that $h \mid g$.

Let us assume that G_1 and G_2 are two minimal generating sets for the same $\mathrm{Mon}(R)$-multiple closed set S. Let $g_1 \in G_1$. Then there exists $g_2 \in G_2$ such that $g_2 \mid g_1$ because G_2 generates S. Since G_1 is also a generating set for S, there exists $\widetilde{g}_1 \in G_1$ satisfying $\widetilde{g}_1 \mid g_2$. It follows $\widetilde{g}_1 \mid g_1$, which by the minimality of G_1 implies $\widetilde{g}_1 = g_2 = g_1$. By reversing the roles of G_1 and G_2 we find $G_1 = G_2$. □

Definition 2.1.7. A pair $(C, \mu) \in \mathcal{P}(\mathrm{Mon}(R)) \times \mathcal{P}(\{x_1, \ldots, x_n\})$ is called a *cone* if there exists $v \in C$ such that

$$\mathrm{Mon}(\mu)v = \{mv \mid m \in \mathrm{Mon}(\mu)\} = C.$$

Then $v(C) := v$ is called the *vertex* of the cone (C, μ) and the elements of μ are the *multiplicative variables* for (C, μ) (or simply for C). Finally, $\overline{\mu} := \{x_1, \ldots, x_n\} - \mu$ denotes the set of *non-multiplicative variables* for (C, μ) (or simply for C).

Remark 2.1.8. The vertex $v(C)$ of a cone (C, μ) is well-defined since it is the unique monomial in C with minimal total degree.

The purpose of this section is to describe a method to represent certain sets of monomials as a disjoint union of the monomials of finitely many cones. For instance, every union of cones of monomials can be written as a finite union of disjoint cones of monomials. However, our main application will be to $\mathrm{Mon}(R)$-multiple closed sets and their complements in $\mathrm{Mon}(R)$.

Definition 2.1.9. Let S be any set of monomials in $\mathrm{Mon}(R)$. A *decomposition of S into disjoint cones* is given by a finite set

$$\{\, (C_1, \mu_1), \ldots, (C_l, \mu_l) \,\} \subset \mathcal{P}(\mathrm{Mon}(R)) \times \mathcal{P}(\{x_1, \ldots, x_n\})$$

such that each pair (C_i, μ_i) is a cone,

$$\bigcup_{i=1}^{l} C_i = S \quad \text{and} \quad C_i \cap C_j = \emptyset \quad \text{for all} \quad i \neq j.$$

Passing over to cone vertices, we will also call a finite set

$$T = \{ (m_1, \mu_1), \ldots, (m_l, \mu_l) \} \subset \mathrm{Mon}(R) \times \mathcal{P}(\{x_1, \ldots, x_n\})$$

a *decomposition of S into disjoint cones* if

$$\{ (\mathrm{Mon}(\mu_1) m_1, \mu_1), \ldots, (\mathrm{Mon}(\mu_l) m_l, \mu_l) \} \tag{2.1}$$

satisfies the above conditions. Then, T is also said to be *complete*.

Given a finite set $\{m_1, \ldots, m_l\}$ of monomials, there are many possible ways of how to arrange sets of multiplicative variables $\mu_1, \ldots, \mu_l$ such that (2.1) is a set of cones such that $\mathrm{Mon}(\mu_i) m_i \cap \mathrm{Mon}(\mu_j) m_j = \emptyset$ for $i \neq j$. These possibilities are addressed by the notion of *involutive division* which we will not discuss in detail here. Important for us is only the Janet division:

Definition 2.1.10. [GB98a] Let $M \subset \mathrm{Mon}(R)$ be finite. For each $m \in M$, the *Janet division* defines the set μ of multiplicative variables for the cone generated by m as follows. Let $m = x^a \in M$. For $1 \leq i \leq n$, we have:

$$x_i \in \mu \quad \Longleftrightarrow \quad \max\{b_i \mid x^b \in M;\, b_j = a_j \text{ for all } j < i\} = a_i,$$

i.e., x_i is a multiplicative variable for the cone generated by m if and only if its exponent in m is maximal among all exponents of monomials in M which have the same sequence of exponents of $x_1, \ldots, x_{i-1}$ as m.

There are also other common involutive divisions, e.g. J. Thomas [Tho37] proposed another way of defining the multiplicative variables of cones.

Let $S \subseteq \mathrm{Mon}(R)$ be $\mathrm{Mon}(R)$-multiple closed. We are going to describe next how to decompose S into disjoint cones. The result will provide the separation of $\{x_1, \ldots, x_n\}$ into multiplicative and non-multiplicative variables for each cone in accordance with the Janet division. The method is a recursive scan through the tuples of exponents of the elements of S. Therefore we fix a permutation $\sigma \in S_n$ and set $y_i := x_{\sigma(i)}$, $i = 1, \ldots, n$. Without loss of generality one could just restrict to the case $y_i = x_i$, $i = 1, \ldots, n$. However, distinct permutations $\sigma_1 \neq \sigma_2 \in S_n$ lead to different decompositions of S into disjoint cones in general.

Given a generating set G for the $\mathrm{Mon}(R)$-multiple closed set S, the following algorithm constructs a finite set of cones such that their sets of monomials are disjoint and the union of these sets of monomials equals S.

Algorithm 2.1.11 (Decompose)**.**

Input: (G, η), where $G \subset \mathrm{Mon}(\{y_1, \ldots, y_n\})$ is finite and $\emptyset \neq \eta \subseteq \{y_1, \ldots, y_n\}$

Output: $\{\, (m_1, \mu_1), \ldots, (m_l, \mu_l) \,\} \subset \mathrm{Mon}(\{y_1, \ldots, y_n\}) \times \mathcal{P}(\eta)$, a decomposition of $\mathrm{Mon}(\eta)G$ into disjoint cones

Algorithm:

// *minimize the generating set G (see Prop. 2.1.6):*
$G \leftarrow \{g \in G \mid \nexists\, h \in G : h \mid g\}$
if $|G| \leq 1$ **or** $|\eta| = 1$ **then**
  return $\{(m, \eta) \mid m \in G\}$
else
  // *determine the first variable y in the sequence $y_1, \ldots, y_n$ with $y \in \eta$:*
  $y \leftarrow y_a$ with $a = \min\{i \mid 1 \leq i \leq n, y_i \in \eta\}$
  // *partition G into sets of monomials with the same degree in y:*
  $d \leftarrow \max\{\deg_y(g) \mid g \in G\}$
  $G_i \leftarrow \{g \in G \mid \deg_y(g) = i\}, \qquad i = 0, \ldots, d$
  // *add monomials not covered by cones due to y becoming non-mult.:*
  $G_i \leftarrow G_i \cup \bigcup_{j=0}^{i-1} \{y^{i-j} g \mid g \in G_j\}, \qquad i = 1, \ldots, d$
  // *for all cones resulting from G_d, y is a multiplicative variable:*
  $T_d \leftarrow \{\, (m, \zeta \cup \{y\}) \mid (m, \zeta) \in \mathrm{Decompose}(G_d, \eta - \{y\}) \,\}$
  // *for all cones resulting from G_i, $i < d$, y is a non-multiplicative variable:*
  $T_i \leftarrow \mathrm{Decompose}(G_i, \eta - \{y\}), \qquad i = 0, \ldots, d-1$
  return $\bigcup_{i=0}^{d} T_i$
fi

Theorem 2.1.12. *Let S be a $\mathrm{Mon}(\{y_1, \ldots, y_n\})$-multiple closed set. Given a generating set G for S,*

$$\mathrm{Decompose}(G, \{y_1, \ldots, y_n\})$$

returns a decomposition of S into disjoint cones. This decomposition only depends on the set S and the choice of $\sigma \in S_n$ defining $y_i := x_{\sigma(i)}$, $i = 1, \ldots, n$. It does not depend on the choice of G.

Proof. Termination of Alg. 2.1.11 is clear because, if the algorithm does not return in step 4, then it is called recursively with less and less variables in η until reaching the case $|\eta| = 1$.

We show the correctness of Alg. 2.1.11 by induction on $|\eta|$. First of all, a $\mathrm{Mon}(\eta)$-multiple closed set generated by a single element is already a cone. If $|\eta| = 1$, then the result in step 4 is a decomposition of $\mathrm{Mon}(\eta)G$ into disjoint cones because no monomial in G divides any other monomial in G due to step 2. In any other case, G is partitioned into sets of monomials distinguished by their degree in y. Let us assume that Alg. 2.1.11 is correct for all second arguments of smaller size than $|\eta|$. Then we first assert that

$$\mathrm{Mon}(\mu_1)m_1 \cap \mathrm{Mon}(\mu_2)m_2 = \emptyset \qquad \text{for all} \quad (m_1, \mu_1), (m_2, \mu_2) \in T_i, \quad i = 0, \ldots, d.$$

This assertion holds for $i = d$ because mutually disjoint cones of monomials of the same degree d in y for which y is a non-multiplicative variable stay disjoint when y gets a multiplicative variable again. For $i = 0, \ldots, d-1$, the assertion is true simply because of the induction hypothesis. Since y is only chosen to be multiplicative for the cones resulting from G_d, and cones resulting from different G_i, $i < d$, contain only monomials of distinct degree in y, it is clear that the cones generated by elements in $\bigcup_{i=0}^{d} T_i$ are mutually disjoint.

Finally, we show that

$$\bigcup_{(m,\mu) \in \bigcup_{i=0}^{d} T_i} \mathrm{Mon}(\mu)m = \mathrm{Mon}(\eta)G. \tag{2.2}$$

By the induction hypothesis we have

$$\begin{cases} \bigcup_{(m,\mu) \in T_d} \mathrm{Mon}(\mu)m & = & \mathrm{Mon}(\eta)G_d, \\ \bigcup_{(m,\mu) \in T_i} \mathrm{Mon}(\mu)m & = & \mathrm{Mon}(\eta - \{y\})G_i, \quad i = 0, \ldots, d-1. \end{cases} \tag{2.3}$$

The "completion step" (step 12) ensures that

$$\{m \in \mathrm{Mon}(\eta)G_d \mid \deg_y(m) = i\} = \mathrm{Mon}(\eta - \{y\})G_i.$$

Therefore, (2.3) implies (2.2).

That the result of Alg. 2.1.11 does not depend on the choice of the generating set G follows directly from the uniqueness of a minimal generating set for S (see Prop. 2.1.6). $\square$

In the next corollary we conclude that the method of decomposing $\mathrm{Mon}(R)$-multiple closed sets of monomials into disjoint cones also achieves the separation of $\{y_1, \ldots, y_n\}$ into multiplicative and non-multiplicative variables for each cone as defined by the Janet division.

Corollary 2.1.13. *Let T be the output of Algorithm 2.1.11 and $(m, \mu) \in T$ with $m = y^a$, $a \in (\mathbb{Z}_{\geq 0})^n$. Then, for $1 \leq i \leq n$, we have:*

$$y_i \in \mu \qquad \Longleftrightarrow \qquad \max\{b_i \mid (y^b, \nu) \in T;\ b_j = a_j \textit{ for all } j < i\} = a_i.$$

Proof. The case $|T| = 1$ is clear. Let $|T| > 1$. Then T is obtained by recursive calls of Algorithm 2.1.11. From steps 14 and 16 it is apparent that y_i is a multiplicative variable for m if and only if $m \in G_d$ in the run of Decompose for which $y = y_i$, i.e. if and only if m has maximal exponent in y_i among all monomials in $\{c \mid (c, \nu) \in T\}$ having the same starting sequence of exponents as m. □

Example 2.1.14. Let us apply Alg. 2.1.11 to the Mon(R)-multiple closed set S from Ex. 2.1.2. We choose $y_1 = x_1$, $y_2 = x_2$. The input for Alg. 2.1.11 is (G, η) with $G = \{x_1 x_2^2,\ x_1^3 x_2,\ x_1^4\}$ and $\eta = \{x_1, x_2\}$. The algorithm switches to the "else"-case and determines the first variable $y = x_1$ in η. Therefore, the set G is partitioned into subsets G_i according to the degree of the given monomials in $y = x_1$. Hence, we obtain $d = 4$ and

$$G_4 = \{x_1^4\}, \quad G_3 = \{x_1^3 x_2\}, \quad G_2 = \emptyset, \quad G_1 = \{x_1 x_2^2\}, \quad G_0 = \emptyset.$$

The "completion step" (step 12) causes the following changes:

$$G_4 = \{x_1^4,\ x_1^4 x_2,\ x_1^4 x_2^2\}, \quad G_3 = \{x_1^3 x_2,\ x_1^3 x_2^2\}, \quad G_2 = \{x_1^2 x_2^2\}.$$

The algorithm is applied recursively to $(G_i, \{x_2\})$, $i = 0, \ldots, 4$. For all these inputs, step 2 computes a minimal generating set containing at most one element. In particular, the multiples of $x_1^3 x_2$ and $x_1^3 x_2^2$ added to G_4 and G_3 in the "completion step" are removed immediately in the recursive runs. However, the addition of $x_1^2 x_2^2$ to G_2 is necessary. For an illustration of the case where a "completed" G_i cannot be minimized to a set with a single element see the next example.

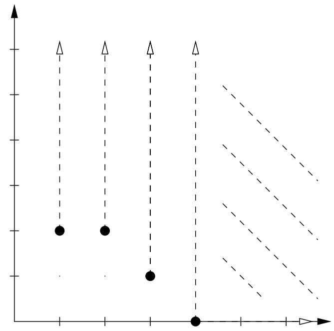
The runs of Decompose$(G_i, \{x_2\})$, $i = 0, \ldots, 4$, return with values

$$\{\,(x_1^4, \{x_2\})\,\}, \quad \{\,(x_1^3 x_2, \{x_2\})\,\}, \quad \{\,(x_1^2 x_2^2, \{x_2\})\,\}, \quad \{\,(x_1 x_2^2, \{x_2\})\,\}, \quad \emptyset.$$

The set G_d from the partition of G containing the monomials with greatest exponent in $y = x_1$ yields a "two-dimensional" cone, whereas the other sets G_i, $i < d$, result in "one-dimensional" cones, i.e. cones with one multiplicative and one non-multiplicative variable. The final result

$$\{\,(x_1^4, \{x_1, x_2\})\,\}, \quad \{\,(x_1^3 x_2, \{x_2\})\,\}, \quad \{\,(x_1^2 x_2^2, \{x_2\})\,\}, \quad \{\,(x_1 x_2^2, \{x_2\})\,\} \tag{2.4}$$

is visualized in the following figure:



Example 2.1.15. We are going to find a decomposition of the $\mathrm{Mon}(R)$-multiple closed set S in Ex. 2.1.3. Now we choose $y_1 = x_2$, $y_2 = x_1$, $y_3 = x_3$. Note that now our decomposition method distinguishes monomials by their degree in x_2 first, then by their degree in x_1. The input for Alg. 2.1.11 is (G, η) with $G = \{x_1x_2,\, x_2^3x_3\}$ and $\eta = \{x_1, x_2, x_3\}$. Since no of the trivial cases in step 3 is present, G is partitioned into subsets of monomials according to their degree in x_2, where $d = 3$ is the maximal occurring degree:

$$G_3 = \{x_2^3x_3\}, \quad G_2 = \emptyset, \quad G_1 = \{x_1x_2\}, \quad G_0 = \emptyset.$$

The "completion step" (step 12) yields:

$$G_3 = \{x_2^3x_3,\, x_1x_2^3\}, \quad G_2 = \{x_1x_2^2\}.$$

Note that G_3 is already a minimal generating set for $\mathrm{Mon}(\{x_1, x_3\})G_3$. Therefore, $\mathrm{Decompose}(G_3,\, \{x_1, x_3\})$ sub-divides G_3 into sets of monomials distinguished by their degree in x_1. We denote the local variables for this recursive call by a tilde. Then we find $\widetilde{d} = 1$ and the sets

$$\widetilde{G}_1 = \{x_1x_2^3\}, \quad \widetilde{G}_0 = \{x_2^3x_3\},$$

where $\widetilde{G}_1$ is "completed" to

$$\widetilde{G}_1 = \{x_1x_2^3,\, x_1x_2^3x_3\}.$$

By an additional recursive call, $\widetilde{G}_1$, $\widetilde{G}_0$ yield the cones

$$\{\, (x_1x_2^3, \{x_1, x_3\})\,\}, \quad \{\, (x_2^3x_3, \{x_3\})\,\}.$$

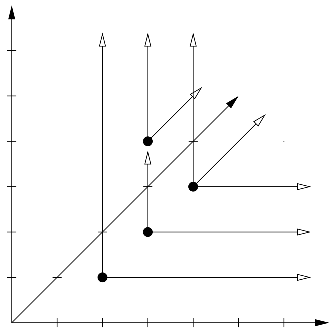
Moreover, G_2, G_1 from above result in the cones

$$\{\, (x_1x_2^2, \{x_1, x_3\})\,\}, \quad \{\, (x_1x_2, \{x_1, x_3\})\,\}.$$

Therefore, the final result is

$$\{\, (x_1x_2^3, \{x_1, x_2, x_3\})\,\}, \{\, (x_2^3x_3, \{x_2, x_3\})\,\}, \{\, (x_1x_2^2, \{x_1, x_3\})\,\}, \{\, (x_1x_2, \{x_1, x_3\})\,\},$$

which is depicted in the next figure:



In order to obtain a decomposition of S with the least number of cones, one could unite the cones with vertices x_1x_2, $x_1x_2^2$, $x_1x_2^3$. We would have found this minimal decomposition, if we had chosen $y_1 = x_1$, $y_2 = x_2$, $y_3 = x_3$, but in this example cones were constructed by comparing the degrees of monomials in x_2 first.

For applications of Janet bases, a decomposition of the complement $\mathrm{Mon}(R) - S$ of a $\mathrm{Mon}(R)$-multiple closed set S is more important than the decomposition of S (see Chapter 4). The next algorithm constructs such a decomposition of $\mathrm{Mon}(R) - S$. It is a modification of Algorithm 2.1.11 (Decompose) which yields cones of monomials in the complement of S instead of the decomposition of S. Of course, $\mathrm{Mon}(R) - S$ is $\mathrm{Mon}(R)$-multiple closed only if $S = \emptyset$.

Algorithm 2.1.16 (DecomposeComplement)**.**

Input: (G, v, η), where $G \subset \mathrm{Mon}(\{y_1, \ldots, y_n\})$ is finite, $v \in \mathrm{Mon}(\{y_1, \ldots, y_n\})$ with $v \mid g$ for all $g \in G$, and $\emptyset \neq \eta \subseteq \{y_1, \ldots, y_n\}$

Output: $\{\, (m_1, \mu_1), \ldots, (m_l, \mu_l) \,\} \subset \mathrm{Mon}(\{y_1, \ldots, y_n\}) \times \mathcal{P}(\eta)$, a decomposition of $\mathrm{Mon}(\eta)v - \mathrm{Mon}(\eta)G$ into disjoint cones

Algorithm:

// minimize the generating set G (see Prop. 2.1.6):

$G \leftarrow \{g \in G \mid \nexists\, h \in G : h \mid g\}$

if $G = \emptyset$ **then** *// the complement equals* $\mathrm{Mon}(\eta)v$*, which is a cone*

  return $\{\, (v, \eta) \,\}$

elif $|\eta| = 1$ **then** *// the complement is a finite set of monomials*

  return $\{\, (mv, \emptyset) \mid m \in \mathrm{Mon}(\eta), mv \notin \mathrm{Mon}(\eta)G\}$

else

  // determine the first variable y in the sequence $y_1, \ldots, y_n$ *with* $y \in \eta$*:*

  $y \leftarrow y_a$ with $a = \min\{i \mid 1 \leq i \leq n, y_i \in \eta\}$

```
10: // partition G into sets of monomials with the same degree in y:
11: d ← max{deg_y(g) | g ∈ G}
12: G_i ← {g ∈ G | deg_y(g) = i},        i = 0, ..., d
13: // add monomials not covered by cones due to y becoming non-mult.:
14: G_i ← G_i ∪ ⋃_{j=0}^{i-1} {y^{i-j} g | g ∈ G_j},        i = 1, ..., d
15: // for all cones resulting from G_d, y is a multiplicative variable:
16: T_d ← { (m, ζ ∪ {y}) | (m, ζ) ∈ DecomposeComplement(G_d, y^d v, η − {y}) }
17: // for all cones resulting from G_i, i < d, y is a non-multiplicative variable:
18: T_i ← DecomposeComplement(G_i, y^i v, η − {y}),        i = 0, ..., d − 1
19: return ⋃_{i=0}^{d} T_i
20: fi
```

Theorem 2.1.17. *Let S be a $\mathrm{Mon}(\{y_1, \ldots, y_n\})$-multiple closed set. Given a generating set G for S,*

$$\mathrm{DecomposeComplement}(G, 1, \{y_1, \ldots, y_n\})$$

returns a decomposition of the complement $\mathrm{Mon}(\{y_1, \ldots, y_n\}) - S$ of S into disjoint cones. This decomposition only depends on the set S and the choice of $\sigma \in S_n$ defining $y_i := x_{\sigma(i)}$, $i = 1, \ldots, n$. It does not depend on the choice of G.

Proof. Termination of Alg. 2.1.16 is clear because it is called recursively with less and less variables in η until $|\eta| = 1$.

If G is empty, then (v, η) is a trivial decomposition of $\mathrm{Mon}(\eta)v$ into disjoint cones proving the correctness of step 4. If $|\eta| = 1$, then the algorithm enumerates the monomials in the complement of $\mathrm{Mon}(\eta)G$ in $\mathrm{Mon}(\eta)v$ which are finitely many. These monomials provide cones with no multiplicative variables. The rest of Alg. 2.1.16 is similar to Alg. 2.1.11, in particular the recursive treatment of the partition of G into sets of monomials according to their degrees in y. The only difference is the additional argument v which comprises the information in which set $\mathrm{Mon}(\eta)v$ the complement is to be taken. The rest of the argumentation carries over from the proof of Theorem 2.1.12. □

Example 2.1.18. We apply Alg. 2.1.16 to the same data as in Ex. 2.1.14 in order to find a decomposition of $\mathrm{Mon}(\{x_1, x_2, x_3\}) - S$ into disjoint cones. The input for Alg. 2.1.16 is

$$G = \{x_1 x_2^2,\ x_1^3 x_2,\ x_1^4\}, \quad v = 1, \quad \eta = \{x_1, x_2\}.$$

Since $G \neq \emptyset$ and $|\eta| > 1$, the set G is partitioned and "completed" in the same way as in Ex. 2.1.14. We have $d = 4$ and

$$G_4 = \{x_1^4,\ x_1^4x_2,\ x_1^4x_2^2\},\ G_3 = \{x_1^3x_2,\ x_1^3x_2^2\},\ G_2 = \{x_1^2x_2^2\},\ G_1 = \{x_1x_2^2\},\ G_0 = \emptyset.$$

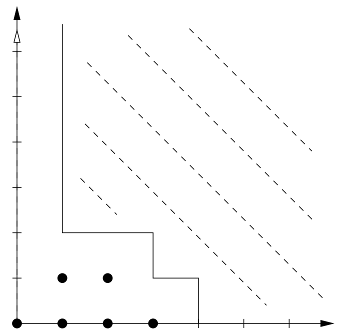
In the next table the recursive calls of Alg. 2.1.16 are listed:

Input	Output
$(G_4, x_1^4, \{x_2\})$	$\emptyset$
$(G_3, x_1^3, \{x_2\})$	$\{\,(x_1^3, \emptyset)\,\}$
$(G_2, x_1^2, \{x_2\})$	$\{\,(x_1^2, \emptyset), (x_1^2x_2, \emptyset)\,\}$
$(G_1, x_1, \{x_2\})$	$\{\,(x_1, \emptyset), (x_1x_2, \emptyset)\,\}$
$(G_0, 1, \{x_2\})$	$\{\,(1, \{x_2\})\,\}$

The final result is:

$$\{\,(1, \{x_2\}), (x_1, \emptyset), (x_1x_2, \emptyset), (x_1^2, \emptyset), (x_1^2x_2, \emptyset), (x_1^3, \emptyset)\,\}. \tag{2.5}$$

This decomposition is depicted in the following figure:



Example 2.1.19. We reconsider Ex. 2.1.15, where $y_1 = x_2$, $y_2 = x_1$, $y_3 = x_3$. In order to decompose the complement $\mathrm{Mon}(\{x_1, x_2, x_3\}) - [G]$ into disjoint cones we apply Alg. 2.1.16 to

$$G = \{x_1x_2,\ x_2^3x_3\}, \qquad v = 1, \qquad \eta = \{x_1, x_2, x_3\}.$$

The algorithm obtains the same partition of G as in Ex. 2.1.15, which is "completed" to

$$G_3 = \{x_2^3x_3,\ x_1x_2^3\}, \quad G_2 = \{x_1x_2^2\}, \quad G_1 = \{x_1x_2\}, \quad G_0 = \emptyset.$$

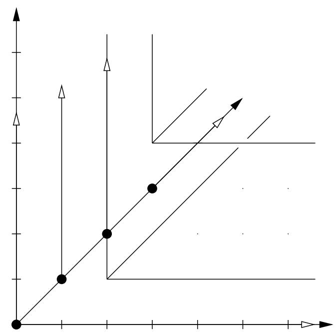
The recursive calls of Alg. 2.1.16 are listed in the following table:

Input	Output
$(G_3, x_2^3, \{x_1, x_3\})$	$\{\,(x_2^3, \{x_2\})\,\}$
$(G_2, x_2^2, \{x_1, x_3\})$	$\{\,(x_2^2, \{x_3\})\,\}$
$(G_1, x_2, \{x_1, x_3\})$	$\{\,(x_2, \{x_3\})\,\}$
$(G_0, 1, \{x_1, x_3\})$	$\{\,(1, \{x_1, x_3\})\,\}$

Therefore, the final result is

$$\{ (1, \{x_1, x_3\}), (x_2, \{x_3\}), (x_2^2, \{x_3\}), (x_2^3, \{x_2\}) \},$$

which is shown in the next figure:



2.2 Janet's Algorithm

In this section, Janet's algorithm for commutative polynomial rings over fields is described. An adaptation of it to Ore algebras, which are in general non-commutative rings, is presented in Section 2.5.

The combinatorial part of Janet's algorithm is driven by the set of leading monomials of a given generating set for a module of tuples over a polynomial ring. Therefore, monomial orderings which define the leading monomial of a polynomial are discussed first. Note that the combinatorial methods in the previous section depend on a certain order of the variables, but no ordering on the whole set of monomials is needed. In Janet's algorithm the generating set from the input is possibly enlarged or reduced in many steps in order to finally achieve one whose set of leading monomials generates the multiple-closed set of all leading monomials of the module under consideration. The method of decomposing multiple-closed sets of monomials as described in Section 2.1 is applied to the set of leading monomials. It indicates by the non-multiplicative variables how new leading monomials can be found. Then, Janet-reduction is defined which produces normal forms of representatives for residue classes modulo the given module. After giving the definition of Janet bases, Janet's algorithm to compute a Janet basis is presented.

Let k be a field, $R := k[x_1, \ldots, x_n]$, and $q \in \mathbb{N}$. In this section we denote by $e_1, \ldots, e_q$ the elements of the standard basis of the free R-module R^q, and we set $\mathrm{Mon}(R^q) := \bigcup_{i=1}^q \mathrm{Mon}(R)e_i$. For $m_1 e_i, m_2 e_j \in \mathrm{Mon}(R^q)$ we define

$$m_1 e_i \mid m_2 e_j \quad \Longleftrightarrow \quad i = j \quad \text{and} \quad m_1 \mid m_2.$$

After fixing a total ordering $<$ on $\mathrm{Mon}(R^q)$ the monomials with non-zero coefficient in a tuple $p \in R^q$ can be compared with respect to $<$. If $p \neq 0$, then the greatest of these monomials is called the *leading monomial* of p (with respect to $<$). The coefficient of the leading monomial of p is called the *leading coefficient* of p. If the ordering $<$ on $\mathrm{Mon}(R^q)$ is clear from the context, then references to it will be omitted so that the leading monomial resp. coefficient of p will just be denoted by $\mathrm{lm}(p)$ resp. $\mathrm{lc}(p)$ in what follows.

For Janet's algorithm we always choose the total ordering $<$ on $\mathrm{Mon}(R^q)$ to be compatible with the multiplication $\mathrm{Mon}(R) \times \mathrm{Mon}(R^q) \to \mathrm{Mon}(R^q)$ and to be a well-ordering.

Definition 2.2.1. A *monomial ordering* on $\mathrm{Mon}(R^q) = \bigcup_{i=1}^{q} \mathrm{Mon}(R)e_i$ is a total ordering on $\mathrm{Mon}(R^q)$ satisfying for all $1 \leq i \leq q$:

(a) $e_i < me_i$ for all $1 \neq m \in \mathrm{Mon}(R)$ and

(b) $m_1e_i < m_2e_i$ with $m_1, m_2 \in \mathrm{Mon}(R)$ implies $xm_1e_i < xm_2e_i$ for all $x \in \{x_1, \ldots, x_n\}$.

Examples 2.2.2. The most common monomial orderings in the case $q = 1$ are defined as follows. We let $m_1 = x_1^{a_1} \cdots x_n^{a_n}$, $m_2 = x_1^{b_1} \cdots x_n^{b_n} \in \mathrm{Mon}(R)$.

(a) The *degree reverse lexicographical ordering* (degrevlex) on $\mathrm{Mon}(R)$ is defined by:

$$m_1 < m_2 \quad \text{if and only if} \quad \begin{array}{l} \deg m_1 < \deg m_2 \ \text{ or} \\ \deg m_1 = \deg m_2 \ \text{ and } \ m_1 \neq m_2 \ \text{ and } \ a_i > b_i \\ \text{for } \ i = \max\{j \mid a_j \neq b_j\}. \end{array}$$

(b) The *lexicographical ordering* (lex) on $\mathrm{Mon}(R)$ is defined by:

$$m_1 < m_2 \quad \text{if and only if} \quad m_1 \neq m_2 \ \text{ and } \ a_i < b_i \ \text{ for } \ i = \min\{j \mid a_j \neq b_j\}.$$

The degree reverse lexicographical ordering is used in most situations. If p_1 and p_2 are two non-zero polynomials with $\mathrm{lm}(p_1) \mid \mathrm{lm}(p_2)$, where the leading monomials are determined with respect to degrevlex, then a suitable multiple of p_1 can be subtracted from p_2 in order to obtain a polynomial r such that $\mathrm{lm}(r)$ is less than $\mathrm{lm}(p_2)$ with respect to degrevlex. This also holds with respect to lex, but when degrevlex is chosen, then it is clear that $\deg(\mathrm{lm}(r)) \leq \deg(\mathrm{lm}(p_2))$, i.e. the total degree does not increase.

The lexicographical ordering can be exploited in Janet's algorithm to find tuples in a given module $M \leq R^q$ which contain only the least variable with respect to lex. Since in general the total degree of a polynomial does not decrease in the reduction process described above, computations with the lexicographical ordering may be very time-consuming.

Remark 2.2.3. The most useful ways of extending a monomial ordering $<$ on $\mathrm{Mon}(R)$ to a monomial ordering on $\mathrm{Mon}(R^q)$ for $q > 1$ are the following ($1 \leq i, j \leq q$, $m_1, m_2 \in \mathrm{Mon}(R)$):

(a) $m_1 e_i <_{\mathrm{TOP}} m_2 e_j$ if and only if $m_1 < m_2$ or $m_1 = m_2$ and $i < j$ ("term over position" ordering).

(b) $m_1 e_i <_{\mathrm{POT}} m_2 e_j$ if and only if $i < j$ or $i = j$ and $m_1 < m_2$ ("position over term" ordering).

(c) Fix some $1 \leq r \leq q$. Define $m_1 e_i <_{\mathrm{ELIM}} m_2 e_j$ if and only if

$$(i < r \text{ and } j \geq r) \text{ or } (\text{not } (i \geq r \text{ and } j < r) \text{ and } m_1 < m_2).$$

The monomial ordering $<_{\mathrm{TOP}}$ allows one to select the $<$-greatest monomial in all non-zero entries of a tuple $0 \neq p \in R^q$, whereas $<_{\mathrm{POT}}$ defines the leading monomial of p to be the leading monomial with respect to $<$ of the last non-zero entry of p. The monomial ordering $<_{\mathrm{ELIM}}$ can be used in Janet's algorithm to find tuples in a given module $M \leq R^q$ whose last $(q - r + 1)$ entries are zero.

In what follows we fix a monomial ordering $<$ on $\mathrm{Mon}(R^q)$.

Remark 2.2.4. The concept of $\mathrm{Mon}(R)$-multiple closed sets in $\mathrm{Mon}(R)$ carries over to $\mathrm{Mon}(R)$-*multiple closed sets in* $\mathrm{Mon}(R^q)$. Definition 2.1.7 of a cone and Definition 2.1.9 of a decomposition of a monomial set into disjoint cones are extended in a similar way. Moreover, every result of Section 2.1 is easily adapted to the more general case of $\mathrm{Mon}(R)$-multiple closed sets $S \subseteq \mathrm{Mon}(R^q)$ and their complements by treating each subset $S_i := S \cap \mathrm{Mon}(R) e_i$, $i = 1, \ldots, q$, separately. In particular, decompositions of $\mathrm{Mon}(R)$-multiple closed sets and their complements into disjoint cones can be constructed by straightforward extensions of Algorithm 2.1.11 and Algorithm 2.1.16.

The link between submodules of R^q and $\mathrm{Mon}(R)$-multiple closed sets, as described in the following important remark, and the results of Section 2.1 provide the combinatorial part of Janet's algorithm.

Remark 2.2.5. For every submodule M of R^q the set of leading monomials

$$\mathrm{lm}(M) := \{\mathrm{lm}(p) \mid 0 \neq p \in M\}$$

is $\mathrm{Mon}(R)$-multiple closed. Let L be a generating set for M. Following Section 2.1, a decomposition

$$\{\,(m_1, \mu_1), \ldots, (m_l, \mu_l)\,\} \subset \mathrm{Mon}(R^q) \times \mathcal{P}(\{x_1, \ldots, x_n\})$$

of $[\mathrm{lm}(L)]$ into disjoint cones can be constructed (for a fixed order of the variables). In fact, Algorithm 2.1.11 (Decompose) can be applied directly to a finite generating set $G \subset R^q - \{0\}$ of M by considering for decisions of the algorithm in each element its leading monomial only, but performing multiplications ("completion step" 12) on the actual generators and not only on their leading monomials. Apart from that, the only necessary modification in Algorithm 2.1.11 is to replace $\deg_y(\,\cdot\,)$ by $\deg_y(\mathrm{lm}(\,\cdot\,))$. In Janet's algorithm, Decompose will be applied in this adapted version.

Let M be a submodule of R^q. Starting with a finite generating set L of M, Janet's algorithm possibly removes elements from L and inserts new elements of M into L repeatedly in order to achieve finally that $[\mathrm{lm}(L)] = \mathrm{lm}(M)$. An element $p \in L$ is removed if it is reduced to zero by subtraction of suitable multiples of other elements of L. This is described now as the process of auto-reduction.

Definition 2.2.6. A finite set $L \subset R^q - \{0\}$ is said to be *auto-reduced* if no monomial occurring with non-zero coefficient in any $p_1 \in L$ is divisible by any $\mathrm{lm}(p_2)$, $p_2 \in L$.

There is an obvious way of auto-reducing a given set L of (tuples of) polynomials, namely by subtracting suitable multiples of elements $p_1 \in L$ from other elements $p_2 \in L$.

Algorithm 2.2.7 (Auto-reduce)**.**

Input: $(L, <)$, where $L \subset R^q$ is a finite set and $<$ is a monomial ordering on $\mathrm{Mon}(R^q)$

Output: A finite set $L' \subset R^q - \{0\}$ such that $\langle L' \rangle = \langle L \rangle$ and L' is auto-reduced

Algorithm:

$L' \leftarrow L - \{0\}$
while $\exists\, p_1, p_2 \in L'$, $p_1 \neq p_2$: p_2 has a monomial m with coefficient $c \neq 0$ such that $\mathrm{lm}(p_1) \mid m$ **do**
    $L' \leftarrow L' - \{p_2\}$
    $r \leftarrow p_2 - \frac{c}{\mathrm{lc}(p_1)} \frac{m}{\mathrm{lm}(p_1)} p_1$
    if $r \neq 0$ **then**
        $L' \leftarrow L' \cup \{r\}$
    fi
od
return L'

The output of Algorithm 2.2.7 depends on the order in which reductions are performed. However, our intention is to construct any auto-reduced set L' satisfying $[\mathrm{lm}(L)] \subseteq [\mathrm{lm}(L')]$. Obviously, this property is ensured. When the monomials m in step 2 are chosen in decreasing order with respect to $<$, then it is clear that Algorithm 2.2.7 terminates.

Next we describe a reduction process for (tuples of) polynomials which takes the Janet division into account and is therefore uniquely determined.

Definition 2.2.8. Let $T = \{ (b_1, \mu_1), \ldots, (b_l, \mu_l) \}$ be a subset of $(R^q - \{0\}) \times \mathcal{P}(\{x_1, \ldots, x_n\})$.

(a) T is said to be *complete*, if $\{ (\mathrm{lm}(b_1), \mu_1), \ldots, (\mathrm{lm}(b_l), \mu_l) \}$ is a decomposition into disjoint cones of the $\mathrm{Mon}(R)$-multiple closed set generated by $\{ \mathrm{lm}(b_1), \ldots, \mathrm{lm}(b_l) \}$.

(b) $p \in R^q$ is *Janet-reducible modulo T* if p contains some $m \in \mathrm{Mon}(R^q)$ with non-zero coefficient for which there is $(b, \mu) \in T$ such that $m \in \mathrm{Mon}(\mu)b$. In this case, (b, μ) is called a *Janet-divisor* of p. If p is not Janet-reducible modulo T, then p is also said to be *Janet-reduced modulo T*.

The following algorithm subtracts suitable multiples of Janet-divisors from a given $p \in R^q$ as long as a monomial in p is Janet-reducible.

Algorithm 2.2.9 (Janet-reduce)**.**

Input: $(p, T, <)$, where $p \in R^q$, $T = \{ (b_1, \mu_1), \ldots, (b_l, \mu_l) \} \subset (R^q - \{0\}) \times \mathcal{P}(\{x_1, \ldots, x_n\})$ is complete, and $<$ is a monomial ordering on $\mathrm{Mon}(R^q)$

Output: $r \in R^q$ such that $r + \langle b_1, \ldots, b_l \rangle = p + \langle b_1, \ldots, b_l \rangle$ and r is Janet-reduced modulo T

Algorithm:

$r \leftarrow 0$
while $p \neq 0$ **do**
  if $\exists\, (b, \mu) \in T : \mathrm{lm}(p) \in \mathrm{Mon}(\mu) b$ **then**
    $p \leftarrow p - \frac{\mathrm{lc}(p)}{\mathrm{lc}(b)} \frac{\mathrm{lm}(p)}{\mathrm{lm}(b)}\, b$
  else
    $r \leftarrow r + \mathrm{lc}(p)\, \mathrm{lm}(p)$
    $p \leftarrow p - \mathrm{lc}(p)\, \mathrm{lm}(p)$
  fi
od
return r

Remarks 2.2.10. (a) Alg. 2.2.9 terminates because, as long as $p \neq 0$, the leading monomial of p gets properly smaller with respect to the monomial ordering $<$ and $<$ is a well-ordering. The result r of Alg. 2.2.9 is uniquely defined for the given input because T is complete. As opposed to reduction procedures for (tuples of) polynomials disregarding multiplicative variables, the course of Alg. 2.2.9 is uniquely determined.

(b) Let $p_1, p_2 \in R^q$ and T be as in the input of Alg. 2.2.9. In general, the equality $p_1 + \langle b_1, \ldots, b_l \rangle = p_2 + \langle b_1, \ldots, b_l \rangle$ does not imply that the results of applying Janet-reduce to p_1 and p_2 are equal. But later on (see Theorem 2.2.13) it is shown that, if T is a Janet basis, then the result of Janet-reduce constitutes a unique representative for every coset in $R^q / \langle b_1, \ldots, b_l \rangle$. It is called the *Janet-normal form of* p *modulo* T. In order to simplify notation, we denote the *result of Janet-reduce* by $\mathrm{NF}(p, T)$, even if T is not a Janet basis.

Definition 2.2.11. Let $T = \{ (b_1, \mu_1), \ldots, (b_l, \mu_l) \}$ be a subset of $(R^q - \{0\}) \times \mathcal{P}(\{x_1, \ldots, x_n\})$ and assume T is complete (see Def. 2.2.8 (a)). Then T is said to be *passive*, if $\mathrm{NF}(x \cdot b_i, T) = 0$ holds for all $x \in \overline{\mu_i}$, $1 \le i \le l$. In this case T is also called a *Janet basis for* $\langle b_1, \ldots, b_l \rangle \le R^q$.

Now Janet's algorithm is presented which computes a Janet basis for a given submodule of R^q. For the decomposition of $\mathrm{Mon}(R)$-multiple closed sets, an order of the variables is fixed (see Section 2.1 and Remark 2.2.5).

Algorithm 2.2.12 (JanetBasis)**.**

Input: $(L, <)$, where $L \subset R^q$ is a finite set and $<$ is a monomial ordering on $\mathrm{Mon}(R^q)$

Output: $J \subset R^q \times \mathcal{P}(\{x_1, \ldots, x_n\})$, a Janet basis satisfying $\langle p \mid (p, \mu) \in J \rangle = \langle L \rangle$

Algorithm:

```
1: G ← L
2: do
3:     G ← Auto-reduce(G)                                  // see Alg. 2.2.7
4:     J ← Decompose(G, {x_1, ..., x_n})                   // see Rem. 2.2.5
5:     P ← { NF(x · p, J) | (p, μ) ∈ J, x ∉ μ }            // see Alg. 2.2.9
6:     G ← { p | (p, μ) ∈ J } ∪ P
7: od while P ≠ {0}
8: return J
```

Theorem 2.2.13. *(a) Algorithm 2.2.12 terminates and is correct.*

(b) A k-basis of $\langle L \rangle$ is given by $\dot{\bigcup}_{(g,\mu) \in J} \mathrm{Mon}(\mu) g$*, where J is the result of Algorithm 2.2.12.*

(c) A k-basis of $R^q / \langle L \rangle$ is given by the cosets represented by $\dot{\bigcup}_{(m,\mu) \in T} \mathrm{Mon}(\mu) m$*, where T is a decomposition of* $\mathrm{Mon}(R^q) - [\{\mathrm{lm}(p) \mid (p, \mu) \in J\}]$ *into disjoint cones.*

(d) *Given $p_1, p_2 \in R^q$, we have*

$$p_1 + \langle L \rangle = p_2 + \langle L \rangle \iff \mathrm{NF}(p_1, J) = \mathrm{NF}(p_2, J).$$

Proof. (a) First we show that JanetBasis terminates. Auto-reduction of G possibly enlarges $[\mathrm{lm}(G)]$. Decompose only augments the generating set G by tuples p of polynomials with $\mathrm{lm}(p) \in [\mathrm{lm}(G)]$, if it is necessary for the chosen way of decomposing $[\mathrm{lm}(G)]$ into disjoint cones. In any case it ensures $[\{\mathrm{lm}(p) \mid (p, \mu) \in J\}] = [\mathrm{lm}(G)]$. If all Janet-normal forms in step 5 are zero, then the algorithm terminates. If $P \neq \{0\}$, then $G' := G \cup P$ satisfies $[\mathrm{lm}(G)] \subsetneq [\mathrm{lm}(G')]$. By Corollary 2.1.5, after finitely many steps we have $[\mathrm{lm}(G)] = [\mathrm{lm}(G')]$ which is equivalent to $P = \{0\}$. Therefore, JanetBasis terminates in any case.

For the correctness we note that the set J resulting in step 4 is complete. Therefore $\mathrm{NF}(x \cdot p, J)$ in step 5 is well-defined. Once $P = \{0\}$ holds in step 7, J is passive, thus a Janet basis. The equality $\langle p \mid (p, \mu) \in J \rangle = \langle L \rangle$ holds in every round of the algorithm.

(b) Set $B := \bigcup_{(g,\mu) \in J} \mathrm{Mon}(\mu) g$.

For the k-linear independence of B we note first that $0 \notin B$ holds because J is constructed by completing an auto-reduced set of polynomials. Furthermore, $\mathrm{lm}(p_1) \neq \mathrm{lm}(p_2)$ for all $p_1, p_2 \in B$ with $p_1 \neq p_2$ because J is complete, which proves that B is k-linearly independent.

We show that B is a generating set for the k-vector space $\langle L \rangle$. Let $0 \neq p \in \langle L \rangle$. Since $\mathrm{lm}(\langle L \rangle) = [\{\mathrm{lm}(p) \mid (p, \mu) \in J\}]$, p is Janet-reducible modulo J. For the Janet-normal form of p modulo J we have $\mathrm{NF}(p, J) \in \langle L \rangle$, and $\mathrm{NF}(p, J)$ is not Janet-reducible modulo J. The previous argument implies $\mathrm{NF}(p, J) = 0$. Therefore, $p \in \langle B \rangle$.

(c) Since $[\{\mathrm{lm}(p) \mid (p, \mu) \in J\}] = \mathrm{lm}(\langle L \rangle)$, T is a decomposition of the complement $\mathrm{Mon}(R^q) - \mathrm{lm}(\langle L \rangle)$ into disjoint cones. Hence, the cosets represented by $\bigcup_{(m,\mu) \in T} \mathrm{Mon}(\mu) m$ are k-linearly independent. We show that they generate $R^q / \langle L \rangle$ as a k-vector space. Let $0 \neq r \in R^q / \langle L \rangle$ and choose any representative $p \in R^q$ of the coset r. Then $\mathrm{NF}(p, J)$ is also a representative of r, and $\mathrm{NF}(p, J) \neq 0$ because otherwise $p \in \langle L \rangle$ and $r = 0$. Janet-reduction (Alg. 2.2.9) ensures that the set of monomials occurring with non-zero coefficient in $\mathrm{NF}(p, J)$ has empty intersection with $[\{\mathrm{lm}(p) \mid (p, \mu) \in J\}]$. Therefore, p is a k-linear combination of elements in $\mathrm{Mon}(R^q) - [\{\mathrm{lm}(p) \mid (p, \mu) \in J\}]$.

(d) The only remaining detail to fill is to show that $\mathrm{NF}(p, J)$ is uniquely determined by the coset $p + \langle L \rangle \in R^q / \langle L \rangle$. But if $n_1, n_2 \in R^q$ are Janet-normal forms of the same coset $p + \langle L \rangle$, then $n_1 - n_2 \in \langle L \rangle$, and $n_1 - n_2$ is Janet-reduced modulo J because n_1 and n_2 are so. The same argument as in the last part of (b) shows that $n_1 - n_2 = 0$. □

An example for applying Janet's algorithm is given in the more general context of Ore algebras in Ex. 2.5.6. Janet's algorithm will also be used in the subsequent chapters. We finish this section with a few remarks about the connection of Janet bases to Gröbner bases and minimal Janet bases.

Remarks 2.2.14. (a) Every Janet basis is a Gröbner basis because the Janet-normal form of the S-polynomial [AL94] of each pair of elements of a Janet basis is zero and this Janet-normal form coincides with the normal form of the S-polynomial in the sense of Gröbner bases.

(b) For a submodule M of R^q which is generated by a set G of *monomials* it is sufficient to apply Decompose to G in order to obtain the Janet basis for M. In fact, Decompose returns the minimal Janet basis J for M, i.e. J is contained in every Janet basis (for the same order of the variables and monomial ordering on $\mathrm{Mon}(R^q)$) of M which consists of monomials only. More generally, a Janet basis J for an arbitrary submodule M of R^q is said to be minimal if $\{\mathrm{lm}(p) \mid (p, \mu) \in J\}$ is the minimal Janet basis of $\langle \mathrm{lm}(M) \rangle$ and the leading coefficient of p equals 1 for every $p \in J$. It is easily proved that a minimal Janet basis for M is unique. Because of auto-reduction Algorithm 2.2.12 (JanetBasis) returns the minimal Janet basis for $\langle L \rangle$ up to the condition on the leading coefficients.

2.3 Combinatorial Tools

For the study of modules over polynomial algebras (or Ore algebras as described later) several combinatorial objects are very useful. In this section we define only a few of them.

Definition 2.3.1. (a) For any set $S \subseteq \mathrm{Mon}(R^q)$ of monomials, the *generalized Hilbert series* of S is the formal power series

$$H_S(x_1, \ldots, x_n) := \sum_{m \in S} m \ \in \ \bigoplus_{i=1}^{q} \mathbb{Z}[[x_1, \ldots, x_n]]\, e_i.$$

(The Hilbert series usually encountered in commutative algebra is obtained from the generalized Hilbert series as $H_S(\lambda, \ldots, \lambda)$ for an indeterminate λ.)

(b) Let $S \subseteq \mathrm{Mon}(R^q)$ be a $\mathrm{Mon}(R)$-multiple closed set and consider a decomposition $T = \{ (m_1, \mu_1), \ldots, (m_l, \mu_l) \}$ of S into disjoint cones. Then the *Janet graph* of T is the labeled directed graph with vertex set $V = \{m_1, \ldots, m_l\}$

and with edge set E as follows: E contains an edge from m_i to m_j labeled by x_r if and only if $x_r \in \overline{\mu_i} = \{x_1, \ldots, x_n\} - \mu_i$ and $x_r \cdot m_i \in \mathrm{Mon}(\mu_j)m_j$, i.e., for each non-multiplicative variable x_r of a cone (m_i, μ_i) there is one edge labeled by x_r from the vertex m_i to the vertex m_j of the unique Janet-divisor of $x_r \cdot m_i$ in T.

The importance of the complement in $\mathrm{Mon}(R^q)$ of the $\mathrm{Mon}(R)$-multiple closed set generated by the leading monomials of the elements of a Janet basis is explained in the following remark.

Remark 2.3.2. Let M be a submodule of R^q and let J be a Janet basis for M with respect to some monomial ordering on $\mathrm{Mon}(R^q)$. We denote by S the $\mathrm{Mon}(R)$-multiple closed set generated by $\{\mathrm{lm}(p) \mid (p, \mu) \in J\}$. According to Theorem 2.2.13 (c), a k-basis of R^q/M is formed by the cosets in R^q/M represented by the monomials in $\bigcup_{(m,\mu)\in T} \mathrm{Mon}(\mu)m$, where T is a decomposition of $C := \mathrm{Mon}(R^q) - S$ into disjoint cones. Therefore, the generalized Hilbert series $H_C(x_1, \ldots, x_n)$ enumerates a k-basis of R^q/M.

The next remark shows that the computation of the generalized Hilbert series of a set S of monomials is trivial if a decomposition of S into disjoint cones is available.

Remark 2.3.3. Let $(C, \mu) \in \mathcal{P}(\mathrm{Mon}(R)) \times \mathcal{P}(\{x_1, \ldots, x_n\})$ be a cone. We use the geometric series

$$\frac{1}{1-x} = \sum_{i\geq 0} x^i$$

to write down the generalized Hilbert series $H_{[C]}(x_1, \ldots, x_n)$ as follows:

$$H_{[C]}(x_1, \ldots, x_n) = \frac{v(C)}{\prod_{x\in\mu}(1-x)}.$$

More generally, every decomposition of a $\mathrm{Mon}(R)$-multiple closed set S into disjoint cones allows to compute the generalized Hilbert series of S by adding the generalized Hilbert series of the cones. In an analogous way this remark applies to the complements of $\mathrm{Mon}(R)$-multiple closed sets.

Example 2.3.4. Let $R = k[x_1, x_2]$ and $G = \{x_1x_2^2,\, x_1^3x_2,\, x_1^4\}$ be the same generating set as in Ex. 2.1.14 and Ex. 2.1.18. Using the decomposition of $[G]$ into disjoint cones from (2.4), the generalized Hilbert series of $[G]$ is easily determined to be

$$H_{[G]}(x_1, x_2) = \frac{x_1^4}{(1-x_1)\,(1-x_2)} + \frac{x_1^3x_2}{1-x_2} + \frac{x_1^2x_2^2}{1-x_2} + \frac{x_1x_2^2}{1-x_2}.$$

The complement $C = \mathrm{Mon}(R) - [G]$ of $[G]$ was partitioned into cones in Ex. 2.1.18. From (2.5) we easily compute the generalized Hilbert series of C:

$$H_C(x_1, x_2) = \frac{1}{1-x_2} + x_1 + x_1x_2 + x_1^2 + x_1^2x_2 + x_1^3.$$

Note that

$$H_{[G]}(x_1, x_2) + H_C(x_1, x_2) = \frac{1}{(1 - x_1)\,(1 - x_2)}.$$

In the next example we visualize the Janet graphs for the results of Ex. 2.1.14 and Ex. 2.1.15.

Example 2.3.5. The Janet graph of the decomposition into disjoint cones (2.4) of the Mon($k[x_1, x_2]$)-multiple closed set generated by $\{x_1 x_2^2,\, x_1^3 x_2,\, x_1^4\}$ in Ex. 2.1.14 is:

$$x_1 x_2^2 \xrightarrow{x_1} x_1^2 x_2^2 \xrightarrow{x_1} x_1^3 x_2 \xrightarrow{x_1} x_1^4.$$

For the decomposition of the Mon($k[x_1, x_2, x_3]$)-multiple closed set generated by $G = \{x_1 x_2,\, x_2^3 x_3\}$ obtained in Ex. 2.1.15 we have the following Janet graph:

$$x_2^3 x_3 \xrightarrow{x_1} x_1 x_2^3 \xleftarrow{x_2} x_1 x_2^2 \xleftarrow{x_2} x_1 x_2.$$

We finish this section by proving a simple property of the Janet graph.

Proposition 2.3.6. *Let $S \subseteq \mathrm{Mon}(R^q)$ be $\mathrm{Mon}(R)$-multiple closed. The Janet graph of the decomposition of S into disjoint cones constructed by Algorithm 2.1.11 (Decompose) has no cycles.*

Proof. Without loss of generality we consider the case $q = 1$, i.e. a Mon(R)-multiple closed set S in Mon(R). Let $T = \{\,(m_1, \mu_1), \ldots, (m_l, \mu_l)\,\}$ be the decomposition of S into disjoint cones obtained by Alg. 2.1.11.

Let us consider an edge from $m_1 = x_1^{a_1} \cdots x_n^{a_n}$ to $m_2 = x_1^{b_1} \cdots x_n^{b_n}$ labeled by x_r in the Janet graph of T. Hence, x_r is a non-multiplicative variable for (m_1, μ_1), i.e. $x_r \notin \mu_1$.

If there is $1 \leq j < r$ such that $a_j > b_j$, then there is a minimal j with this property. For such an index j, the definition of the Janet division (see Def. 2.1.10) implies that $x_j \notin \mu_2$ which contradicts $x_r m_1 \in \mathrm{Mon}(\mu_2) m_2$.

If $a_i = b_i$ for all $i = 1, \ldots, r-1$, then $b_r > a_r$, again due to the definition of Janet division, because $x_r \notin \mu_1$ and $x_r \in \mu_2$.

Hence, in any case m_2 is greater than m_1 with respect to the lexicographical monomial ordering on Mon(R). A cycle in the Janet graph would contradict the defining properties of this monomial ordering. $\square$

Although the lexicographical monomial ordering was used in the previous proof, it should be clear from Section 2.1 that no monomial ordering on Mon(R^q) is needed to decompose Mon(R)-multiple closed sets (or their complements) into disjoint cones. The lexicographical monomial ordering arises naturally in this context because the process of constructing a decomposition can be viewed as an induction on the number of variables.

2.4 Ore Algebras

Ore algebras are certain algebras of linear operators. For instance, for a field k, the Weyl algebra $A_1(k)$ consists of the polynomials in the differential operator $\frac{d}{dt}$ whose coefficients are polynomials in t with coefficients in k. Many types of linear systems can be analyzed structurally by viewing them as modules over appropriate Ore algebras as explained in Section 4.1. The Ore algebra is chosen to contain all polynomials in the operators occurring in the equations that describe the system.

The construction of an Ore algebra can be thought of as an iteration of "Ore extensions" of a field or a polynomial ring or an Ore algebra itself. An "Ore extension" forms a skew polynomial ring in one indeterminate where the given ground field or ring provides the coefficients. After giving the definition of skew polynomial rings and Ore algebras following [CS98], several examples of Ore algebras are presented. In Section 4.1 the importance of these examples for the analysis of different types of linear systems will be discussed. At the end of this section crucial properties of Ore algebras are given.

In what follows, let k be a field and A a domain[2] which is also a k-algebra.

Definition 2.4.1. [MR00] The *skew polynomial ring* $A[\partial;\sigma,\delta]$ is the (not necessarily commutative) ring consisting of all polynomials in ∂ with coefficients in A obeying the commutation rule

$$\partial\, a = \sigma(a)\,\partial + \delta(a), \qquad a \in A,$$

where $\sigma : A \to A$ is a k-algebra endomorphism (i.e. σ is multiplicative, k-linear and satisfies $\sigma(1) = 1$) and $\delta : A \to A$ is a σ-derivation, i.e. δ is k-linear and satisfies:

$$\delta(a\,b) = \sigma(a)\,\delta(b) + \delta(a)\,b, \qquad a, b \in A.$$

Remark 2.4.2. If σ is injective, then $A[\partial;\sigma,\delta]$ is a domain because the degree in ∂ of a product of two non-zero elements of $A[\partial;\sigma,\delta]$ equals the sum of the degrees in ∂ of the factors. Then the construction of a skew polynomial ring can be iterated.

We recall the notion of Ore algebra as defined in [Chy98], [CS98].

Definition 2.4.3. Let $A = k$ or $A = k[x_1, \ldots, x_n]$, the commutative polynomial algebra over k. The *Ore algebra*

$$D = A[\partial_1;\sigma_1,\delta_1] \ldots [\partial_m;\sigma_m,\delta_m]$$

[2] A *domain* is a (not necessarily commutative) ring A with 1 which satisfies for all $a_1, a_2 \in A$ that $a_1 \neq 0$, $a_2 \neq 0$ implies $a_1 a_2 \neq 0$.

is the (not necessarily commutative) ring consisting of all polynomials in $\partial_1, \ldots, \partial_m$ with coefficients in A, where

$$\partial_i\,\partial_j = \partial_j\,\partial_i \qquad \text{for all} \quad 1 \leq i, j \leq m,$$

and all other commutation rules in D are defined by

$$\partial_i\, a = \sigma_i(a)\,\partial_i + \delta_i(a), \qquad a \in D, \qquad i = 1, \ldots, m, \tag{2.6}$$

where for $i = 1, \ldots, m$ the maps $\sigma_i : D \to D$ are k-algebra endomorphisms and $\delta_i : D \to D$ are σ_i-derivations (see Def. 2.4.1) satisfying

$$\begin{cases} \sigma_i \circ \sigma_j &= \sigma_j \circ \sigma_i, \\ \delta_i \circ \delta_j &= \delta_j \circ \delta_i, \\ \sigma_i \circ \delta_j &= \delta_j \circ \sigma_i, \\ \sigma_i(\partial_j) &= \partial_j, \\ \delta_i(\partial_j) &= 0 \end{cases} \qquad \text{for all} \quad 1 \leq i, j \leq m. \tag{2.7}$$

Remarks 2.4.4. Let $D = A[\partial_1; \sigma_1, \delta_1] \ldots [\partial_m; \sigma_m, \delta_m]$ be an Ore algebra.

(a) A straightforward calculation shows that the conditions (2.7) ensure that the commutation rules (2.6) are compatible with the postulation that ∂_i and ∂_j commute in D.

(b) If $\sigma_i = \mathrm{id}_D$ and $\delta_i = 0$ for all $i = 1, \ldots, m$, then D is the commutative polynomial algebra over k in $n + m$ indeterminates.

(c) Although the Ore algebra D in Def. 2.4.3 is defined “in one go”, it can be thought of as an iterated skew polynomial ring. However, formally it is more elaborate then to specify the maps σ_i, δ_i, $i = 1, \ldots, m$.

We list important examples of Ore algebras. They will be used for the study of linear systems in Chapter 5 and the subsequent chapters.

Examples 2.4.5. (a) For $n \in \mathbb{N}$, the *Weyl algebra* $A_n(k)$ over the field k is defined by

$$A_n(k) := k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_n; \sigma_n, \delta_n],$$

where for $i = 1, \ldots, n$, $\sigma_i = \mathrm{id}_{A_n(k)}$ and $\delta_i = \frac{\partial}{\partial x_i}$ is partial differentiation with respect to x_i. In the case $n = 1$ we usually write

$$A_1(k) := k[t][\partial; \sigma, \delta],$$

where $\sigma = \mathrm{id}_{A_1(k)}$ is the identity and $\delta = \frac{d}{dt}$ is formal differentiation with respect to t.

In $A_n(k)$ the commutation rules

$$\partial_i \, x_j = x_j \, \partial_i + \delta_{ij}, \quad 1 \leq i, j \leq n,$$

hold, where δ_{ij} is the Kronecker symbol. Hence, ∂_i represents the partial differential operator with respect to x_i (cf. also Ex. 4.1.1 (a)).

Another variant of the Weyl algebra is

$$B_n(k) := k(x_1, \ldots, x_n)[\partial_1; \sigma_1, \delta_1] \ldots [\partial_n; \sigma_n, \delta_n],$$

where $\sigma_i = \mathrm{id}_{B_n(k)}$ and $\delta_i = \frac{\partial}{\partial x_i}$, $i = 1, \ldots, n$, are defined in the same way as above, i.e. the *localized Weyl algebra* $B_n(k)$ is defined analogously to $A_n(k)$ but starting with the field $A = k(x_1, \ldots, x_n)$ of rational functions in x_1, $\ldots$, x_n.

(b) For $h \in \mathbb{R}$, we define the *algebra of shift operators of "length" h* by $S_h := \mathbb{R}[t][\delta_h; \sigma_h, \delta]$, where

$$\sigma_h(a) = a(t - h, \delta_h), \quad \delta(a) = 0, \qquad a \in S_h.$$

This implies the commutation rule

$$\delta_h \, t = (t - h) \, \delta_h$$

in S_h. Hence, δ_h represents the shift operator of length h (cf. also Ex. 4.1.1 (b)).

(c) Let $h \in \mathbb{R}$. We define $D_h := \mathbb{R}[t][\partial; \sigma_1, \delta_1][\delta_h; \sigma_2, \delta_2]$, where $\sigma_1 = \mathrm{id}_{D_h}$ is the identity, $\delta_1 = \frac{d}{dt}$ is formal differentiation with respect to t,

$$\sigma_2(a) = a(t - h, \partial), \qquad a \in D_h$$

and $\delta_2 = 0$. This algebra can be applied for differential time-delay systems (cf. Ex. 4.1.1 (c)).

(d) We define $D = k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_n; \sigma_n, \delta_n]$, where

$$\sigma_i(a) = a(x_1, \ldots, x_{i-1}, \ x_i - 1, \ x_{i+1}, \ldots, x_n, \partial_1, \ldots, \partial_n), \qquad a \in D,$$

and $\delta_i = 0$, $1 \leq i \leq n$. This algebra is suitable for the algebraic treatment of (multidimensional) discrete systems (cf. Ex. 4.1.1 (d)). Of course, the direction of the shifts can be reversed.

Convention 2.4.6. *When the commutation rules for an Ore algebra are clear from the context (e.g. for the Weyl algebras), then we also write*

$$k[x_1, \ldots, x_n][\partial_1, \ldots, \partial_m]$$

instead of

$$k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_m; \sigma_m, \delta_m],$$

which is, of course, a non-commutative ring in general.

We only recall the property of Ore algebras, studied by Oystein Ore [Ore33], which ensures the existence of left quotient division rings. (All concepts dealing with left multiplication, left ideals etc. can of course be translated into similar concepts for right multiplication, right ideals and so on.)

Definition 2.4.7. A ring D *satisfies the left Ore condition* if for all $a_1, a_2 \in D - \{0\}$ there exist $b_1, b_2 \in D - \{0\}$ such that $b_1 a_1 = b_2 a_2$.

Proposition 2.4.8. [MR00, Cor. 2.1.14] *A domain D has a left quotient division ring if and only if D satisfies the left Ore condition.*

In fact, if we confine ourselves to left Noetherian rings, then every domain has this property.

Proposition 2.4.9. [MR00, Thm. 2.1.15] *If D is a left Noetherian domain, then D satisfies the left Ore condition.*

Moreover, we have the following important proposition.

Proposition 2.4.10. [MR00, Thm. 1.2.9 (iv)] *If A is a left Noetherian domain and σ is an automorphism of A, then $A[\partial; \sigma, \delta]$ is also a left Noetherian domain.*

Note that all Ore algebras in Ex. 2.4.5 are left Noetherian.

Remark 2.4.11. If k is of characteristic zero, then the Weyl algebras $A_n(k)$ are simple rings, i.e., they have no two-sided ideals except for $\{0\}$ and $A_n(k)$. The same holds for $B_n(k)$. This property can be used when adapting the elementary divisor theory from commutative principal ideal domains to $B_1(k)$: for every matrix $A \in B_1(k)^{l \times m}$ there exist $r \in \mathbb{N}$ and matrices $P \in \mathrm{GL}(l, B_1(k))$ and $Q \in \mathrm{GL}(m, B_1(k))$ such that

$$P\,A\,Q = \mathrm{diag}(\underbrace{1, \ldots, 1}_{r}, \lambda, 0, \ldots, 0)$$

with a monic element $\lambda \in B_1(k)$ (Jacobson normal form, see [Reh02], [Coh85]).

2.5 Janet Bases for Ore Algebras

In this section, Janet's algorithm, as presented in the previous sections, is adapted to a certain class of Ore algebras. In contrast to the commutative case, the definition of the set of leading monomials of a left ideal in an Ore algebra needs the fixing of a normal form for polynomials. After defining monomial orderings and leading monomials for tuples of polynomials in Ore algebras, the adaptation of Janet's algorithm is explained. Then an example is treated in detail.

Let k be a field, $q \in \mathbb{N}$, and $D = k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_m; \sigma_m, \delta_m]$ an Ore algebra, where σ_i is an automorphism of $k[x_1, \ldots, x_n]$ for all $i = 1, \ldots, m$.

By Proposition 2.4.10, D is left Noetherian. In order to be able to base Janet's algorithm for Ore algebras on the notion of multiple-closed sets of monomials as treated before in the commutative case, we restrict to the class of Ore algebras D for which the commutation rules have the form

$$\left\{\begin{array}{rcll} \sigma_i(x_j) & = & c_{ij}x_j + d_{ij}, & c_{ij} \in k - \{0\}, \quad d_{ij} \in k, \\ \deg(\delta_i(x_j)) & \leq & 1, & 1 \leq i \leq m, \quad 1 \leq j \leq n. \end{array}\right. \tag{2.8}$$

Throughout the rest of this chapter, we denote by D^q the free left D-module of tuples of length q with entries in D. Let $(e_1, \ldots, e_q)$ be the standard basis of D^q.

Definition 2.5.1. We define the set of *monomials* of D by

$$\mathrm{Mon}(D) := \{x^a\,\partial^b \mid a \in (\mathbb{Z}_{\geq 0})^n, b \in (\mathbb{Z}_{\geq 0})^m\},\ x^a := x_1^{a_1} \cdots x_n^{a_n},\ \partial^b := \partial_1^{b_1} \cdots \partial_m^{b_m}.$$

We set $\mathrm{Mon}(D^q) := \bigcup_{i=1}^q \mathrm{Mon}(D)e_i$.

Remark 2.5.2. $\mathrm{Mon}(D^q)$ is a k-basis of D^q, i.e. every element $p \in D^q$ has a unique representation

$$p = \sum_{i=1}^{q} \sum_{\substack{a\in(\mathbb{Z}_{\geq 0})^n \\ b\in(\mathbb{Z}_{\geq 0})^m}} c_{i,a,b}\, x^a\, \partial^b\, e_i \tag{2.9}$$

as linear combination of the elements of $\mathrm{Mon}(D^q)$ with coefficients $c_{i,a,b} \in k$, where only finitely many $c_{i,a,b}$ are non-zero. Since x_i and ∂_j do not commute in general, by the previous definition of monomials we distinguish a *normal form* (2.9) for the elements $p \in D^q$.

If D is non-commutative, it is not reasonable to generalize the notion of $\mathrm{Mon}(R)$-multiple closed set of monomials from Section 2.1 to "left $\mathrm{Mon}(D)$-multiple closed set" of monomials because left multiples of elements in $\mathrm{Mon}(D^q)$ are not elements in $\mathrm{Mon}(D^q)$ in general: e.g. in the Weyl algebra $A_1(k)$ with commutation rule $\partial\, t = t\,\partial + 1$, the left multiple $\partial\, t$ of t is not a monomial. However, the type of commutation rules implied by (2.8) allows to use a bijection between the non-commutative monomials and commutative monomials which is compatible with left multiplication.

Remark 2.5.3. Let $R = k[\tilde{x}_1, \ldots, \tilde{x}_n, \tilde{\partial}_1, \ldots, \tilde{\partial}_m]$ be the commutative polynomial algebra in $n + m$ indeterminates over the field k. By using the bijection

$$\widetilde{\ } : \mathrm{Mon}(D^q) \to \mathrm{Mon}(R^q)\ :\ x^a\,\partial^b\, e_i \mapsto \tilde{x}^a\,\tilde{\partial}^b\, e_i,$$

$a \in (\mathbb{Z}_{\geq 0})^n$, $b \in (\mathbb{Z}_{\geq 0})^m$, $1 \leq i \leq q$, the sets $\mathrm{Mon}(D^q)$ and $\mathrm{Mon}(R^q)$ can be identified and we have $\widetilde{m_1\, m_2} = \widetilde{m_1}\,\widetilde{m_2}$ for all $m_1, m_2 \in \mathrm{Mon}(D^q)$.

Since the bijection defined in the previous remark is compatible with left multiplication, we use it to define monomial orderings on $\mathrm{Mon}(D^q)$.

Definition 2.5.4. Let R and $\widetilde{\ }$ be defined as in Remark 2.5.3.

(a) A *monomial ordering* on $\mathrm{Mon}(D^q)$ is a total ordering $<$ on $\mathrm{Mon}(D^q)$ for which there exists a monomial ordering $\widetilde{<}$ on $\mathrm{Mon}(R^q)$ such that

$$m_1 < m_2 \qquad \Longleftrightarrow \qquad \widetilde{m_1} \mathrel{\widetilde{<}} \widetilde{m_2} \qquad \text{for all } m_1, m_2 \in \mathrm{Mon}(D^q).$$

(b) Fix a monomial ordering $<$ on $\mathrm{Mon}(D^q)$. Let $p \in D^q - \{0\}$ and consider its unique representation (2.9) with respect to the k-basis $\mathrm{Mon}(D^q)$. The *leading monomial* $\mathrm{lm}(p)$ of p is defined as the $<$-greatest element $x^a\,\partial^b\,e_i$ in $\mathrm{Mon}(D^q)$ for which $c_{i,a,b} \neq 0$. Then the *leading coefficient* $\mathrm{lc}(p)$ of p is defined by $c_{i,a,b}$. Hence, we have maps $\mathrm{lm} : D^q - \{0\} \to \mathrm{Mon}(D^q)$ and $\mathrm{lc} : D^q - \{0\} \to k$.

In what follows, we fix a monomial ordering $<$ on $\mathrm{Mon}(D^q)$.

We are going to discuss now the adaptation of Janet's algorithm to Ore algebras. We succeed in doing so by keeping Algorithm 2.2.12 (JanetBasis) literally and explaining the differences caused by the non-commutativity of D.

Remark 2.5.5. Let M be a left submodule of D^q and let G be a finite generating set for M. The set $\mathrm{lm}(G)$ is utilized again to steer the course of Janet's algorithm. In Algorithm 2.2.12 (JanetBasis) it generated a $\mathrm{Mon}(R)$-multiple closed set. Here we apply the bijection $\widetilde{\ }$ of Remark 2.5.3 in order to consider $\mathrm{lm}(G)$ as a set of monomials in a commutative polynomial ring in $n+m$ variables. The procedure to decompose $[\mathrm{lm}(G)]$ into disjoint cones is applied as described in Remark 2.2.5 to the module generators in G by respecting that the module generators are multiplied by monomials only from the left. Such multiplications were already written down as left multiplications in Algorithm 2.1.11 (Decompose). For the case of commutative polynomial rings, this was not essential, but in the present context this restriction needs to be maintained.

The notion of Janet-reducibility and the process of Janet-reduction are adopted to the present situation by restricting all multiplications of monomials in $\mathrm{Mon}(D)$ by arbitrary elements in D^q to left multiplications as was already indicated in Definition 2.2.8 and Algorithm 2.2.9 (Janet-reduce) and Algorithm 2.2.7 (Auto-reduce). Both algorithms form a left multiple of an element of D^q in step 4 and subtract it from a given tuple in D^q in order to cancel its leading term. Because of the form of the commutation rules of D implied by (2.8) cancellation of the leading term is achieved similarly to the commutative case. Since $<$ is a well-ordering, Janet-reduction terminates.

Again the non-multiplicative variables of the cones guide Janet's algorithm to new module generators. Since Janet's algorithm for D again produces an ascending sequence of sets of leading monomials (see part (a) in the proof of Theorem 2.2.13: $[\mathrm{lm}(G)] \subsetneq [\mathrm{lm}(G')]$), Corollary 2.1.5 shows the termination. The correctness is proved in the same way as in the proof of Theorem 2.2.13.

Let us demonstrate Janet's algorithm on a simple example.

Example 2.5.6. Let us investigate the following system of linear ordinary differential equations (with a shift) for an unknown differentiable function y:

$$\begin{cases} t\,\dot{y}(t) - 2\,t\,y(t) & = & 0, \\ y(t+1) - y(t) & = & 0. \end{cases} \tag{2.10}$$

We are going to represent the operators which are applied to the unknown function y in the above equations by polynomials in the Ore algebra

$$D := \mathbb{R}[t][\partial; \sigma_1, \delta_1][\delta; \sigma_2, \delta_2]$$

from Ex. 2.4.5 (c) with $h = -1$; i.e., ∂ represents the differential operator with respect to t and δ the shift operator of "length" -1 in t. Hence, in D we have the commutation rules

$$\begin{array}{rcl} \partial\,t & = & t\,\partial + 1, \\ \delta\,t & = & (t+1)\,\delta, \\ \partial\,\delta & = & \delta\,\partial. \end{array}$$

Let us first discuss the kind of solutions of (2.10) which are admissible. Since the space of functions among which solutions of (2.10) are to be determined is considered as a left module over D, these functions need to be infinitely often differentiable. So for instance, smooth functions, analytic functions or distributions are admissible.

For $t \neq 0$, we have the ordinary differential equation $\dot{y}(t) = 2\,y(t)$ whose smooth solutions are $c\,e^{2t}$, $c \in \mathbb{R}$. But the second equation in (2.10) implies $c = 0$, so that the zero function is the only smooth solution of (2.10). If we are not only interested in smooth solutions, then we suppress the division by t, and the first equation in (2.10) also admits the Heaviside distribution. Again, this solution is ruled out by the second equation in (2.10). Let us demonstrate that Janet's algorithm relieves us of this latter consideration.

As monomial ordering on $\mathrm{Mon}(D)$ we choose the degree reverse lexicographical ordering (cf. Ex. 2.2.2 (a)) which is uniquely determined by

$$\partial > \delta > t.$$

In what follows, the sets of leading monomials (translated to sets of commutative monomials via the bijection defined in Remark 2.5.3) will be decomposed into disjoint cones with respect to $y_1 = \partial$, $y_2 = \delta$, $y_3 = t$.

Let us consider the left ideal in D generated by $G_1 := \{p_1, p_2\}$, where

$$p_1 := t\,\partial - 2\,t, \qquad p_2 := \delta - 1 \in D.$$

The leading monomials of p_1 and p_2 are the following:

$$\mathrm{lm}(p_1) = t\,\partial, \qquad \mathrm{lm}(p_2) = \delta.$$

Hence, G_1 is already auto-reduced. By decomposing $[\mathrm{lm}(G_1)]$ into disjoint cones we find the complete set

$$J_1 := \{\, (p_1, \{\partial, t\}),\ (p_2, \{\delta, t\}),\ (p_3, \{\partial, \delta, t\})\,\}, \tag{2.11}$$

where $p_3 := \partial \cdot p_2$. The variable δ is non-multiplicative for the first cone, whereas the variable ∂ is non-multiplicative for the second cone. Therefore, we compute the Janet-normal forms of $\delta \cdot p_1$ and $\partial \cdot p_2$. The latter polynomial reduces to zero simply by subtracting p_3. For the former polynomial, Janet-reduction is performed as follows:

$$p_4 := \delta \cdot p_1 = (t+1)\,\delta\,\partial - 2\,(t+1)\,\delta.$$

Then p_4 has leading monomial $t\,\delta\,\partial$. The unique Janet-divisor is the third cone in (2.11). Hence, we compute

$$p_5 := p_4 - (t+1)\,(\delta\,\partial - \partial) = (t+1)\,\partial - 2\,(t+1)\,\delta.$$

We have $\mathrm{lm}(p_5) = t\,\partial$ with unique Janet-divisor $(p_1, \{\partial, t\})$. Then

$$p_6 := p_5 - (t\,\partial - 2\,t) = \partial + 2\,t - 2\,(t+1)\,\delta$$

is Janet-reducible with Janet-divisor $(p_2, \{\delta, t\})$ which finally yields

$$r := p_6 + 2\,(t+1)\,(\delta - 1) = \partial - 2 = \mathrm{NF}(\delta \cdot p_1, J_1).$$

The passivity check therefore yields $P = \{r\} \neq \{0\}$, so that we have to auto-reduce $G_2 := \{p_1, p_2, p_3, r\}$ in the next round. This removes p_1 and p_3 from G_2. Now Decompose($\{\delta - 1,\ \partial - 2\}$, $\{\partial, \delta, t\}$) returns the complete set

$$J_2 = \{\, (p_2, \{\delta, t\}),\ (r, \{\partial, \delta, t\})\,\},$$

so that only the Janet-normal form of $\partial \cdot p_2$ needs to be computed. But it is easily checked that $\mathrm{NF}(\partial \cdot p_2, J_2) = 0$. Therefore, J_2 is a Janet basis.

Translating these polynomials back to equations, we therefore have the following differential time-delay system which is formally equivalent to (2.10):

$$\left\{ \begin{array}{rcl} \dot{y}(t) - 2\,y(t) & = & 0, \\[2ex] y(t+1) - y(t) & = & 0. \end{array} \right.$$

The way how a Janet basis determines certain kinds of solutions of a given system of linear equations is discussed in more detail in Chapter 4.

2.6 Implementations

Some implementations of Janet's algorithm (realized at Lehrstuhl B für Mathematik, RWTH Aachen) are discussed in this section.

The Maple packages `Involutive` [BCG+03a] and `Janet` [BCG+03b] implement the involutive basis algorithm [GB98a, GB98b] of V. P. Gerdt and Y. A. Blinkov for commutative polynomial algebras respectively for certain differential rings (see Section 3.3). These packages were developed initially by C. F. Cid in 2000, continued by the author of this thesis from 2001 on. Meanwhile, both implementations have been adjusted to more recent versions of the involutive basis algorithm by Gerdt and Blinkov. In particular, many unnecessary Janet-reductions are avoided by remembering multiplications by non-multiplicative variables which have already been considered in previous passivity checks and by involutive analogues of Buchberger's criteria for Gröbner basis computations [AH05], [GY05]. `Involutive` and `Janet` also contain some combinatorial tools, e.g. the computation of generalized Hilbert series, Hilbert polynomials, Cartan characters etc., and several valuable procedures for commutative algebra and differential algebra.

`JanetOre` is a Maple package which extends `Involutive` to Ore algebras as explained in Section 2.5. The polynomials given to and resulting from `JanetOre` are always to be understood as the normal forms (2.9) on page 32. Since the order of the factors in a product displayed by Maple can change in different runs of the same commands, the user may have to sort the variables in each term of the result according to the normal form (2.9) in order to write down the actual polynomial.

A specialized implementation of the involutive basis technique for linear difference ideals [Ger06] has recently been realized in the Maple package `LDA` (abbreviation for Linear Difference Algebra) [GR06]. In particular, this implementation has been applied for the generation of finite difference schemes for partial differential equations [GBM06].

2.7 Reducing the Complexity of Janet Basis Computations

In this section a simple preprocessing tool for the computation of Janet bases is presented. It reduces, if possible, the number of variables that occur in a given generating set for a polynomial ideal. More precisely, this procedure produces a generating set for the intersection of the given ideal with a polynomial ring in a subset of the variables. This is achieved by solving some generating equations for variables which occur only linearly and by substituting the resulting expressions for these variables into the other equations. By remembering the equations that were used to reduce the number of variables, the original system of algebraic equations can be solved. Since the computation time of Janet bases usually depends doubly exponentially on the number of variables, the presented procedure quite often allows to compute Janet bases of ideals that otherwise could not be found by using naively the existing implementations of Janet's algorithm. After describing the algorithm for this preprocessing tool, the method is demonstrated on two examples.

Remark 2.7.1. The problem of computing Gröbner bases of ideals in (commutative) polynomial algebras is in general not efficiently solvable in the sense of complexity theory. Computing a reduced Gröbner basis [AL94] of an ideal of polynomials over the rational numbers was proved to be an EXPSPACE-complete problem. Some upper bounds on the total degrees of the polynomials of a reduced Gröbner basis are known. If the total degrees of the n-variate polynomials in a generating set for an ideal I are bounded by d, then the total degrees of the polynomials in a reduced Gröbner basis of I are bounded by a polynomial in d which is, however, doubly exponentially in n (for more details see [vzGG03], [May97]). Since every Janet basis is a Gröbner basis and a Janet basis can be constructed from a Gröbner basis G just by turning G into a complete set (see Def. 2.2.8 (a)), similar remarks hold for Janet basis computations.

It happens quite often that problems which seem to be intractable using the current Maple implementations of the Janet basis algorithms can be handled successfully by eliminating some of the variables which are present in the given equations. By the previous remark the complexity of computing a Janet basis is then reduced drastically.

The simplest situation is that in a system of algebraic equations one variable occurs only linearly with constant coefficient in some equation. Then such an equation can be used to eliminate the variable in all other equations. This case is exploited in the following algorithm, whose termination and correctness are clear. For simplicity we confine ourselves to a commutative polynomial algebra R over a field k, but the method can also be applied for Ore algebras. We set $x := \{x_1, \ldots, x_n\}$.

Algorithm 2.7.2 (Substitute)**.**

Input: A finite set $L_1 \subset R$, where $R = k[x]$

Output: (L_2, E, v), where $L_2 \subset k[v]$ is a finite set, $v \subseteq x$, and E is a finite set of equations $y = r$, where $y \in x - v$ and $r \in k[x - \{y\}]$, such that

$$(p = 0 \quad \forall p \in L_1) \qquad \Longleftrightarrow \qquad \begin{cases} q = 0 & \forall q \in L_2, \\ y = r & \forall (y = r) \in E. \end{cases}$$

Algorithm:

1: $L_2 \leftarrow L_1$
2: $E \leftarrow \emptyset$
3: $v \leftarrow x$
4: **while** $\exists\, y \in v,\ p \in L_2,\ 0 \neq c \in k,\ \tilde{p} \in k[v - \{y\}] : p = cy + \tilde{p}$ **do**
5: $\quad r \leftarrow (p - \tilde{p})/c$
6: $\quad L_2 \leftarrow L_2 - \{p\}$ and substitute r for y in every $q \in L_2$
7: $\quad E \leftarrow E \cup \{y = r\}$
8: $\quad v \leftarrow v - \{y\}$
9: **od**
10: **return** (L_2, E, v)

Remark 2.7.3. The result of Algorithm 2.7.2 of course depends on the choices in step 4. Keeping expressions small (in particular the total degree of the polynomials) is a first heuristic.

Algorithm 2.7.2 is implemented in `Involutive`.

Example 2.7.4.

```
> with(Involutive):
```

We consider the polynomial ring $R = \mathbb{Q}[x, y, z]$ and the ideal of R which is generated by the elements of L defined below.

```
> var := [x,y,z];
```

$$var := [x,\ y,\ z]$$

```
> L := [x*y-z, z-x*y+y-1, x^2-y^2];
```

$$L := [x\,y - z,\ z - x\,y + y - 1,\ x^2 - y^2]$$

Instead of computing a Janet basis of this ideal in the polynomial ring with three variables, we apply `Substitute` to L in order to reduce, if possible, the number of indeterminates involved in the above polynomials:

```
> Substitute(L, var);
```

$$[[x^2 - 1],\ [z = x\,y,\ y = 1],\ [x]]$$

In fact the first equation $xy - z = 0$ can be solved for z so that z can be eliminated in the equation $z - xy + y - 1 = 0$. Then we obtain $y = 1$ which can be substituted into $x^2 - y^2 = 0$. Therefore, one is left with only one equation $x^2 - 1 = 0$ involving only one variable which is very easy to solve.

A more realistic example is presented next. It emerges in the context of constructing matrix representations of groups and was dealt with in the preparation of [PR06].

Example 2.7.5. In order to find projective matrix representations of degree d over $\mathbb{C}$ of the group

$$G_{2,3,7} = \langle a, b \mid a^2,\, b^3,\, (ab)^7 \rangle$$

the images of a and b under such a representation are first written down as matrices A and B with indeterminate entries. The relators a^2, b^3, $(ab)^7$ of the above presentation are translated into relations for commutative polynomials obtained from the entries of the matrix equations

$$A^2 = c_1\, I_d, \quad B^3 = c_2\, I_d, \quad (AB)^7 = c_3\, I_d,$$

where $c_1, c_2, c_3 \in \mathbb{C} - \{0\}$ are fixed and I_d is the $(d \times d)$-identity matrix. In this example we give only a few details about the case $d = 6$, $c_1 = -1$, $c_2 = 1$, $c_3 = 1$.

```
> with(Involutive):
> with(LinearAlgebra):
```

We need to define a primitive seventh root of unity in Maple:

```
> alias(omega=RootOf(add(x^i, i=0..6)));
```

$$\omega$$

```
> simplify(omega^7);
```

$$1$$

The conjugation action of $\mathrm{GL}(6, \mathbb{C})$ is exploited to normalize the matrices A and B in the following way (for more details see [PR06]):

```
> A := Matrix(6, 6, [[0,omega,0,0,c[1],d[1]],
> [-omega^6,0,0,0,c[2],d[2]],
> [0,0,0,-1,c[3],d[3]],
> [0,0,1,0,c[4],d[4]],
> [0,0,0,0,a[1],a[2]],
> [0,0,0,0,a[3],-a[1]]]);
```

$$A := \begin{bmatrix} 0 & \omega & 0 & 0 & c_1 & d_1 \\ -\omega^6 & 0 & 0 & 0 & c_2 & d_2 \\ 0 & 0 & 0 & -1 & c_3 & d_3 \\ 0 & 0 & 1 & 0 & c_4 & d_4 \\ 0 & 0 & 0 & 0 & a_1 & a_2 \\ 0 & 0 & 0 & 0 & a_3 & -a_1 \end{bmatrix}$$

```
> B := SubMatrix(Matrix(6, 6)+1, 1..6, [2,3,1,5,6,4]);
```

$$B := \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

In particular, we already have $B^3 = I_6$.

The following indeterminates occur in the matrix A. We are going to derive a system of algebraic equations for them.

```
>  v := [c[1],d[1],c[2],d[2],c[3],d[3],c[4],d[4],a[1],a[2],a[3]];
```

$$v := [c_1,\, d_1,\, c_2,\, d_2,\, c_3,\, d_3,\, c_4,\, d_4,\, a_1,\, a_2,\, a_3]$$

The first condition for these indeterminates to satisfy comes from the matrix equation $A^2 + I_6 = 0$. The left hand sides of these equations are listed in $L1$:

```
>  L1 := evala(map(op, convert(A^2+1, listlist)));
```

$$\begin{aligned}&L1 := [0,\, 0,\, 0,\, 0,\, \omega\, c_2 + c_1\, a_1 + d_1\, a_3,\, \omega\, d_2 + c_1\, a_2 - d_1\, a_1,\, 0,\, 0,\, 0,\, 0,\\ &c_1 + \omega\, c_1 + \omega^2\, c_1 + c_1\, \omega^3 + c_1\, \omega^4 + c_1\, \omega^5 + c_2\, a_1 + d_2\, a_3,\\ &d_1 + \omega\, d_1 + \omega^2\, d_1 + \omega^3\, d_1 + d_1\, \omega^4 + d_1\, \omega^5 + c_2\, a_2 - d_2\, a_1,\, 0,\, 0,\, 0,\, 0,\\ &-c_4 + c_3\, a_1 + d_3\, a_3,\, -d_4 + c_3\, a_2 - d_3\, a_1,\, 0,\, 0,\, 0,\, 0,\, c_3 + c_4\, a_1 + d_4\, a_3,\\ &d_3 + c_4\, a_2 - d_4\, a_1,\, 0,\, 0,\, 0,\, 0,\, {a_1}^2 + a_2\, a_3 + 1,\, 0,\, 0,\, 0,\, 0,\, 0,\, 0,\, {a_1}^2 + a_2\, a_3 + 1]\end{aligned}$$

Instead of repeating this method for $(AB)^7 = I_6$, which would produce quite large expressions, we prescribe the characteristic polynomial of the matrix AB to be $\lambda^6 + \lambda^5 + \ldots + \lambda + 1$.

```
>  chi := CharacteristicPolynomial(A.B, lambda);
```

$$\begin{aligned}&\chi := \lambda^6 + (-a_2 - c_4 - \omega)\, \lambda^5 + (-d_4\, a_1 + c_4\, a_2 + a_2\, \omega - c_2 + c_4\, \omega)\, \lambda^4 +\\ &(d_4\, a_1\, \omega - d_2\, a_1 + c_2\, a_2 - a_2\, c_4\, \omega - c_3 - c_3\, \omega - c_3\, \omega^2 - c_3\, \omega^3 - c_3\, \omega^4 - c_3\, \omega^5 +\\ &\omega\, c_2)\lambda^3 + \omega(c_3 - c_2\, a_2 + d_2\, a_1 + c_3\, \omega + c_3\, \omega^2 + c_3\, \omega^3 + c_3\, \omega^4 + c_3\, \omega^5 - a_3\, \omega^5 +\\ &a_1\, d_3\, \omega^5 - c_3\, a_2\, \omega^5)\lambda^2 + (-a_3\, a_2\, \omega + a_3 - d_3\, a_1 - a_2\, a_3 + c_3\, a_2 - {a_1}^2 - {a_1}^2\, \omega -\\ &{a_1}^2\, \omega^2 - {a_1}^2\, \omega^3 - {a_1}^2\, \omega^4 - {a_1}^2\, \omega^5 - a_3\, a_2\, \omega^2 - a_3\, a_2\, \omega^3 - a_3\, a_2\, \omega^4 - a_3\, a_2\, \omega^5)\lambda\\ &- a_2\, a_3 - {a_1}^2\end{aligned}$$

Hence, all coefficients of χ are equated with 1, resulting in the following list of left hand sides of algebraic equations for the indeterminates:

```
>  L2 := map(i->coeff(chi, lambda, i)-1, [$0..5]);
```

$$\begin{aligned}&L2 := [-a_2\, a_3 - {a_1}^2 - 1, -a_3\, a_2\, \omega + a_3 - d_3\, a_1 - a_2\, a_3 + c_3\, a_2 - {a_1}^2 - {a_1}^2\, \omega -\\ &{a_1}^2\, \omega^2 - {a_1}^2\, \omega^3 - {a_1}^2\, \omega^4 - {a_1}^2\, \omega^5 - a_3\, a_2\, \omega^2 - a_3\, a_2\, \omega^3 - a_3\, a_2\, \omega^4 - a_3\, a_2\, \omega^5 - 1,\\ &\omega(c_3 - c_2\, a_2 + d_2\, a_1 + c_3\, \omega + c_3\, \omega^2 + c_3\, \omega^3 + c_3\, \omega^4 + c_3\, \omega^5 - a_3\, \omega^5 + a_1\, d_3\, \omega^5 -\\ &c_3\, a_2\, \omega^5) - 1, -d_2\, a_1 + d_4\, a_1\, \omega + c_2\, a_2 - a_2\, c_4\, \omega - c_3 - c_3\, \omega - c_3\, \omega^2 - c_3\, \omega^3 -\\ &c_3\, \omega^4 - c_3\, \omega^5 + \omega\, c_2 - 1,\, -d_4\, a_1 + c_4\, a_2 + a_2\, \omega - c_2 + c_4\, \omega - 1,\, -a_2 - c_4 - \omega - 1]\end{aligned}$$

Both lists of left hand sides of algebraic equations are united in the list L:

```
> L := [op(L1),op(L2)]:
```

Applying the present implementation of Janet's algorithm (for the commutative polynomial case) in Maple directly to this generating set for an ideal in $\mathbb{Q}(\omega)[c_1, d_1, c_2, d_2, c_3, d_3, c_4, d_4, a_1, a_2, a_3]$ leads to a rather long run:

```
> st := time():
> IB := InvolutiveBasis(L, v):
> time()-st;
```

$$860.470$$

The computation takes about 14 minutes on a computer equipped with two Pentium III processors (1 GHz, 2 GB memory) running Maple 9.5. The Hilbert series of the residue class ring is obtained essentially without further computation:

```
> PolHilbertSeries(lambda);
```

$$1 + 7\,\lambda + 17\,\lambda^2 + 27\,\lambda^3 + 37\,\lambda^4 + 47\,\lambda^5 + \lambda^6\,(\frac{47}{1-\lambda} + \frac{10}{(1-\lambda)^2})$$

From the degree of the last denominator in the Hilbert series, we conclude that the Krull dimension of the residue class ring is two which corresponds to a two-dimensional variety of representations for $G_{2,3,7}$.

A faster method to discuss the above system of algebraic equations is to apply `Substitute` to L first.

```
> st := time():
> S := Substitute(L, v):
> time()-st;
```

$$0.539$$

In less than one second, the procedure `Substitute` chose the following equations:

```
> for i in S[2] do print(i); od;
```

$$c_2 = (1 + \omega + \omega^2 + \omega^3 + \omega^4 + \omega^5)\,(c_1\,a_1 + d_1\,a_3)$$
$$d_2 = (1 + \omega + \omega^2 + \omega^3 + \omega^4 + \omega^5)\,(c_1\,a_2 - d_1\,a_1)$$
$$c_4 = c_3\,a_1 + d_3\,a_3$$
$$d_4 = -d_3\,a_1 + c_3\,a_2$$
$$c_3 = \omega + \omega\,c_1\,a_1 + \omega\,d_1\,a_3 + a_2\,d_1\,a_3 + d_1\,{a_1}^2 + a_2\,\omega^2\,d_3\,a_3 + {a_1}^2\,\omega^2\,d_3$$

These equations were used to reduce the number of variables in the algebraic equations. Now we are left with six variables:

```
> S[3];
```

$$[c_1,\, d_1,\, d_3,\, a_1,\, a_2,\, a_3]$$

If the solutions for the variables in $S[3]$ are found, then the values of the variables in the left hand sides of $S[2]$ are determined. The computation of the Janet basis of the ideal generated by $S[1]$ is much faster than the previous Janet basis computation:

```
> st := time():
> IB := InvolutiveBasis(S[1], S[3]):
> time()-st;
```

$$151.940$$

Hence, the Janet basis is computed in 2.5 minutes. Of course, since we consider a different polynomial ring than before, the Hilbert series of the residue class ring differs from the one above. But we still can read off the dimension of the variety in the last denominator:

```
> PolHilbertSeries(lambda);
```

$$1 + 6\,\lambda + 15\,\lambda^2 + 24\,\lambda^3 + 33\,\lambda^4 + \lambda^5\,(\frac{33}{1-\lambda} + \frac{9}{(1-\lambda)^2})$$

Chapter 3

Symbolic Computation with Differential Equations

When computing with differential equations in a formal way, one is immediately led to view all derivatives of the unknown functions as "variables" which are differentially independent. The differential equations under consideration then define relations among these variables. This approach is nowadays known as *jet calculus*.

In this chapter we describe relevant issues for the symbolic treatment of differential equations. The methods we are going to present are used in Chapter 5 for the structural analysis of linear systems. The jet formalism mentioned above is explained in the first section. Next we introduce the notion of *general linearization* of partial differential equations in this formal approach. It is described using a minimum of notions of differential geometry and differential algebra (in particular, we will not give the details about the vertical bundle for which we refer to [Pom94]). In Section 3.3 we show that every system of (nonlinear) partial differential equations defines a differential ring. Necessary arrangements for an implementation of Janet's algorithm to deal with the general linearization of a nonlinear system of PDEs are discussed.

The calculus explained here is described in a more general and rigorous way in [Bar01a], where it is part of a formal approach to functional spaces. Moreover, the Maple package `jets` [Bar01b] makes available many tools for dealing with jet expressions and implements many valuable procedures for formal differential geometry.

3.1 The Jet Formalism

The way of viewing a differential equation as a relation for *jet variables* which represent the derivatives of the unknown functions and which are differentially independent a priori enables symbolic manipulations of differential equations.

In this section we describe this formal approach which is known as jet calculus and which we will apply in the next section to linearize differential equations in a generic way. Good references for the introduction to the jet formalism are [Olv93], [Pom94].

First we recall the definition of a fibered manifold. Then the jet bundle of a fibered manifold is defined. After giving an example of how the coordinate changes of a jet bundle can be derived, we comment on the connection of the coordinates of a jet bundle, which are also called jet variables, to the variables occurring in a system of differential equations.

In what follows, all manifolds are understood to be smooth real manifolds. For fixed $n, m \in \mathbb{N}$ we denote by $\mathrm{pr}_n : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ the projection onto the first n components.

Definition 3.1.1 (Fibered manifold)**.** Let X be a manifold of dimension n, E a manifold of dimension $m + n$ and $\pi : E \longrightarrow X$ a surjective submersion, i.e., π is surjective and $d\pi$ has rank n at all points of E. Then E is a *fibered manifold over X (with projection π)*, if for every $p \in E$ there exist a neighborhood U of p, a neighborhood V of $\pi(U)$ and local coordinates $\Theta : U \longrightarrow \mathbb{R}^n \times \mathbb{R}^m$ of E and $\theta : V \longrightarrow \mathbb{R}^n$ of X such that

$$\theta \circ \pi|_U = \mathrm{pr}_n \circ \Theta,$$

i.e., the following diagram of smooth maps commutes:

$$\begin{tikzcd} U \arrow[r, "\Theta"] \arrow[d, "\pi|_U"'] & \mathbb{R}^n \times \mathbb{R}^m \arrow[d, "\mathrm{pr}_n"] \\ V \arrow[r, "\theta"] & \mathbb{R}^n \end{tikzcd}$$

For $x \in X$, $\pi^{-1}(\{x\})$ is called the *fiber over* x.

Two fibered manifolds E_1 and E_2 over X with projections π_1 resp. π_2 are said to be *equivalent*, if there exists a diffeomorphism $\Phi : E_1 \longrightarrow E_2$ such that the diagram

$$\begin{tikzcd} E_1 \arrow[rr, "\Phi"] \arrow[dr, "\pi_1"'] & & E_2 \arrow[dl, "\pi_2"] \\ & X & \end{tikzcd}$$

commutes.

Before discussing the admissible coordinate changes of a fibered manifold we give some well-known examples of fibered manifolds.

Examples 3.1.2. (a) If X and Y are manifolds of dimensions n resp. m, then $E := X \times Y$ is a trivial fibered manifold over X with projection $X \times Y \to X : (x, y) \mapsto x$.

(b) Let X be a manifold of dimension n. Then the tangent bundle TX and the cotangent bundle T^*X are fibered manifolds over X with projections $TX \to X : (p, v) \mapsto p$ resp. $T^*X \to X : (p, \lambda) \mapsto p$.

(c) The Möbius band is a fibered manifold over the circle S^1. For each $x \in S^1$, the fiber over x is homeomorphic to the interval $(0, 1) \subset \mathbb{R}$.

Remark 3.1.3. Let E be a fibered manifold over X with projection π and let $(x, u) : U \to \mathbb{R}^n \times \mathbb{R}^m$ be local coordinates for E. Then $\pi : E \to X$ is represented locally by $(x, u) \mapsto x$ with respect to these coordinates. This implies that only the smooth maps having the following property are admissible *coordinate changes* of E. Let $\Theta_i : U_i \to \mathbb{R}^n \times \mathbb{R}^m$, $i = 1, 2$, be local coordinates for E, where $U_1 \cap U_2 \neq \emptyset$. Then

$$\Psi = \Theta_2 \circ \Theta_1^{-1} : \Theta_1(U_1 \cap U_2) \to \Theta_2(U_1 \cap U_2)$$

is a coordinate change of E if and only if Ψ is of the form

$$(\overline{x}, \overline{u}) = \Psi(x, u) = (\varphi(x), \psi(x, u)),$$

where the new coordinates are denoted with a bar.

Definition 3.1.4. Let E be a fibered manifold over X with projection π and let U be an open subset of X. A smooth map $\zeta : U \to E$ is called a *local section* of E, if $\pi \circ \zeta = \mathrm{id}_U$. If $U = X$, then ζ is called a *global section* of E.

Examples 3.1.5. (a) Let X and Y be manifolds of dimensions n resp. m. Then the local sections of $E := X \times Y$ can be identified with the smooth maps $U \to Y$, where U is an open subset of X. In particular, the set of global sections of E can be viewed as the set of smooth maps $X \to Y$.

(b) Let X be a manifold of dimension n. Then the local sections of the tangent bundle TX are (smooth) vector fields, and the local sections of the cotangent bundle T^*X are (smooth) differential 1-forms.

Definition 3.1.6 (Jet bundle)**.** Let X be a manifold of dimension n, E a fibered manifold over X of dimension $m + n$, and let $q \in \mathbb{Z}_{\geq 0}$. The *$q$-th jet bundle $J_q(E)$ of E* is the fibered manifold over X of dimension $n + m\binom{q+n}{n}$ defined by local coordinates

$$(x^1, \ldots, x^n, u^k_J \mid 1 \leq k \leq m, J \in (\mathbb{Z}_{\geq 0})^n, |J| = J_1 + \ldots + J_n \leq q)$$

and the coordinate changes derived from the coordinate changes

$$\overline{x} = \varphi(x), \quad \overline{u} = \psi(x, u)$$

of E by identifying u^k_J with the partial derivative $\frac{\partial^{|J|} f^k}{\partial x^J}$, $1 \leq k \leq m$, $J \in (\mathbb{Z}_{\geq 0})^n$, $|J| \leq q$, of an arbitrary local section ζ of E and by using the chain rule for differentiation (cf. Ex. 3.1.11 below for $q = 1$).

Remark 3.1.7. Local coordinates for the q-th jet bundle of E are used to represent local sections ζ of E and their (partial) derivatives up to order q. By definition the symbols u^k_J among the jet coordinates have both upper and lower indices. The former enumerate the component of a local section ζ of E which is represented by u, the latter indicate the variables with respect to which this local section is differentiated. For instance, if E is a four-dimensional fibered manifold over a two-dimensional manifold X, then $J_2(E)$ has coordinates

$$x^1, x^2, u^1_{(0,0)}, u^2_{(0,0)}, u^1_{(1,0)}, u^1_{(0,1)}, u^2_{(1,0)}, u^2_{(0,1)}, u^1_{(2,0)}, u^1_{(1,1)}, u^1_{(0,2)}, u^2_{(2,0)}, u^2_{(1,1)}, u^2_{(0,2)}.$$

Note that the jet coordinates are differentially independent (e.g. $\partial u^1_{(1,0)}/\partial x^1 = 0$). Therefore, all jet coordinates x^i, u^k_J are also called *jet variables*.

Definition 3.1.8. Let X be a manifold, E a fibered manifold over X, $q \in \mathbb{Z}_{\geq 0}$.

(a) The *order* of the jet variable u^k_J of $J_q(E)$ is defined to be $|J|$, whereas the *order* of the jet variable x^i is defined to be zero.

(b) Any smooth real-valued function $f = f(x^i, u^k_J)$ which depends on finitely many jet variables of $J_q(E)$ is called a *jet expression* over E. We denote by $A(E)$ the vector space of all jet expressions over E (depending on finitely many jet variables of any order).

Example 3.1.9. Let X have dimension 2 and E have dimension $2 + 1$. Then

$$f(x^i, u^k_J) = x^1 \sin u^1_{(2,0)} + e^{x^2 + u^1_{(0,1)}} + (u^1_{(0,0)})^3$$

is a jet expression over E. It depends on the jet variables x^i, u^k_J up to order 2.

Convention 3.1.10. *If a derivative with respect to a single variable x^j is represented by a jet variable, then we also write u^k_j instead of $u^k_{(j)}$. Moreover, u^k is used as a synonym for $u^k_{(0,\ldots,0)}$.*

In what follows, each expression containing jet variables is understood to be a sum over all indices which occur at the same time as upper and lower indices of jet variables.

Example 3.1.11. The coordinate changes for the first jet bundle $J_1(E)$ are derived as follows. Denoting new coordinates with a bar, a coordinate change

$$\overline{x} = \varphi(x), \quad \overline{u} = \psi(x, u)$$

of E implies for a local section ζ of E that

$$\overline{\zeta}(\overline{x}) = \overline{\zeta}(\varphi(x)) = \psi(x, \zeta(x)),$$

where $\overline{\zeta}$ represents the local section ζ in the new coordinates. Differentiating this equation with respect to x^i and using the chain rule we obtain

$$\frac{\partial \overline{\zeta}^k(\varphi(x))}{\partial x^i} = \frac{\partial \overline{\zeta}^k}{\partial \overline{x}^j}|_{\varphi(x)} \frac{\partial \varphi^j}{\partial x^i} = \frac{\partial \psi^k}{\partial x^i}|_{(x,\zeta(x))} + \frac{\partial \psi^k}{\partial u^j}|_{(x,\zeta(x))} \frac{\partial \zeta^j}{\partial x^i} = \frac{\partial \psi^k(x,\zeta(x))}{\partial x^i}.$$

The middle of the previous sequence of equations is translated into a coordinate change of the jet variables as follows:

$$\overline{u}^k_j \frac{\partial \varphi^j}{\partial x^i} = \frac{\partial \psi^k}{\partial x^i} + \frac{\partial \psi^k}{\partial u^j} u^j_i.$$

Locally this equation can be solved for $\overline{u}^k_j$ or for u^j_i.

In a similar way, but involving larger and larger expressions, the coordinate changes for the higher jet variables can be derived. The Maple package `jets` [Bar01b] allows to compute coordinate changes for jet variables automatically.

Remark 3.1.12. In a system of partial differential equations which contain the unknown functions $u^1, \ldots, u^q$ and finitely many of their derivatives with respect to the variables $x^1, \ldots, x^n$ (of any order), $u^1, \ldots, u^q$ are called the *dependent variables* and $x^1, \ldots, x^n$ are called the *independent variables*. In the context of jet calculus the independent and dependent variables are identified with the jet coordinates $x^i : E \to \mathbb{R}$, $i = 1, \ldots, n$, resp. $u^k : E \to \mathbb{R}$, $k = 1, \ldots, q$. Hence, the jet variables represent the independent variables and the derivatives of the dependent variables up to a certain order.

Using the jet formalism, a system of partial differential equations of order q can be written as

$$\left\{ \begin{array}{rcl} f^1(x^i, u^k_J) & = & 0, \\ & \vdots & \\ f^r(x^i, u^k_J) & = & 0, \end{array} \right.$$

where the jet variables u^k_J represent the $|J|$-th order derivatives of the k-th component of the unknown tuple of functions.

We finish this section by defining the total derivative of a jet expression.

Definition 3.1.13. Let X be a manifold of dimension n, E a fibered manifold over X of dimension $m + n$ and $f \in A(E)$ a jet expression over E. Then the *i-th total derivative of f* is defined by

$$D_i\, f := \frac{\partial f}{\partial x^i} + u^k_{J+1_i} \frac{\partial f}{\partial u^k_J}, \tag{3.1}$$

where $1_i \in (\mathbb{Z}_{\geq 0})^n$ is the multi-index whose only non-zero entry is the i-th entry which equals 1. Note that, according to Convention 3.1.10, the last term in (3.1) is a double sum over k and J. The definition of total derivative is extended to tuples of jet expressions $f \in A(E)^r$ in the component-wise way.

For any $J = (j_1, \ldots, j_n) \in (\mathbb{Z}_{\geq 0})^n$ and any jet expression f over E we set

$$D_J\, f := D_1^{j_1} \cdots D_n^{j_n}\, f.$$

Example 3.1.14. In the situation of Def. 3.1.13 we have for any jet variable u_J^k, $1 \leq k \leq m$, $J \in (\mathbb{Z}_{\geq 0})^n$,

$$D_K\, u_J^k = u_{J+K}^k, \qquad K \in (\mathbb{Z}_{\geq 0})^n.$$

More interestingly, let $n = 2$, $m = 1$, and consider the jet expression $f = x^1\, u + (u_{(1,0)})^2$. Then the first total derivative of f is

$$D_1\, f = u + x^1\, u_{(1,0)} + 2\, u_{(1,0)}\, u_{(2,0)}.$$

3.2 Linearization of Differential Equations

As is well-known, among differential equations the linear ones are best understood and most easily handled. Linear differential equations in no way exhibit the variety of phenomena which arise for nonlinear differential equations, but it is a good technique in general to study a linearization of a given nonlinear differential equation to gain insight in local properties of the latter. In this section we present a way of linearizing differential equations which is generic, i.e. no solution (e.g. critical point) of the differential equation needs to be known in advance, and which is adequate for symbolic computation. This *general linearization* is applied in Sections 5.3 and 5.4 for the structural analysis of linear systems.

In this section we first explain how the linearization of an ordinary differential equation along a given trajectory can be understood in a functional analytic framework using the Fréchet derivative. Following [Olv93] we then define the formal Fréchet derivative for (tuples of) jet expressions which leads to the notion of general linearization of a jet expression. This is applied to differential equations, and an example demonstrates the differences between the general linearization and linearizations along certain solutions.

We start by recalling the definition of the Fréchet derivative.

Definition 3.2.1. [Aub77], [Die69] Let V and W be (real) Banach spaces, $x_0 \in V$, $N \subseteq V$ a neighborhood of x_0 and $f : N \to W$ a map. If there exists a continuous linear map $Df(x_0) : V \to W$ such that

$$\lim_{\|x - x_0\| \to 0} \frac{\|f(x) - f(x_0) - Df(x_0)(x - x_0)\|}{\|x - x_0\|} = 0,$$

then f is said to be *differentiable at* x_0 and $Df(x_0)$ is called the *Fréchet derivative of* f *at* x_0. Let $\mathcal{L}(V,W)$ the set of continuous linear maps from V to W. If the map $V \to \mathcal{L}(V,W) : x \mapsto Df(x)$ is differentiable at x_0, then f is said to be *twice differentiable at* x_0.

Remark 3.2.2. [Aub77], [AMR88] By definition of the Fréchet derivative,

$$x \mapsto f(x_0) + Df(x_0)\,(x - x_0)$$

is tangent to f at x_0, i.e. it is a first-order approximation of f at x_0.

Remark 3.2.3. Let us investigate an ordinary differential equation

$$\dot{u} = f(u) \tag{3.2}$$

for a function $u = u(t) \in B$, where B is a Banach space with norm $\|\cdot\|$ consisting of certain differentiable real-valued functions defined on an open subset of $\mathbb{R}$. In order to define a linearization of this differential equation, we assume that f is a map $B \to B$ which is twice differentiable at every $u_0 \in B$. Let $u_{\mathrm{ref}} \in B$ be a solution of (3.2). Then we want to determine a function $0 \neq U \in B$ such that

$$u_{\mathrm{ref}} + \varepsilon\, U \tag{3.3}$$

approximates a solution of (3.2) in B for small $\varepsilon > 0$.

To this end let $0 \in I \subseteq \mathbb{R}$ be an open interval and define the map

$$F : I \longrightarrow B : \varepsilon \longmapsto (t \mapsto f(u_{\mathrm{ref}}(t) + \varepsilon\, U(t))).$$

We are going to show that F is twice differentiable at every $x_0 \in I$.

By definition of the Fréchet derivative of f at $u_{\mathrm{ref}} + x_0\, U$, $Df(u_{\mathrm{ref}} + x_0\, U)$ is a continuous linear map $B \to B$. The following computation proves that $DF(x_0)$ exists and equals $\Delta(x_0) := Df(u_{\mathrm{ref}} + x_0\, U)\, U$. We have:

$$\begin{aligned}
& \lim_{|x-x_0|\to 0} \frac{\|F(x) - F(x_0) - \Delta(x_0)\,(x - x_0)\|}{|x - x_0|} \\
= & \lim_{|x-x_0|\to 0} \frac{\|f(u_{\mathrm{ref}} + x\, U) - f(u_{\mathrm{ref}} + x_0\, U) - \Delta(x_0)\,(x - x_0)\|}{|x - x_0|} \\
= & \lim_{|x-x_0|\to 0} \frac{\|f(u_{\mathrm{ref}} + x\, U) - f(u_{\mathrm{ref}} + x_0\, U) - \Delta(x_0)\,(x - x_0)\|}{\|U \cdot (x - x_0)\|} \cdot \frac{\|U \cdot (x - x_0)\|}{|x - x_0|} \\
= & \lim_{|x-x_0|\to 0} \frac{\|f(u_{\mathrm{ref}} + x\, U) - f(u_{\mathrm{ref}} + x_0\, U) - \Delta(x_0)\,(x - x_0)\|}{\|U \cdot (x - x_0)\|} \cdot \|U\| \\
= & \ 0
\end{aligned}$$

because $Df(u_{\mathrm{ref}} + x_0\, U)$ is the Fréchet derivative of f at $u_{\mathrm{ref}} + x_0\, U$. Therefore, F is differentiable at every $x_0 \in I$ with Fréchet derivative $DF(x_0) = \Delta(x_0)$.

We continue by showing that $DF : x \mapsto DF(x)$ is differentiable at every $x_0 \in I$. The candidate for the Fréchet derivative of DF at x_0 is now

$$\Delta^{(2)}(x_0) := D^2 f(u_{\text{ref}} + x_0\, U)(U)(U).$$

By applying the usual inequality for the operator norm we find:

$$\begin{aligned}
&\frac{\|DF(x) - DF(x_0) - \Delta^{(2)}(x_0)\,(x - x_0)\|}{|x - x_0|} \\
= \ &\frac{\|Df(u_{\text{ref}} + x\,U)\,U - Df(u_{\text{ref}} + x_0\,U)\,U - \Delta^{(2)}(x_0)\,(x - x_0)\|}{|x - x_0|} \\
\leq \ &\frac{\|Df(u_{\text{ref}} + x\,U) - Df(u_{\text{ref}} + x_0\,U) - D^2 f(u_{\text{ref}} + x_0\,U)(U)(\,\cdot\,)\cdot(x - x_0)\|}{\|U \cdot (x - x_0)\|} \cdot \\
&\|U\| \cdot \frac{\|U \cdot (x - x_0)\|}{|x - x_0|} \\
= \ &\frac{\|Df(u_{\text{ref}} + x\,U) - Df(u_{\text{ref}} + x_0\,U) - D^2 f(u_{\text{ref}} + x_0\,U)(U)(\,\cdot\,)\cdot(x - x_0)\|}{\|U \cdot (x - x_0)\|} \cdot \\
&\|U\|^2.
\end{aligned}$$

Since $D^2 f(u_{\text{ref}} + x_0\,U)$ is the Fréchet derivative of Df at $u_{\text{ref}} + x_0\,U$, we have

$$\lim_{|x - x_0| \to 0} \frac{\|DF(x) - DF(x_0) - \Delta^{(2)}(x_0)\,(x - x_0)\|}{|x - x_0|} = 0.$$

We conclude that F is twice differentiable at every $x_0 \in I$.

Hence, F has a Taylor expansion around 0 [Die69]:

$$F(\varepsilon) = F(0) + DF(0)\,\varepsilon + O(\varepsilon^2) = f(u_{\text{ref}}) + Df(u_{\text{ref}})\,U\,\varepsilon + O(\varepsilon^2),$$

where $O(\varepsilon^2)$ denotes terms which are of higher order in ε than 1. By substituting (3.3) for u in (3.2) we obtain:

$$\dot{\overline{u_{\text{ref}} + \varepsilon\,U}} = f(u_{\text{ref}} + \varepsilon\,U) = f(u_{\text{ref}}) + Df(u_{\text{ref}})\,U\,\varepsilon + O(\varepsilon^2),$$

i.e.

$$\dot{u}_{\text{ref}} + \varepsilon\,\dot{U} = f(u_{\text{ref}}) + Df(u_{\text{ref}})\,U\,\varepsilon + O(\varepsilon^2).$$

Since u_{ref} is a solution of (3.2), $\dot{u}_{\text{ref}}$ and $f(u_{\text{ref}})$ cancel in the previous equation. By neglecting the terms of higher order in ε we therefore have to solve the linear ordinary differential equation

$$\dot{U} = Df(u_{\text{ref}})\,U \tag{3.4}$$

for U. The equation (3.4) is the *linearization of* (3.2) *along the trajectory* u_{ref}.

In many applications u_{ref} is taken to be constant, if possible, i.e. a critical point of (3.2). Having solved the linear differential equation (3.4), $u_{\text{ref}} + \varepsilon\,U$ approximates a solution of (3.2) in B for small ε.

Example 3.2.4. Let us consider

$$\dot{u} = u^2. \tag{3.5}$$

This ordinary differential equation has the following solutions:

$$v_\infty(t) = 0, \qquad v_c(t) = \frac{1}{c-t}, \quad c \in \mathbb{R}.$$

We fix a compact interval $J \subset \mathbb{R}$ and define $B := C^1(J)$. Then B is a Banach space with norm

$$\|g\| := \sup_{x\in J} |g(x)| + \sup_{x\in J} |g'(x)|, \qquad g \in C^1(J),$$

and B contains the restrictions to J of the solutions v_∞ and v_c, $c \notin J$.

We define $f : B \to B : u \mapsto u^2$ which is twice differentiable at every $x_0 \in B$. The Fréchet derivative of f at $x_0 \in B$ is

$$Df(x_0) : B \to B : u \mapsto 2\, x_0\, u.$$

For every solution $u_{\mathrm{ref}} \in B$ of (3.5) we have the linearization of (3.5) along the trajectory u_{ref}:

$$\dot{U} = Df(u_{\mathrm{ref}})\, U.$$

The linearization along v_∞ yields the zero solution. More interesting is the linearization along the trajectory v_c, $c \in \mathbb{R}$:

$$\dot{U} = 2\, v_c\, U = \frac{2}{c-t}\, U. \tag{3.6}$$

The general solution of (3.6) is

$$U_{\tilde{c}}(t) = \frac{\tilde{c}}{(c-t)^2}, \qquad \tilde{c} \in \mathbb{R}.$$

Let us fix $c \in \mathbb{R} - J$. For an ε-neighborhood of c in $\mathbb{R} - J$ we compare $v_c + \varepsilon\, U_{\tilde{c}}$ for $\tilde{c} = -1$ with the solution $v_{c+\varepsilon}$ of (3.5). In fact, we have

$$\begin{aligned} v_{c+\varepsilon}(t) &= \frac{1}{(c+\varepsilon)-t} \\ &= \frac{1}{c-t} + \frac{-1}{(c-t)^2}\,\varepsilon + O(\varepsilon^2) \\ &= v_c(t) + \varepsilon\, U_{-1}(t) + O(\varepsilon^2). \end{aligned}$$

Hence, up to terms of higher order in ε, $v_{c+\varepsilon}$ coincides with $v_c + \varepsilon\, U_{-1}$.

In order to arrive at a formal approach to linearize partial differential equations, we will no longer consider the right hand side of a differential equation to be defined by a map from a certain Banach space of functions to itself but to be a jet expression. More precisely, an ordinary differential equation $\dot{u} - f(u) = 0$ for one function of t as above will be viewed as $g(t, u_J) = 0$ for an appropriate jet expression g. First, however, we recall the directional derivatives of a function.

Remark 3.2.5. [AMR88] Let V and W be Banach spaces, $x_0 \in V$, $N \subseteq V$ a neighborhood of x_0 and $f : N \to W$ a map. If f is differentiable at x_0, then the directional derivatives

$$\frac{d}{d\varepsilon} f(x_0 + \varepsilon\, v)|_{\varepsilon=0}$$

of f at x_0 exist for all $v \in V$ and equal $Df(x_0)\, v$.

The previous remark motivates the definition of the formal Fréchet derivative of a jet expression. Note that it is differentiated only in the "directions" of the dependent variables and their derivatives.

Definition 3.2.6. [Olv93] Let X be a manifold of dimension n and E a fibered manifold over X of dimension $m+n$. Let $p = p(x^i, u^k_J) \in A(E)^r$ be a tuple of jet expressions over E. The *formal Fréchet derivative of* p is the differential operator D_p which is defined for any $f = f(x^i, u^k_J) \in A(E)^m$ by

$$D_p(f) := \frac{d}{d\varepsilon}|_{\varepsilon=0}\, p(x^i, D_J\,(u + \varepsilon\, f))$$

(i.e. before differentiating with respect to ε, every jet variable u^k_J on which p depends is substituted by the total derivative $D_J(u^k + \varepsilon\, f^k)$).

Remark 3.2.7. In the definition of the formal Fréchet derivative we have

$$\begin{aligned} & \frac{d}{d\varepsilon}\, p(x^i, D_J\,(u + \varepsilon\, f)) \\ = \quad & \frac{\partial p}{\partial u^l_K}(x^i, D_J\,(u + \varepsilon\, f)) \cdot \frac{d}{d\varepsilon}\, D_K\,(u^l + \varepsilon\, f^l) \\ = \quad & \frac{\partial p}{\partial u^l_K}(x^i, D_J\,(u + \varepsilon\, f))\, D_K\, f^l, \end{aligned}$$

which is a double sum over $l = 1, \ldots, m$ and $K \in (\mathbb{Z}_{\geq 0})^n$. Hence,

$$\begin{aligned} D_p(f) \quad &= \quad \frac{d}{d\varepsilon}|_{\varepsilon=0}\, p(x^i, D_J\,(u + \varepsilon\, f)) \\ &= \quad \frac{\partial p}{\partial u^l_K}(x^i, D_J u^k)\, D_K\, f^l \\ &= \quad \frac{\partial p}{\partial u^l_K}(x^i, u^k_J)\, D_K\, f^l. \end{aligned}$$

By considering the total derivatives $D_K f^l$ of the "directions" f^l as new dependent variables U^l_K we arrive at the central notion of this section.

Definition 3.2.8. Let X be a manifold of dimension n, E a fibered manifold over X of dimension $m+n$, $q \in \mathbb{Z}_{\geq 0}$, and $J_q(E)$ the q-th jet bundle over E. For any jet expression $f = f(x^i, u^k_J)$ over E depending on finitely many jet variables up to order q we define the *general linearization* of f to be

$$F(x^i, u^k_J, U^l_K) := \frac{\partial f}{\partial u^k_J}\, U^k_J, \tag{3.7}$$

$$i = 1, \ldots, n, \quad J,\, K \in (\mathbb{Z}_{\geq 0})^n, \quad k,\, l = 1, \ldots, q,$$

where the U^l_K are new indeterminates (i.e. all x^i, u^k_J, U^l_K are differentially independent).

Remarks 3.2.9. (a) Following Convention 3.1.10, the general linearization F of f is defined as a double sum over the index k enumerating the components of a section and the index J representing the partial differentiations. In general, for a jet expression f depending on jet variables of order at most q, the linearization F is a real-valued function of jet variables x^i, u^k_J up to order q and the new variables U^l_K. It depends on the latter variables only linearly.

(b) Of course, the notion of general linearization is extended in a componentwise way to tuples of jet expressions which we encounter as left hand sides of systems of differential equations.

(c) With a higher amount of differential geometric language this kind of linearization can be developed using the notion of *vertical bundle* of a fibered manifold. The variables x^i, u^k_J, U^l_K in Def. 3.2.8 can be taken as the coordinates of the vertical bundle of the fibered manifold $J_q(E)$ over X, i.e. the subbundle of the tangent bundle of $J_q(E)$ ignoring the TX-components. For more details we refer to [Pom94].

Remark 3.2.10. Let

$$\left\{ \begin{array}{rcl} f^1(x^i, u^k_J) & = & 0, \\ & \vdots & \\ f^r(x^i, u^k_J) & = & 0 \end{array} \right. \tag{3.8}$$

be a (nonlinear) system of partial differential equations of order q with independent variables $x^1, \ldots, x^n$ and dependent variables $u^1, \ldots, u^m$. We consider the system of linear partial differential equations

$$\left\{ \begin{array}{rcl} F^1(x^i, u^k_J, U^l_K) & = & 0, \\ & \vdots & \\ F^r(x^i, u^k_J, U^l_K) & = & 0, \end{array} \right.$$

where F^i is the general linearization of f^i in the sense of Def. 3.2.8 and call it the *general linearization* of (3.8). Let $v = (v^1, \dots, v^m)$ be a solution of (3.8), i.e.

$$f^j(x^i, \frac{\partial^{|J|} v^k}{\partial x^J}) = 0 \qquad \text{for all} \quad 1 \le j \le r.$$

Then

$$\begin{cases} F^1(x^i, \frac{\partial^{|J|} v^k}{\partial x^J}, U^l_K) & = & 0, \\ & \vdots & \\ F^r(x^i, \frac{\partial^{|J|} v^k}{\partial x^J}, U^l_K) & = & 0 \end{cases}$$

is the linearization of (3.8) along the solution v (trajectory v in the case of ordinary differential equations).

Example 3.2.11. Let us revisit Ex. 3.2.4 and apply jet notation. The ordinary differential equation under consideration is $\dot{u}^1 = (u^1)^2$. Since only one independent and one dependent variable are involved, i.e. $n = m = 1$, we set $x = x^1$, $u = u^1$. Then the differential equation reads

$$\dot{u} = u^2. \tag{3.9}$$

Using the jet formalism, the equation is represented by

$$g(x, u_J) = 0, \qquad \text{where} \quad g(x, u_J) := u_1 - u^2.$$

The linearization of (3.9) is

$$G(x, u_J, U_J) = 0,$$

where $G(x, u_J, U_J)$ is the general linearization of $g(x, u_J)$:

$$G(x, u_J, U_J) = \frac{\partial g}{\partial u_1} U_1 + \frac{\partial g}{\partial u} U = U_1 - 2\, u\, U.$$

By substituting a solution $u_{2,c}$, $c \in \mathbb{R}$, of (3.9) (see Ex. 3.2.4) for the jet variable u in this general linearization we obtain the linearization (3.6) along the trajectory $u_{2,c}$.

Remark 3.2.12. For the algebraic treatment of differential equations according to our intentions we want the space of functions which contains solutions of the given equations to be domain. Therefore, we consider analytic functions f in the following example, i.e. $f \in C^\omega(\mathbb{R})$.

The next example demonstrates that computations with the general linearization of a nonlinear system of differential equations are not well-defined in general, when substituting a solution of the nonlinear equations into the coefficients of the general linearization. It deals with a system of partial differential equations with independent variables x, y, z and dependent variable u, we write

$$u_{\underbrace{x,\ldots,x}_{i},\underbrace{y,\ldots,y}_{j},\underbrace{z,\ldots,z}_{k}}$$

for the jet variable $u_{(i,j,k)}$. This notation will also be used later on in other examples.

Example 3.2.13. Let us consider the following system of nonlinear partial differential equations:

$$\left\{\begin{array}{rcl} u_x &=& u_y\, u_z, \\ u_y &=& u_x\, u_z. \end{array}\right. \tag{3.10}$$

We substitute the right hand side of the first equation for u_x in the second equation and obtain

$$u_y\,(1-u_z^2)=0.$$

If $u_y=0$, then $u_x=0$ due to (3.10). Therefore

$$u^{(1)}(x,y,z) := F(z) \qquad \text{with} \quad F\in C^\omega(\mathbb{R}) \quad \text{arbitrary}$$

are solutions of (3.10). Otherwise, $u_z=1$ or $u_z=-1$. In the first case, (3.10) implies $u_y=u_x$ which yields the solutions

$$u^{(2)}(x,y,z) := z+G(x+y) \qquad \text{with} \quad G\in C^\omega(\mathbb{R}) \quad \text{arbitrary.}$$

In the second case, we have $u_y=-u_x$ resulting in the solutions

$$u^{(3)}(x,y,z) := -z+H(x-y) \qquad \text{with} \quad H\in C^\omega(\mathbb{R}) \quad \text{arbitrary.}$$

The general linearization of (3.10) is

$$\left\{\begin{array}{rcl} U_x - u_z\,U_y - u_y\,U_z &=& 0, \\ U_y - u_z\,U_x - u_x\,U_z &=& 0. \end{array}\right. \tag{3.11}$$

According to Remark 3.2.10, when we substitute $u^{(1)}$ for u in (3.11), we obtain the linearization of (3.10) along the solution $u^{(1)}$:

$$\left\{\begin{array}{rcl} U_x(x,y,z) - \dot{F}(z)\,U_y(x,y,z) &=& 0, \\ U_y(x,y,z) - \dot{F}(z)\,U_x(x,y,z) &=& 0. \end{array}\right. \tag{3.12}$$

Analogously, the linearizations of (3.10) along the solutions $u^{(2)}$ and $u^{(3)}$ are

$$\left\{ \begin{array}{rcl} U_x(x,y,z) - U_y(x,y,z) - \dot{G}(x+y)\,U_z(x,y,z) & = & 0, \\ -U_x(x,y,z) + U_y(x,y,z) - \dot{G}(x+y)\,U_z(x,y,z) & = & 0 \end{array} \right. \tag{3.13}$$

resp.

$$\left\{ \begin{array}{rcl} U_x(x,y,z) + U_y(x,y,z) + \dot{H}(x-y)\,U_z(x,y,z) & = & 0, \\ U_x(x,y,z) + U_y(x,y,z) - \dot{H}(x-y)\,U_z(x,y,z) & = & 0. \end{array} \right. \tag{3.14}$$

A Janet basis of the general linearization (3.11) is obtained roughly as follows. Since the leading terms of the equations in (3.11) are U_x resp. $-u_z\,U_x$, the first equation in (3.11) is multiplied by u_z and added to the second equation. The resulting differential equation E has leading term $(1-u_z^2)\,U_y$, and the process of Janet separation defines x to be a non-multiplicative variable for this equation. The term $-u_z\,U_y$ in the first equation can be eliminated using this new equation, but this requires to divide by

$$d_1 := 1 - u_z^2 \tag{3.15}$$

first. Since we deal with the general linearization, this operation can be performed, but for linearizations along certain solutions v this division may be undefined (namely when $v_z^2 = 1$). When working with the general linearization we assume that this is not the case.

Then the new equation E is differentiated with respect to x because x is a non-multiplicative variable for this equation. By using the previous equations the result is Janet-reduced to

$$U_z = 0.$$

In a certain step the process of Janet-reduction divides by

$$\begin{aligned} d_2 := {} & -2\,u_x\,u_{y,z}\,{u_z}^2 + u_{y,y}\,{u_z}^2 + 2\,u_y\,u_{x,z}\,{u_z}^2 - u_{x,x}\,{u_z}^2 - 2\,u_y\,u_z\,u_{y,z} + \\ & 2\,u_{x,z}\,u_z\,u_x + {u_x}^2\,u_{z,z} - u_{y,y} - {u_y}^2\,u_{z,z} + u_{x,x}. \end{aligned} \tag{3.16}$$

Since the left hand sides of all remaining equations are linear combinations of U_x, U_y, and U_z, it is clear that the Janet basis of (3.11) is given by

$$U_x = 0, \qquad U_y = 0, \qquad U_z = 0. \tag{3.17}$$

Now we examine the two expressions (3.15), (3.16) by which Janet's algorithm divides. Substituting the solutions $u^{(1)}$, $u^{(2)}$, $u^{(3)}$ for u into these expressions, we find generically

	$u^{(1)}$	$u^{(2)}$	$u^{(3)}$
d_1	$\neq 0$	$= 0$	$= 0$
d_2	$= 0$	$= 0$	$= 0$

We have $d_1 \neq 0$ if and only if $1 + F'(z)^2 \neq 0$.

Hence, the first step of Janet's algorithm described above is valid for the linearization (3.12) along $u^{(1)}$, but not for the linearizations (3.13) and (3.14) along $u^{(2)}$ and $u^{(3)}$.

In fact, for these specific linearizations the Janet bases are easily obtained as

$$U_x = 0, \qquad U_y = 0 \qquad\qquad (\text{along } u^{(1)}),$$

where the algorithm has to divide by

$$-1 + F'(z)^2,$$

resp.

$$U_x - U_y = 0, \qquad U_z = 0 \qquad\qquad (\text{along } u^{(2)}),$$

where

$$G'(x + y) \neq 0$$

must be assumed, resp.

$$U_x + U_y = 0, \qquad U_z = 0 \qquad\qquad (\text{along } u^{(3)}),$$

where

$$H'(x - y) \neq 0$$

is necessary.

We conclude that a Janet basis of a general linearization of a nonlinear system of differential equations can only be specialized to certain linearizations along trajectories of the nonlinear system, if division by zero during the Janet basis computation can be excluded. For instance, the equation $U_z = 0$, as part of the Janet basis (3.17) of the general linearization, does not hold in general for the linearization along $u^{(1)}$. Similarly, $U_x = 0$ does not hold in general for the linearization along $u^{(2)}$ or $u^{(3)}$. In the next section the question by which coefficients Janet's algorithm may divide is answered.

3.3 Janet Bases for Linear Differential Equations with Non-constant Coefficients

As shown in the last example of the previous section, computations with linear differential equations having non-constant coefficients need to obey relations for these coefficients, if this linear system is given as the general linearization of a nonlinear system. The coefficients of the linearized equations are subject to the original nonlinear equations. An implementation of Janet's algorithm to deal with such equations needs to take these nonlinear equations into account in order to prevent division by zero. In this section we explain this situation using the notion of differential ring. We are going to present a simple strategy to respect the relations for the coefficients of a general linearization.

Definition 3.3.1. [Kol73] Let R be a ring. A map $\delta : R \to R$ is a *derivation of* R if it satisfies

$$\begin{aligned} \delta(a+b) &= \delta(a) + \delta(b), \\ \delta(a\,b) &= \delta(a)\,b + a\,\delta(b), \qquad a, b \in R. \end{aligned}$$

A *differential ring* is a ring R with a finite set of commuting derivations of R. Accordingly, a *differential field* is a differential ring which is also a field. A differential ring which is also an algebra over some ring or field is a *differential algebra*. An ideal I of a differential ring R is called a *differential ideal* if it is closed under the action of the derivations of the differential ring R.

Examples 3.3.2. (a) Every ring is a differential ring endowed with a trivial derivation.

(b) Let $R = \mathbb{Q}[x_1, \ldots, x_n]$. Then R is a differential ring with derivations $\delta_i : R \to R : p \mapsto \frac{\partial p}{\partial x_i}$, $i = 1, \ldots, n$. The quotient field $F = \mathbb{Q}(x_1, \ldots, x_n)$ of R is a differential field, where δ_i is extended to $F \to F : r \mapsto \frac{\partial r}{\partial x_i}$, $i = 1, \ldots, n$.

(c) Let $G \subseteq \mathbb{C}$ be open and connected. Then the field of meromorphic functions $G \to \mathbb{C}$ is a differential field with derivation $f \mapsto \frac{df}{dz}$.

For simplicity and in view of our applications we confine ourselves to differential fields of characteristic zero in what follows.

Remark 3.3.3. Systems of partial differential equations with only *polynomial nonlinearities* in the dependent variables are usually considered as follows. Let $\underline{u} := \{u^1, \ldots, u^q\}$ be a set of indeterminates which are supposed to represent the unknown scalar-valued functions of the system of PDEs. By $\delta_1, \ldots, \delta_n$ we denote the partial differential operators with respect to the independent variables $x^1, \ldots, x^n$. Moreover, let F be a field containing all coefficients of the partial differential equations under consideration, and view $\delta_1, \ldots, \delta_n$ as derivations on F by their differentiating action on these coefficients (i.e. if all coefficients of the equations are real numbers, then F can be chosen to be $\mathbb{R}$ with trivial derivations $\delta_1, \ldots, \delta_n$; if the coefficients of the PDEs exclusively consist of rational functions in $x^1, \ldots, x^n$, then we set $F := \mathbb{R}(x^1, \ldots, x^n)$ with the usual derivations). Hence, F is a differential field. Note that $\delta_1, \ldots, \delta_n$ do not act on the indeterminates $u^1, \ldots, u^q$, so that we can consider

$$\Delta\underline{u} := \{\delta_1^{a_1} \cdots \delta_n^{a_n} u^k \mid a \in (\mathbb{Z}_{\geq 0})^n, 1 \leq k \leq q\}$$

as a set of indeterminates.

Definition 3.3.4. Let $\underline{u} = \{u^1, \ldots, u^q\}$, $\delta_1, \ldots, \delta_n$, F, and Δ be defined as in the preceding remark. Then the (commutative) polynomial algebra

$$F\{\underline{u}\} := F[\Delta\underline{u}]$$

over F consists of all *differential polynomials* in $u^1, \ldots, u^q$ with coefficients in F. It is also customary to write the variables $\delta_1^{a_1} \cdots \delta_n^{a_n} u^k$ of $F\{\underline{u}\}$ in jet notation as u_J^k with $J = (a_1, \ldots, a_n) \in (\mathbb{Z}_{\geq 0})^n$, $k = 1, \ldots, q$.

Remark 3.3.5. $F\{\underline{u}\}$ is a differential algebra over F and an integral domain. It has the following universal property similar to any polynomial algebra. For every differential F-algebra A and every map $\phi : \underline{u} \to A$ there exists a unique homomorphism $\psi : F\{\underline{u}\} \to A$ of differential F-algebras which extends ϕ, i.e. $\psi|_{\underline{u}} = \phi$. This property is traced back to polynomial algebras in a straightforward way.

We quote a theorem by Ritt and Raudenbush [Rit66] (see also [BLOP95], [Hub00]) which establishes a prime decomposition of radical differential ideals similar to the decomposition of radical ideals in commutative algebra.

Theorem 3.3.6. *Every radical differential ideal I of $F\{\underline{u}\}$ is an intersection of finitely many prime differential ideals, i.e. there exist prime differential ideals P_1, $\ldots$, P_r of $F\{\underline{u}\}$ such that*

$$I = P_1 \cap \ldots \cap P_r.$$

Remark 3.3.7. In the situation of Remark 3.3.3 let $R := F\{\underline{u}\}$ be the F-algebra of differential polynomials in $u^1, \ldots, u^q$ with coefficients in F. A system of partial differential equations with independent variables $x^1, \ldots, x^n$ and dependent variables $u^1, \ldots, u^q$, where every nonlinear term in the dependent variables is a polynomial, can then be written (in jet notation) as

$$\left\{ \begin{array}{rcl} p_1(u_J^k) & = & 0, \\ & \vdots & \\ p_m(u_J^k) & = & 0 \end{array} \right. \tag{3.18}$$

with finitely many differential polynomials $p_1, \ldots, p_m \in R$. Note that R is not Noetherian, but every system of polynomial partial differential equations involves only finitely many indeterminates of R which allows to use constructive methods nevertheless.

Remark 3.3.8. Let us investigate a system of partial differential equations which are polynomial in the dependent variables and assume it is represented by (3.18). Let I be the differential ideal generated by $p_1, \ldots, p_m$ in $F\{\underline{u}\}$. We recall that the differential field F is defined in such a way that it contains all coefficients of these equations; e.g. $F = \mathbb{R}(x^1, \ldots, x^n)$ if the coefficients are rational functions. By identifying $\delta_1^{a_1} \cdots \delta_n^{a_n} u^k$ with $u_{(a_1,\ldots,a_n)}^k$ the general linearization $P_i(u_J^k, U_K^l)$ of $p_i(u_J^k)$ can formally be determined as in Def. 3.2.8, so that we obtain the linearized system

$$\left\{ \begin{array}{rcl} P_1(u_J^k, U_K^l) & = & 0, \\ & \vdots & \\ P_m(u_J^k, U_K^l) & = & 0. \end{array} \right. \tag{3.19}$$

Formally, the linearized system (3.19) could be treated by Janet's algorithm as a system of partial differential equations with coefficients which are rational functions in x^i, u^k_J. Then the left hand sides $P_i(u^k_J, U^l_K)$ are identified with elements in the free left module

$$\bigoplus_{l=1}^{q} D\,U^l$$

over the Ore algebra

$$D := \mathrm{Quot}(F\{\underline{u}\})[\partial_1; \sigma_1, \delta_1] \ldots [\partial_n; \sigma_n, \delta_n], \tag{3.20}$$

where $\sigma_i = \mathrm{id}_D$, $\delta_i = \frac{\partial}{\partial x^i}$, $i = 1, \ldots, n$ (cf. the definition of the Weyl algebra, Ex. 2.4.5 (a)) via

$$U^l_K \leftrightarrow \partial^K U^l, \qquad \partial^K := \partial_1^{k_1} \cdots \partial_n^{k_n}, \quad K = (k_1, \ldots, k_n) \in (\mathbb{Z}_{\geq 0})^n, \quad l = 1, \ldots, q.$$

However, this "generic" computation can be undefined due to division by zero when a certain solution v of (3.18) is substituted for u in the linearized system (3.19) (where the substitution is extended to all of $F\{\underline{u}\}$ by the universal property described in Rem. 3.3.5). If the differential ideal I generated by $p_1, \ldots, p_m$ is radical and a prime decomposition

$$I = P_1 \cap \ldots \cap P_r$$

of I is known (see Theorem 3.3.6), then the computation over D from (3.20) should therefore be replaced by the computations over

$$D_i := \mathrm{Quot}(F\{\underline{u}\}/P_i)[\partial_1; \sigma_1, \delta_1] \ldots [\partial_n; \sigma_n, \delta_n], \qquad \sigma_i = \mathrm{id}_{D_i}, \quad \delta_i = \frac{\partial}{\partial x^i},$$

for each minimal prime ideal among the P_i.

The implementation of Janet's algorithm needs to be adapted to the coefficient domains D_i. The Maple package `Janet` provides a rather simple way to deal with these coefficients which works in many examples. Let G be a finite generating set for P_i. The method which implements an arithmetic for D_i requires that the equations $g = 0$, $g \in G$, are solved for certain indeterminates u^k_J in such a way that rewriting rules for the u^k_J are defined that do not result in infinite loops. Using a method from the Maple package `jets` [Bar01b], in each step of the Janet-reduction the given rewriting rules for indeterminates u^k_J are applied to each coefficient representing a residue class in $\mathrm{Quot}(F\{\underline{u}\}/P_i)$ in order to obtain a "reduced" representative. In particular, the "reduced" representative of the zero residue class is expected to be zero when using the given rewriting rules. Then no division by zero can occur when specializing u to a solution v of (3.18). Anyway, the implementation in `Janet` collects all expressions by which JanetBasis divides, so that divisions by zero for certain solutions v can be detected in case the rewriting rules are inappropriate. This even permits to embark on another strategy, namely to compute "generically" over the Ore algebra D defined in (3.20) and to determine critical expressions for denominators afterwards. Hence, it is possible to find certain solutions v of (3.18) that are "singular" in some respect (cf. Ex. 5.3.2).

The technique described above will be applied in Chapter 5. To finish the present section we illustrate possible difficulties with this approach. In particular, the following example demonstrates that, when the initial system of nonlinear partial differential equations is not preprocessed adequately, an expression occurring in some denominator during a Janet basis computation can be zero for all specializations of u to any solution v of (3.18). Hence, a prime decomposition of the differential ideal generated by the left hand sides in (3.18) is necessary to avoid such effects. But this example also demonstrates another limitation of our approach.

Example 3.3.9. In this example we consider a polynomially nonlinear system of PDEs for one unknown function. We continue to replace the multi-index in a jet variable $u_{(i,j)}$ by a sequence of i symbols x and j symbols y. Hence, $u_{(2,1)}$ is also written as $u_{x,x,y}$.

```
>  with(Janet):
>  with(jets):
```

Let us consider the following system of nonlinear partial differential equations with independent variables x, y and dependent variable u:

$$\left\{ \begin{array}{rcl} u_x &=& u^2, \\ u_{y,y} &=& u^3. \end{array} \right. \tag{3.21}$$

We define the independent and dependent variables:

```
>  ivar := [x,y]; dvar := [u];
```

$$ivar := [x,\, y]$$

$$dvar := [u]$$

The left hand sides of the partial differential equations are given by the list L:

```
>  L := [diff(u(x,y),x)-u(x,y)^2, diff(u(x,y),y,y)-u(x,y)^3];
```

$$L := [(\tfrac{\partial}{\partial x}\,\mathrm{u}(x,\, y)) - \mathrm{u}(x,\, y)^2,\ (\tfrac{\partial^2}{\partial y^2}\,\mathrm{u}(x,\, y)) - \mathrm{u}(x,\, y)^3]$$

Maple is able to find the following family of solutions, where $\_C1$ is an arbitrary constant:

```
> S := pdsolve(L);
```

$$S := \{\mathrm{u}(x,\, y) = \frac{2}{-2\,x + \sqrt{2}\,y + 2\,\_C1}\}$$

For later use we assign this family of solutions to p:

```
> p := rhs(op(S));
```

$$p := \frac{2}{-2\,x + \sqrt{2}\,y + 2\,\_C1}$$

Using the command `Linearize` we compute the general linearization of the system of partial differential equations given by L. The new dependent variable for the general linearization is chosen to be U:

```
> Dvar := [U];
```

$$Dvar := [U]$$

We obtain the following linearization:

```
> GL := Linearize(L, ivar, dvar, Dvar);
```

$$GL := [(\tfrac{\partial}{\partial x}\,\mathrm{U}(x,\, y)) - 2\,\mathrm{u}(x,\, y)\,\mathrm{U}(x,\, y),\, (\tfrac{\partial^2}{\partial y^2}\,\mathrm{U}(x,\, y)) - 3\,\mathrm{u}(x,\, y)^2\,\mathrm{U}(x,\, y)]$$

Now we are going to compute a Janet basis for this linear system of partial differential equations over the Ore algebra $\mathrm{Quot}(\mathbb{Q}(x,y)\{u\})[\partial_x, \partial_y]$, where ∂_x resp. ∂_y represents differentiation with respect to x resp. y. We choose (3.21) as rewriting rules for the coefficients u_J.

```
> N := [u[x]=u^2, u[y,y]=u^3];
```

$$N := [u_x = u^2,\, u_{y,\,y} = u^3]$$

```
> JanetBasis(GL, ivar, Dvar, "coeffeqs"=N, "coeffdvar"=dvar);
```

$$[[\mathrm{U}(x,\, y)],\, [x,\, y],\, [U]]$$

The list of expressions by which `JanetBasis` divided is the following:

```
> ZeroSets(ivar);
```

$$[\tfrac{\partial}{\partial y}\,\mathrm{u}(x,\, y),\, -\mathrm{u}(x,\, y)^4 + 2\,(\tfrac{\partial}{\partial y}\,\mathrm{u}(x,\, y))^2]$$

We investigate the second expression more closely. First we translate it into jet notation:

```
> d := Diff2Ind(%[2], ivar, dvar);
```

$$d := -u^4 + 2\,{u_y}^2$$

By substituting the general solution p for u in d we realize that the expression d vanishes for all solutions.

```
> jsubs(u=p, d, ivar, dvar);
```

$$0$$

When tracing the above computation of `JanetBasis` we discover that the expression d occurs only as the coefficient of $U(x, y)$ in the last step of Janet's algorithm. In order to normalize the resulting Janet basis, the present implementation divides by d which is a division by zero for each specialization of u as we have just seen. Therefore, $d\,U(x, y) = 0$ in the last step of Janet's algorithm, which means that the Janet basis is $\{U_x - 2\,u,\ u_y\,U_y - u^3\}$.

The problem for the above application of `JanetBasis` was that no substitution rule for u_y was declared which needs to be known in advance. Returning to the method described in Remark 3.3.8, we want to find a prime decomposition of the radical of the differential ideal generated by $u_x - u^2,\quad u_{y,y} - u^3$. The theory of characteristic sets [Rit66], [Kol73] allows to construct such a decomposition for polynomially nonlinear partial differential equations. Characteristic sets can be computed in Maple by using the package **diffalg** [BLOP95], [Hub99], [Hub00].

```
> with(diffalg):
```

First we define the ring $R := \mathbb{Q}[\Delta U]$ of differential polynomials (see Remark 3.3.3) where $U = \{u\}$, $\Delta = (\delta_x, \delta_y)$.

```
> R := differential_ring(derivations=[x,y], ranking=[u]);
```

$$R := \mathit{PDE\_ring}$$

The generating set for an ideal in R is given in jet notation:

```
> L2 := [u[x]-u[]^2, u[y,y]-u[]^3];
```

$$L2 := [u_x - u^2,\ u_{y,\,y} - u^3]$$

The next commands computes a decomposition of the differential ideal generated by $L2$ into "characterizable" differential ideal. This decomposition is similar to the primary decomposition in commutative algebra because a characterizable ideal is an intersection of prime differential ideals.

```
> P := Rosenfeld_Groebner(L2, R);
```

$$P := [\mathit{characterizable},\ \mathit{characterizable}]$$

Hence, the given differential ideal is decomposed into two characterizable ideals I_1, I_2. A characterizable ideal is represented in Maple by two lists of differential polynomials. The first one consists of left hand sides of differential equations derived from the given generating set of differential polynomials. The second list specifies inequalities for differential polynomials which were assumed in the process of decomposition, e.g. when dividing by a leading term. We display these lists for the computed characterizable ideals:

```
> equations(P[1]), inequations(P[1]);
```

$$[u_x - u^2,\, 2\,{u_y}^2 - u^4],\, [u_y]$$

The second entry in the first list of left hand sides of differential equations is exactly the term d which caused the problem in the previous run of Janet's algorithm. The other characterizable ideal is represented by a single differential polynomial only:

```
> equations(P[2]), inequations(P[2]);
```

$$[u],\, []$$

We now use the above preprocessing of the system of nonlinear equations in order to perform a correct run of Janet's algorithm on the general linearization GL. Since the leading derivative of d is quadratic, we have to choose a square root:

```
> N1 := [u[x]=u^2, u[y]=1/sqrt(2)*u^2];
```

$$N1 := [u_x = u^2,\, u_y = \frac{\sqrt{2}\,u^2}{2}]$$

We compute the Janet basis of the general linearization over the Ore algebra $\mathrm{Quot}(\mathbb{Q}(x,y)\{u\}/I_1)[\partial_x, \partial_y]$:

```
> J1 := JanetBasis(GL, ivar, Dvar, "coeffeqs"=N1, "coeffdvar"=dvar);
```

$$J1 := [[-\mathrm{u}(x,\,y)\,\sqrt{2}\,\mathrm{U}(x,\,y) + (\frac{\partial}{\partial y}\,\mathrm{U}(x,\,y)),\, (\frac{\partial}{\partial x}\,\mathrm{U}(x,\,y)) - 2\,\mathrm{u}(x,\,y)\,\mathrm{U}(x,\,y)],\, [x,\,y],\, [U]]$$

The algorithm divided by the following expressions:

```
> ZeroSets(ivar);
```

$$[\frac{1}{2}\,\sqrt{2}\,\mathrm{u}(x,\,y)^2,\, \sqrt{2}]$$

Now we solve the sytem given by the Janet basis and obtain, of course, the zero solution and another more complicated one.

```
> pdsolve(J1[1]);
```

$$\{\mathrm{U}(x,\, y) = 0,\, \mathrm{u}(x,\, y) = \mathrm{u}(x,\, y)\},$$
$$\{\mathrm{u}(x,\, y) = \frac{1}{2}\,\frac{\mathrm{D}(\_\mathrm{F1})(y + \sqrt{2}\,x)\,\sqrt{2}}{\_\mathrm{F1}(y + \sqrt{2}\,x)},\, \mathrm{U}(x,\, y) = \_\mathrm{F1}(y + \sqrt{2}\,x)\}$$

```
> solu := rhs(%[2][1]); solU := rhs(%%[2][2]);
```

$$solu := \frac{1}{2}\,\frac{\mathrm{D}(\_\mathrm{F1})(y + \sqrt{2}\,x)\,\sqrt{2}}{\_\mathrm{F1}(y + \sqrt{2}\,x)}$$
$$solU := \_\mathrm{F1}(y + \sqrt{2}\,x)$$

Substituting the second solution for the dependent variables in the general linearization reveals that it is in fact only a solution of the linearization of the system defined by the prime ideal I_1:

```
> jsubs([u=solu, U=solU], Diff2Ind(GL, ivar, [op(dvar),op(Dvar)]),
> ivar, [op(dvar),op(Dvar)]);
```

$$[0,\, -\frac{1}{2}\,\frac{-2\,(\mathrm{D}^{(2)})(\_\mathrm{F1})(y - \sqrt{2}\,x)\,\_\mathrm{F1}(y - \sqrt{2}\,x) + 3\,\mathrm{D}(\_\mathrm{F1})(y - \sqrt{2}\,x)^2}{\_\mathrm{F1}(y - \sqrt{2}\,x)}]$$

We continue in a similar way with the other square root:

```
> N2 := [u[x]=u^2, u[y]=-1/sqrt(2)*u^2];
```

$$N2 := [u_x = u^2,\, u_y = -\frac{\sqrt{2}\,u^2}{2}]$$

```
> J2 := JanetBasis(GL, ivar, Dvar, "coeffeqs"=N2, "coeffdvar"=dvar);
```

$$J2 := [[\mathrm{u}(x,\, y)\,\sqrt{2}\,\mathrm{U}(x,\, y) + (\tfrac{\partial}{\partial y}\,\mathrm{U}(x,\, y)),\, (\tfrac{\partial}{\partial x}\,\mathrm{U}(x,\, y)) - 2\,\mathrm{u}(x,\, y)\,\mathrm{U}(x,\, y)],\, [x,\, y],\, [U]]$$

The solutions of the system given by the Janet basis are:

```
> pdsolve(J2[1]);
```

$$\{\mathrm{U}(x,\, y) = 0,\, \mathrm{u}(x,\, y) = \mathrm{u}(x,\, y)\},$$
$$\{\mathrm{u}(x,\, y) = -\frac{1}{2}\,\frac{\mathrm{D}(\_\mathrm{F1})(y - \sqrt{2}\,x)\,\sqrt{2}}{\_\mathrm{F1}(y - \sqrt{2}\,x)},\, \mathrm{U}(x,\, y) = \_\mathrm{F1}(y - \sqrt{2}\,x)\}$$

```
> solu := rhs(%[2][1]); solU := rhs(%%[2][2]);
```

$$solu := -\frac{1}{2}\,\frac{\mathrm{D}(\_\mathrm{F1})(y - \sqrt{2}\,x)\,\sqrt{2}}{\_\mathrm{F1}(y - \sqrt{2}\,x)}$$
$$solU := \_\mathrm{F1}(y - \sqrt{2}\,x)$$

Again, the second solution is only a solution of the linearization of the system defined by the prime ideal I_1:

```
> jsubs([u=solu, U=solU], Diff2Ind(GL, ivar, [op(dvar),op(Dvar)]),
> ivar, [op(dvar),op(Dvar)]);
```

$$[0, -\frac{1}{2}\,\frac{-2\,(\mathrm{D}^{(2)})(\_\mathrm{F1})(y-\sqrt{2}\,x)\,\_\mathrm{F1}(y-\sqrt{2}\,x)+3\,\mathrm{D}(\_\mathrm{F1})(y-\sqrt{2}\,x)^2}{\_\mathrm{F1}(y-\sqrt{2}\,x)}]$$

This example demonstrated that a careful preprocessing of the nonlinear equations is needed in order to compute correctly with the coefficients of the general linearization. More tools for symbolic computation with differential equations are needed to be able to handle linearizations automatically.

Chapter 4

The Generalized Hilbert Series

In this chapter we show the importance of the generalized Hilbert series which was introduced in Section 2.3. First we explain the module-theoretic approach to linear systems which is used throughout this thesis. In Section 4.2, very important notions from homological algebra are collected. The following two sections then deal with injective modules resp. cogenerators which play a crucial role in the algebraic analysis of linear systems (see Chapters 5 and 6). The generalized Hilbert series is exemplified for the case of partial differential equations in Section 4.5. Finally, the Bernstein filtration of the Weyl algebras is generalized to a certain class of Ore algebras and the relation of the generalized Hilbert series to the Hilbert series of left modules over these algebras with grading induced by the Bernstein filtration is shown in Section 4.6.

4.1 Module-theoretic Approach to Linear Systems

In the present and the following chapters we study systems governed by equations with coefficients in an Ore algebra D. This case is sufficiently general for applications in systems theory because by means of the Weyl algebras we can model partial differential equations, and there are also Ore algebras to describe difference equations, discrete systems, etc. In general, given a linear system of equations, the Ore algebra D is chosen in such a way that it contains the polynomials in all operators occurring in the equations. Constant coefficients can be viewed as multiplication operators. For the arising algebraic problems to be algorithmically solvable it is usually required that the ring is at least (left) Noetherian. One can show that every (left) Noetherian domain satisfies the (left) Ore property (see Prop. 2.4.9) which also justifies the choice of Ore algebras.

In this section we show that with every linear system a left D-module is associated. The restriction to left modules in the non-commutative case is just by convention. All constructions can be applied to right modules as well. After giving the most important examples for the types of linear systems that we are going to examine and the corresponding Ore algebras over which these systems can be considered, the module-theoretic approach to linear systems is described. Then Malgrange's isomorphism is recalled which links the left D-module which is associated to the given linear system to the solutions in a fixed left D-module $\mathcal{F}$.

First we give a few examples of types of linear systems that can be considered in the framework of Ore algebras.

Examples 4.1.1. (a) (Ordinary and partial differential equations)

Let $k \in \{\mathbb{R}, \mathbb{C}\}$. By definition of the *Weyl algebra* $A_1(k)$ (cf. Ex. 2.4.5 (a), p. 29) we have

$$\partial\, a = a\, \partial + \frac{da}{dt} \qquad a \in k[t].$$

This commutation rule expresses the product rule for differentiation with respect to t: for differentiable functions $a = a(t)$, $y = y(t)$ we have:

$$\frac{d}{dt}(a \cdot y) = a \frac{d}{dt}\, y + \frac{da}{dt}\, y.$$

The differential operator $\frac{d}{dt}$ is represented by the element ∂ in A_1. Hence, we can associate with any "Kalman system" with polynomial coefficients

$$\dot{x} = A\, x + B\, u, \qquad A \in k[t]^{n\times n}, \quad B \in k[t]^{n\times m},$$

the matrix

$$R = (\partial\, I_n - A \quad -B) \in A_1^{n\times(n+m)}$$

such that $R\, (x(t)^T \quad u(t)^T)^T = 0$.

Partial differential equations are treated in a similar way. The commutation rules of $A_n(k)$ exactly represent the product rules for the partial differentiations. Moreover, linear systems of partial differential equations whose coefficients are rational functions can be represented by matrices over the localized Weyl algebra $B_n(k)$ (cf. Ex. 2.4.5 (a)).

(b) (Time-delay systems)

Let $h \in \mathbb{R}$. For the algebra S_h of shift operators of length h (cf. Ex. 2.4.5 (b)) we have

$$\delta_h\, p(t) = p(t-h)\, \delta_h, \qquad p \in \mathbb{R}[t],$$

representing the "product rule" for the shift action on polynomials[1]:

$$\delta_h(f(t) \cdot g(t)) = f(t-h)\, \delta_h\, g(t), \qquad f, g \in \mathbb{R}[t].$$

[1]In this example, the indeterminate δ_h of the Ore algebra S_h and the shift operator which it represents are identified.

The time-delay system $x(t) = x(t-h) + u(t-2h)$ can then be represented by the matrix

$$R = (1 - \delta_h \quad -\delta_h^2) \in S_h^{1\times 2},$$

such that $R\,(x(t) \quad u(t))^T = 0$.

(c) (Differential time-delay systems)

For linear differential time-delay systems both of the previous Ore extensions can be applied at the same time. The Ore algebra D_h defined in Ex. 2.4.5 (c) combines the product rules of the two previous examples. For instance, the matrix

$$R = (\partial\, I - A \quad -B\,\delta_h) \in D_h^{n\times(n+m)}, \qquad A \in \mathbb{R}[t]^{n\times n}, \quad B \in \mathbb{R}[t]^{n\times m},$$

represents the linear system

$$R \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} = \dot{x}(t) - A(t)\,x(t) - B(t)\,u(t-h) = 0.$$

(d) (Discrete systems)

For (multidimensional) discrete linear systems we choose the Ore algebra defined in Ex. 2.4.5 (d). For instance, the equation

$$f(n) = f(n-1) + f(n-2)$$

defining the Fibonacci sequence (neglecting for now the initial conditions $f(0) = 0$, $f(1) = 1$) can be represented by the matrix

$$R = (1 - \partial - \partial^2) \in D^{1\times 1}$$

over $D = \mathbb{Q}[n][\partial; \sigma, \delta]$ which is a special case of Ex. 2.4.5 (d). The domain of coefficients $\mathbb{Q}[n]$ could be replaced just by $\mathbb{Q}$ because the above equation has constant coefficients.

In what follows, let D be an Ore algebra which is in particular a k-algebra, where k is a field. Moreover, for the time being we consider any left D-module $\mathcal{F}$. This module will play the role of a *signal space*. It is usually chosen as a space of functions which is acted upon by D on the left. The solutions (trajectories) of a linear system $R\,y = 0$ are then searched for in $\mathcal{F}^{p\times 1}$. Further properties of a signal space rather than just being a left D-module will be specified later, when needed.

Convention 4.1.2. *With every matrix $R \in D^{q\times p}$ we associate two maps. On the one hand we consider the homomorphism of left D-modules*

$$(.R) : D^{1\times q} \to D^{1\times p} : m \mapsto m\,R.$$

On the other hand we have a homomorphism of abelian groups

$$(R.) : \mathcal{F}^{p\times 1} \to \mathcal{F}^{q\times 1} : \eta \mapsto R\,\eta.$$

If D is commutative, then $(R.)$ is also a homomorphism of D-modules. Brackets in $(.R)$ and $(R.)$ are often omitted.

Definition 4.1.3. The *solution set* of $R\,\eta = 0$ in $\mathcal{F}$, where $R \in D^{q\times p}$, is defined as

$$\mathrm{Sol}_{\mathcal{F}}(R) := \ker(R.) = \{\eta \in \mathcal{F}^{p\times 1} \mid R\,\eta = 0\}.$$

A set $\mathcal{B} \subseteq \mathcal{F}^{p\times 1}$ which is a solution set $\mathrm{Sol}_{\mathcal{F}}(R)$ for some $R \in D^{q\times p}$, is called a *behavior*. We also write

$$\mathcal{B}_{\mathcal{F}}(R) := \mathrm{Sol}_{\mathcal{F}}(R).$$

Definition 4.1.4. Let

$$\sum_{j=1}^{p} R_{ij}\,\eta_j = 0, \quad i = 1, \ldots, q,$$

be a linear system defined over an Ore algebra D, i.e. $R \in D^{q\times p}$. Then

$$M := D^{1\times p}/D^{1\times q}\,R = \mathrm{coker}(.R)$$

is the *left D-module associated with $R\,\eta = 0$.*

Remark 4.1.5. By applying the rows of R to the vector $\eta = (\eta_1, \ldots, \eta_p)^T$, we obtain the defining equations of the given linear system. The left D-module M is defined in such a way that all left D-linear combinations of the residue classes represented by the rows of R are zero in M. Therefore, we associate to the linear system an algebraic object which is independent of the choice of the equations which describe the system. By investigating the algebraic properties of M we study the structural properties of the given linear system.

Definition 4.1.6. Let M, N be left D-modules. Then $\hom_D(M, N)$ denotes the set of all homomorphisms $f : M \to N$. In the special case $N = D$, we also write M^* for $\hom_D(M, D)$.

Remark 4.1.7. In general, $\hom_D(M, N)$ is an abelian group. If M is a left D-module and N is both a left and a right D-module, then $\hom_D(M, N)$ can be viewed as a right D-module in virtue of

$$(\varphi \cdot d)(m) := \varphi(m) \cdot d, \qquad d \in D, \quad m \in M, \qquad \varphi \in \hom_D(M, N).$$

It is easily checked that $\varphi \cdot d$ defines a homomorphism $M \to N$ of left D-modules. For all $a_1, a_2 \in D$, $m_1, m_2 \in M$ we have:

$$\begin{aligned}(\varphi \cdot d)(a_1\, m_1 + a_2\, m_2) &= \varphi(a_1\, m_1 + a_2\, m_2) \cdot d \\ &= (a_1\, \varphi(m_1) + a_2\, \varphi(m_2))\, d \\ &= a_1\, (\varphi \cdot d)(m_1) + a_2\, (\varphi \cdot d)(m_2).\end{aligned}$$

In particular, this holds for the free left and right D-module $N = D$.

If D is a commutative ring, then for all D-modules M and N, $\hom_D(M, N)$ is a D-module, where the action of D on $\hom_D(M, N)$ is simply defined by

$$(d \cdot \varphi)(m) := d \cdot \varphi(m), \qquad d \in D, \quad m \in M.$$

The following proposition is crucial for the interpretation of module-theoretic constructions which involve the left D-module M associated to a linear system for its solution space.

Proposition 4.1.8 (Malgrange's isomorphism)**.** [Mal63] *Let $M = D^{1\times p}/D^{1\times q}\, R$ be the left D-module associated with a linear system $R\,\eta = 0$ and $\mathcal{F}$ a left D-module. Then we have*

$$\hom_D(M, \mathcal{F}) \cong \mathrm{Sol}_{\mathcal{F}}(R) = \{\eta \in \mathcal{F}^{p\times 1} \mid R\,\eta = 0\}$$

as k-vector spaces.

Proof. Let $(e_1, \ldots, e_p)$ be a basis of $D^{1\times p}$ and denote by $\overline{e_i} := e_i + D^{1\times q}\, R$ the residue class of e_i in M, $i = 1, \ldots, p$. Then

$$\pi : \hom_D(M, \mathcal{F}) \longrightarrow \mathrm{Sol}_{\mathcal{F}}(R) : \varphi \longmapsto (\varphi(\overline{e_j}))_{j=1,\ldots,p}\,,$$

is a well-defined homomorphism of k-vector spaces because

$$\begin{aligned} R\, (\varphi(\overline{e_j}))_{j=1,\ldots,p} &= \left(\textstyle\sum_{j=1}^{p} R_{ij}\, \varphi(\overline{e_j})\right)_{i=1,\ldots,q} \\ &= \left(\varphi\left(\textstyle\sum_{j=1}^{p} R_{ij}\, e_j + D^{1\times q}\, R\right)\right)_{i=1,\ldots,q} \\ &= (\varphi(0 + D^{1\times q}\, R))_{i=1,\ldots,q} \;=\; 0. \end{aligned} \tag{4.1}$$

By the universal property of the free left D-module $D^{1\times p}$ every homomorphism $D^{1\times p} \to \mathcal{F}$ is uniquely determined by its values on the basis $(e_1, \ldots, e_p)$. Therefore, every $\varphi \in \hom_D(M, \mathcal{F})$ is uniquely determined by specifying $\varphi(\overline{e_1})$, $\ldots$, $\varphi(\overline{e_p})$ which satisfy (4.1). This defines a homomorphism of k-vector spaces which is inverse to π. □

Remark 4.1.9. In the case of linear systems with constant coefficients, i.e. whenever D is chosen to be commutative, $\hom_D(M, \mathcal{F})$ is also a D-module, and the above isomorphism of vector spaces is also an isomorphism of D-modules.

In the case of non-constant coefficients, $\hom_D(M, \mathcal{F})$ is only a k-vector space and $\mathrm{Sol}_{\mathcal{F}}(R)$ is also a k-vector space only, because in general a left D-multiple of a solution of the system equations is not a solution of the system equations.

4.2 Homological Algebra

The algebraic notions which are defined in this section are important for the algebraic approach to linear systems described in the subsequent sections and chapters. They will be applied to the left D-module $M = D^{1\times p}/D^{1\times q}\,R$ which is associated with a given linear system $R\,\eta = 0$, where D is an Ore algebra. Homological algebra provides an additional level of abstraction for dealing not only with a single module but with families of modules which are related by homomorphisms between certain of them. By embedding the theory of modules into the more general theory of chain complexes, deeper invariants for modules are obtained. A Maple package `homalg` which implements certain constructions of homological algebra in an abstract way is in preparation at Lehrstuhl B für Mathematik, RWTH Aachen [BR, BR06]. This package computes with modules by letting another Maple package perform the ring arithmetics as specified by the user. In this way the package is designed to depend as little as possible on the rings which are admissible. However, `homalg` is not used in this thesis.

All algebraic notions of this section can be found in more detail in [CE56], [Rot79], or [HS97]. In what follows let D be a (not necessarily commutative) algebra over a field k and a left Ore domain. We consider left D-modules, but all concepts are easily translated to right D-modules as well.

Definition 4.2.1. Let $M = (M_i)_{i\in\mathbb{Z}}$ be a family of left D-modules.

(a) Let $d = (d_i)_{i\in\mathbb{Z}}$ be a family of homomorphisms $d_i : M_i \longrightarrow M_{i-1}$ of left D-modules. (M, d) is called a *chain complex* (of left D-modules), if $d_i \circ d_{i+1} = 0$ holds for all $i \in \mathbb{Z}$, i.e. $\operatorname{im} d_{i+1} \subseteq \ker d_i$ for all $i \in \mathbb{Z}$. A chain complex is written down as follows:

$$\ldots \xrightarrow{d_{i+2}} M_{i+1} \xrightarrow{d_{i+1}} M_i \xrightarrow{d_i} M_{i-1} \xrightarrow{d_{i-1}} \ldots$$

Let $\delta = (\delta_i)_{i\in\mathbb{Z}}$ be a family of homomorphisms $\delta_i : M_{i-1} \longrightarrow M_i$ of left D-modules. (M, δ) is called a *cochain complex* (of left D-modules), if $\delta_{i+1} \circ \delta_i = 0$ for all $i \in \mathbb{Z}$. A cochain complex is written down as a chain complex, but with reversed arrows. We also write *cocomplex* for cochain complex.

(b) The *defect of exactness* of a complex (M, d) at the i-th position or the i-th *homology group* of (M, d) is defined by

$$H_i(M, d) := \ker d_i / \operatorname{im} d_{i+1}.$$

The complex (M, d) is said to be *exact at the i-th position*, if $H_i(M, d) = 0$, i.e. if $\ker d_i = \operatorname{im} d_{i+1}$. The complex (M, d) is called *exact*, if it is exact at the i-th position for all $i \in \mathbb{Z}$. An exact complex is also called an *exact sequence* (of left D-modules).

(c) A *short exact sequence* (of left D-modules) is an exact sequence (M, d) of left D-modules for which all modules M_i are zero except for three consecutive ones. If we call these modules M', M respectively M'', then a short exact sequence is denoted as follows:

$$0 \longrightarrow M' \xrightarrow{f} M \xrightarrow{g} M'' \longrightarrow 0. \tag{4.2}$$

Remarks 4.2.2. (a) A complex $0 \longrightarrow M_1 \xrightarrow{f} M_0$ is exact if and only if f is an injective homomorphism.

(b) A complex $M_1 \xrightarrow{g} M_0 \longrightarrow 0$ is exact if and only if g is a surjective homomorphism.

(c) Let us consider the complex (4.2) of left D-modules. This complex is a short exact sequence if and only if f is injective and $\ker g = \operatorname{im} f$ and g is surjective.

Remark 4.2.3. Let

$$\cdots \xrightarrow{d_{i+2}} M_{i+1} \xrightarrow{d_{i+1}} M_i \xrightarrow{d_i} M_{i-1} \xrightarrow{d_{i-1}} \cdots \tag{4.3}$$

be a complex of left D-modules and let $\mathcal{F}$ be a left D-module. By *applying* $\hom_D(\,\cdot\,,\mathcal{F})$ *to* (4.3) we mean to apply $\hom_D(\,\cdot\,,\mathcal{F})$ to each module in (4.3). Then we obtain a cocomplex

$$\cdots \xleftarrow{d^*_{i+2}} \hom_D(M_{i+1},\mathcal{F}) \xleftarrow{d^*_{i+1}} \hom_D(M_i,\mathcal{F}) \xleftarrow{d^*_i} \hom_D(M_{i-1},\mathcal{F}) \xleftarrow{d^*_{i-1}} \cdots \tag{4.4}$$

of abelian groups, where the homomorphism d^*_i, $i \in \mathbb{Z}$, is defined by

$$d^*_i(\varphi) = \varphi \circ d_i, \qquad \varphi \in \hom_D(M_{i-1},\mathcal{F}).$$

If D is a commutative ring, then (4.4) is also a complex of D-modules. Note that even if (4.3) is exact, (4.4) need not be exact.

Definition 4.2.4. Let M be a left D-module. The submodule

$$t(M) := \{m \in M \mid \exists\, 0 \neq d \in D : d\, m = 0\}$$

of M is the *torsion submodule* of M; its elements are the *torsion elements* of M. The module M is *torsion-free* if $t(M) = \{0\}$. It is *torsion* if $t(M) = M$.

Example 4.2.5. In the situation of Def. 4.2.4 we have the short exact sequence

$$0 \longrightarrow t(M) \xrightarrow{\iota} M \xrightarrow{\rho} M/t(M) \longrightarrow 0$$

where ι and ρ denote the canonical injection respectively the canonical projection. It is common practice in homological algebra to omit the maps which are canonical when writing down a complex. Hence, e.g. in Chapter 6, the names ι and ρ are no longer attached to the above arrows.

Definition 4.2.6. A left D-module M is *projective* if it is a direct summand of each left D-module (up to isomorphism) of which it is an epimorphic image, more precisely, for every epimorphism $\pi : B \to M$ of left D-modules, there exists a homomorphism $\alpha : M \to B$ such that $\pi \circ \alpha = \mathrm{id}_M$.

Definition 4.2.7. A left D-module M is said to be *reflexive* if the canonical map

$$\varepsilon_M : M \longrightarrow \hom_D(\hom_D(M, D), D) : m \longmapsto (f \mapsto f(m)),$$

$m \in M$, $f \in \hom_D(M, D)$, is an isomorphism (of k-vector spaces).

Proposition 4.2.8. [MR00, Rot79] *Let D be a left Ore domain. Then we have the following implications among properties of finitely generated left D-modules:*

$$\textit{free} \Rightarrow \textit{projective} \Rightarrow \textit{reflexive} \Rightarrow \textit{torsion-free.}$$

Theorem 4.2.9. [MR00, Rot79] *Let k be a field.*

(a) Every finitely generated torsion-free left $A_1(k)$-module is projective.

(b) If D is a left principal ideal domain, then every finitely generated torsion-free left D-module is free (in particular, projective). This holds e.g. for $D = k[t]$ and $D = B_1 = k(t)[\partial]$.

(c) (Quillen-Suslin theorem) Every projective module over a commutative polynomial algebra over k is free.

Definition 4.2.10. An exact sequence of left D-modules

$$\cdots \xrightarrow{d_3} P_2 \xrightarrow{d_2} P_1 \xrightarrow{d_1} P_0 \xrightarrow{d_0} M \longrightarrow 0$$

is called a *projective resolution* of the left D-module M if P_i is projective for all i. If the P_i are free, then the above exact sequence is called a *free resolution* of M.

Remark 4.2.11. Let the finitely generated left D-module M be given as the cokernel of a homomorphism $\varphi : D^{1\times q} \to D^{1\times p}$ of free left D-modules. Hence, the complex

$$D^{1\times q} \xrightarrow{\varphi} D^{1\times p} \xrightarrow{\pi} M \longrightarrow 0$$

is exact. Let $R \in D^{q\times p}$ be the matrix representing φ with respect to the standard bases of $D^{1\times q}$ and $D^{1\times p}$. Then M can be considered as being *(finitely) presented by generators and relations*, i.e., M is generated by the images of the standard basis elements of $D^{1\times p}$ under the canonical projection π, and these images satisfy the left D-linear relations given by the rows of R (and all their consequences). Stated in yet another way we have $M \cong D^{1\times p}/D^{1\times q}R$. A free resolution of M

can be constructed as follows. Compute a finite generating set for the left D-linear relations satisfied by the rows of R. Write these relations as row vectors with entries in D again and define the matrix $S \in D^{r \times q}$ composed of these row vectors. The left D-module generated by the rows of S is called the *syzygy module* of R. Since it is the kernel of the homomorphism $(.R)$, it is finitely generated as a submodule of the free module $D^{1 \times q}$ over the left Noetherian algebra D. By construction the complex

$$D^{1\times r} \xrightarrow{.S} D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{\pi} M \longrightarrow 0$$

is exact. An iterated computation of syzygy modules then yields a free resolution of M.

Upper bounds on the lengths of free resolutions depend on the ring D; e.g. for the commutative polynomial algebra $D = k[x_1, \ldots, x_n]$ over k, Hilbert's Syzygy Theorem states that every finitely generated module over D has a free resolution of length at most n. For the class of Ore algebras D dealt with in Section 2.5 finite generating sets for syzygy modules can be obtained from a Janet basis computation with respect to a monomial ordering defined in Remark 2.2.3 (c). The *left global dimension* $\operatorname{lgld}(D)$ of the Ore algebra D is defined as the supremum of the lengths of projective resolutions of left D-modules (which may be infinite). However, if the k-algebra A is a domain with finite left global dimension and $\sigma : A \to A$ is a k-algebra automorphism, then we have

$$\operatorname{lgld} A \leq \operatorname{lgld} A[\partial; \sigma, \delta] \leq \operatorname{lgld} A + 1,$$

which shows that for the Ore algebras D of interest for this thesis, free resolutions of finite length of finitely generated left D-modules exist. (For more details we refer to [MR00], [CQR05].)

We close this section by defining the *extension groups* $\operatorname{ext}^i_D(M, \mathcal{F})$.

Definition 4.2.12. Let D be a left Noetherian Ore algebra, M a finitely generated left D-module, $\mathcal{F}$ a left D-module and

$$\cdots \xrightarrow{d_2} P_1 \xrightarrow{d_1} P_0 \xrightarrow{d_0} M \longrightarrow 0$$

a projective resolution of M. For $i \in \mathbb{Z}_{\geq 0}$, the abelian groups $\operatorname{ext}^i_D(M, \mathcal{F})$ are defined as the defects of exactness of the cochain complex

$$\cdots \longleftarrow \hom_D(P_2, \mathcal{F}) \xleftarrow{d_2^*} \hom_D(P_1, \mathcal{F}) \xleftarrow{d_1^*} \hom_D(P_0, \mathcal{F}) \longleftarrow 0,$$

where the homomorphism d_i^*, $i \geq 1$, is defined by

$$d_i^*(\varphi) = \varphi \circ d_i, \qquad \varphi \in \hom_D(P_{i-1}, \mathcal{F}).$$

In other words, we have:

$$\left\{ \begin{array}{rcl} \mathrm{ext}^0_D(M, \mathcal{F}) & = & \ker d_1^* \ = \ \hom_D(M, \mathcal{F}), \\ \mathrm{ext}^i_D(M, \mathcal{F}) & = & \ker d_{i+1}^* / \operatorname{im} d_i^*, \qquad i \geq 1. \end{array} \right.$$

The next proposition shows that $\mathrm{ext}^i_D(M, \mathcal{F})$ is well-defined.

Proposition 4.2.13. *The abelian group $\mathrm{ext}^i_D(M, \mathcal{F})$, $i \in \mathbb{Z}_{\geq 0}$, only depends on M and $\mathcal{F}$ up to group isomorphism, i.e., we can choose any projective resolution of M to compute it.*

More details about extension groups are given in Section 5.2.

4.3 Injective Modules

In this section modules with a property which is in some sense dual to projectivity are considered more closely. In the context of linear systems over Ore algebras D, the notion of an injective module is important for the interpretation of elements of certain left D-modules as solutions of a given linear system. Together with the property of being a cogenerator for the category of left D-modules, which is explained in the following section, an injective left D-module $\mathcal{F}$ serves as a signal space sharing homological properties with the left D-module M which is associated with the given linear system. In particular, $\mathcal{F}$ is then thought of as containing enough solutions of the linear system. More details on the duality which is admitted by injective cogenerators is presented in the next section and Chapter 6. The notions of injective modules and cogenerators can be found in [Rot79] or [Bou80].

In what follows, we let D be a (not necessarily commutative) ring with 1. We start by giving the definition of an injective module.

Definition 4.3.1. A left D-module $\mathcal{F}$ is *injective* if it is a direct summand of each left D-module which contains it (up to isomorphism), more precisely, for every monomorphism $\iota : \mathcal{F} \to \mathcal{E}$ of left D-modules, there exists a homomorphism $\beta : \mathcal{E} \to \mathcal{F}$ such that $\beta \circ \iota = \mathrm{id}_{\mathcal{F}}$.

The defining property of an injective module is in some sense dual to the defining property of a projective module (cf. Def. 4.2.6).

Lemma 4.3.2. *A left D-module $\mathcal{F}$ is injective if and only if for every left D-module A and every submodule B of A, every homomorphism $\varphi : B \to \mathcal{F}$ can be extended to a homomorphism $\psi : A \to \mathcal{F}$.*

Proof. "$\Rightarrow$": We assume that $\mathcal{F}$ is injective. Let A be a left D-module, B a submodule of A and $\varphi : B \to \mathcal{F}$ a homomorphism of left D-modules. We define $N := \{(\varphi(b), -b) \in \mathcal{F} \oplus A \mid b \in B\}$ which is a submodule of $\mathcal{F} \oplus A$. Then we consider the homomorphism from $\mathcal{F}$ to $\mathcal{E} := (\mathcal{F} \oplus A)/N$

$$\iota : \mathcal{F} \to \mathcal{E} : f \mapsto (f, 0) + N.$$

This homomorphism is injective because

$$\iota(f) = 0 \quad\Longleftrightarrow\quad 0 = \varphi(0) = f, \qquad f \in \mathcal{F}.$$

Since $\mathcal{F}$ is injective, there exists a homomorphism $\beta : \mathcal{E} \to \mathcal{F}$ such that $\beta \circ \iota = \mathrm{id}_{\mathcal{F}}$. Then we define the homomorphism

$$\psi : A \to \mathcal{F} : a \mapsto \beta((0, a) + N)$$

and show that ψ extends φ. Let $b \in B$. Then $(0, b) + N = (\varphi(b), 0) + N$. Hence,

$$\psi(b) = \beta((0, b) + N) = \beta(\iota(\varphi(b))) = \varphi(b).$$

"$\Leftarrow$": We assume that for every left D-module A and every submodule B of A, every homomorphism $\varphi : B \to \mathcal{F}$ can be extended to a homomorphism $\psi : A \to \mathcal{F}$. Suppose that $\mathcal{F}$ is a submodule of a left D-module $\mathcal{E}$. Then $\varphi := \mathrm{id}_{\mathcal{F}}$ can be extended to a homomorphism $\beta : \mathcal{E} \to \mathcal{F}$. This homomorphism satisfies $\beta \circ \iota = \mathrm{id}_{\mathcal{F}}$. $\square$

In fact, for a proof that a left D-module $\mathcal{F}$ is injective it is sufficient to verify the property stated in Lemma 4.3.2 for every left ideal I of D as the following lemma shows [Rot79].

Lemma 4.3.3 (Baer's criterion)**.** *A left D-module $\mathcal{F}$ is injective if and only if for every left ideal I of D every homomorphism $\varphi : I \to \mathcal{F}$ can be extended to a homomorphism $\psi : D \to \mathcal{F}$.*

Proof. "$\Rightarrow$" is a particular case of Lemma 4.3.2.

"$\Leftarrow$": We assume that for every left ideal I of D every homomorphism $\varphi : I \to \mathcal{F}$ can be extended to a homomorphism $\psi : D \to \mathcal{F}$. Suppose that $\mathcal{F}$ is a submodule of a left D-module $\mathcal{E}$. Then we show that for the canonical injection $\iota : \mathcal{F} \to \mathcal{E}$ there exists a homomorphism $\beta : \mathcal{E} \to \mathcal{F}$ such that $\beta \circ \iota = \mathrm{id}_{\mathcal{F}}$. To this end we are going to "approach" β by homomorphisms $\alpha : \mathcal{G} \to \mathcal{F}$, where $\mathcal{F} \leq \mathcal{G} \leq \mathcal{E}$.

$$\begin{tikzcd} 0 \arrow[r] & \mathcal{F} \arrow[r, "\iota"] \arrow[d, "\mathrm{id}_{\mathcal{F}}"'] & \mathcal{E} \arrow[dl, dashed, "\beta"] \\ & \mathcal{F} & \end{tikzcd}$$

We consider the set

$$S := \{ (\mathcal{G}, \alpha) \mid \mathcal{F} \leq \mathcal{G} \leq \mathcal{E},\ \alpha \in \hom_D(\mathcal{G}, \mathcal{F}),\ \alpha|_{\mathcal{F}} = \mathrm{id}_{\mathcal{F}} \}$$

which is partially ordered by

$$(\mathcal{G}_1, \alpha_1) \leq (\mathcal{G}_2, \alpha_2) \qquad \Longleftrightarrow \qquad \mathcal{G}_1 \leq \mathcal{G}_2 \quad \text{and} \quad \alpha_2|_{\mathcal{G}_1} = \alpha_1.$$

Obviously, for every non-empty totally ordered subset T of S there exists an element $(\mathcal{U}, \upsilon) \in T$ such that $(\mathcal{G}, \alpha) \leq (\mathcal{U}, \upsilon)$ for all $(\mathcal{G}, \alpha) \in T$, which means that T is inductively ordered (see [Lan93]). By Zorn's Lemma there exists a maximal element $(\mathcal{M}, \beta)$ in S. We show that $\mathcal{M} = \mathcal{E}$. Then $\beta : \mathcal{E} \to \mathcal{F}$ satisfies $\beta \circ \iota = \mathrm{id}_{\mathcal{F}}$.

Let us assume that $\mathcal{M} \lneqq \mathcal{E}$. Then there exists $e \in \mathcal{E} - \mathcal{M}$. We define the left ideal $I := \{d \in D \mid d\, e \in \mathcal{M}\}$ of D and the homomorphism

$$\varphi : I \to \mathcal{F} : a \mapsto \beta(a\, e).$$

Our assumption applies to the submodule I of the left D-module D, so that there exists a homomorphism $\psi : D \to \mathcal{F}$ which extends φ. Now we define the homomorphism

$$\gamma : \mathcal{M} + D\, e \to \mathcal{F} : m + d\, e \mapsto \beta(m) + \psi(d)$$

and show that $(\mathcal{M} + D\, e, \gamma) \in S$. First of all, γ is well-defined because for $m, \widetilde{m} \in \mathcal{M}$, $d, \widetilde{d} \in D$ satisfying $m + d\, e = \widetilde{m} + \widetilde{d}\, e$ we have

$$m - \widetilde{m} = (\widetilde{d} - d)\, e \in \mathcal{M}$$

and hence $\widetilde{d} - d \in I$, and

$$\begin{aligned} (\beta(m) + \psi(d)) - (\beta(\widetilde{m}) + \psi(\widetilde{d})) &= \beta(m - \widetilde{m}) + \psi(d - \widetilde{d}) \\ &= \beta((\widetilde{d} - d)\, e) - \psi(\widetilde{d} - d) \\ &= \varphi(\widetilde{d} - d) - \psi(\widetilde{d} - d) \\ &= 0. \end{aligned}$$

Secondly, γ extends $\mathrm{id}_{\mathcal{F}}$ because

$$\gamma(f) = \beta(f) + \psi(0) = f \qquad \text{for all} \quad f \in \mathcal{F}.$$

Therefore, we have $(\mathcal{M} + D\, e, \gamma) \in S$, which is a contradiction to the maximality of $\mathcal{M}$. □

The next proposition will be used in Chapter 6, where the problem of parametrizing the solution spaces of linear systems is investigated. There the injectivity of the signal space is needed to establish a duality between the left D-module which is associated with the system and the solution space.

Proposition 4.3.4. *A left D-module $\mathcal{F}$ is injective if and only if for each exact sequence of left D-modules*

$$0 \longrightarrow M_0 \xrightarrow{d_0} M_1 \xrightarrow{d_1} M_2 \xrightarrow{d_2} \cdots \xrightarrow{d_{n-1}} M_n \longrightarrow 0 \tag{4.5}$$

the cochain complex

$$0 \longleftarrow \hom_D(M_0,\mathcal{F}) \xleftarrow{d_0^*} \hom_D(M_1,\mathcal{F}) \xleftarrow{d_1^*} \hom_D(M_2,\mathcal{F}) \xleftarrow{d_2^*} \cdots \xleftarrow{d_{n-1}^*} \hom_D(M_n,\mathcal{F}) \longleftarrow 0 \tag{4.6}$$

is exact, where $d_i^* : \hom_D(M_{i+1},\mathcal{F}) \to \hom_D(M_i,\mathcal{F})$, $\varphi \mapsto \varphi \circ d_i$, $i = 0,\dots,n-1$.

Proof. "$\Leftarrow$": We apply Lemma 4.3.2 in order to prove that $\mathcal{F}$ is injective. Let A be a left D-module and B a submodule of A. Then we have the exact sequence

$$0 \longrightarrow B \xrightarrow{\iota} A \xrightarrow{\rho} A/B \longrightarrow 0,$$

where ι is the canonical injection and ρ is the canonical projection. By assumption, the cocomplex

$$0 \longleftarrow \hom_D(B,\mathcal{F}) \xleftarrow{\iota^*} \hom_D(A,\mathcal{F}) \xleftarrow{\rho^*} \hom_D(A/B,\mathcal{F}) \longleftarrow 0$$

is exact. Let $\varphi \in \hom_D(B,\mathcal{F})$ be arbitrary. Since ι^* is surjective (see Rem. 4.2.2 (b)), there exists $\psi \in \hom_D(A,\mathcal{F})$ such that $\iota^*(\psi) = \varphi$, i.e. $\psi|_B = \varphi$.

"$\Rightarrow$": We assume that $\mathcal{F}$ is injective. The exactness of (4.6) is shown in three steps. First we show that d_{n-1}^* is injective (see Rem. 4.2.2 (a)). Let $\varphi \in \ker d_{n-1}^*$, i.e. $\varphi \circ d_{n-1} = 0$. Let $m \in M_n$ be arbitrary. Since d_{n-1} is surjective (see Rem. 4.2.2 (b)), there exists $m' \in M_{n-1}$ such that $m = d_{n-1}(m')$. It follows

$$\varphi(m) = \varphi(d_{n-1}(m')) = 0.$$

Therefore, $\varphi = 0$, and we have shown that d_{n-1}^* is injective.

Next d_0^* is proven to be surjective. Let $\varphi \in \hom_D(M_0,\mathcal{F})$ be arbitrary. Via d_0 we consider M_0 as a submodule of M_1.

$$\begin{tikzcd} 0 \arrow[r] & M_0 \arrow[r, "d_0"] \arrow[d, "\varphi"'] & M_1 \arrow[dl, dashed, "\psi"] \\ & \mathcal{F} & \end{tikzcd}$$

Lemma 4.3.2 for $A = M_1$, $B = M_0$ implies that there exists an extension $\psi \in \hom_D(M_1,\mathcal{F})$ of φ. Then we have $d_0^*(\psi) = \varphi$.

Finally, we show that $\ker d_i^* = \operatorname{im} d_{i+1}^*$ for all $i = 0, \ldots, n-2$. Since (4.5) is a complex, we have

$$d_i^*(d_{i+1}^*(\varphi)) = \varphi \circ (d_{i+1} \circ d_i) = 0,$$

which means $\ker d_i^* \supseteq \operatorname{im} d_{i+1}^*$. Let $\varphi \in \ker d_i^*$, i.e. $\varphi(\operatorname{im} d_i) = \{0\}$.

$$\begin{tikzcd} M_i \arrow[r, "d_i"] & M_{i+1} \arrow[r, "d_{i+1}"] \arrow[d, "\varphi"'] & M_{i+2} \arrow[dl, dashed, "\psi"] \\ & \mathcal{F} & \end{tikzcd}$$

Since $\operatorname{im} d_i \subseteq \ker \varphi$, there is an induced homomorphism

$$\overline{\varphi} : M_{i+1}/\operatorname{im} d_i \to \mathcal{F} : m + \operatorname{im} d_i \mapsto \varphi(m).$$

Due to the exactness of (4.5) and by the first isomorphism theorem we have

$$M_{i+1}/\operatorname{im} d_i = M_{i+1}/\ker d_{i+1} \cong \operatorname{im} d_{i+1} \leq M_{i+2},$$

so that we can consider $\overline{\varphi}$ as a homomorphism $\operatorname{im} d_{i+1} \to \mathcal{F}$. Now Lemma 4.3.2, applied to $A = M_{i+2}$, $B = \operatorname{im} d_{i+1}$, shows that there exists an extension $\psi \in \hom_D(M_{i+2}, \mathcal{F})$ of $\overline{\varphi}$. For every $m \in M_{i+1}$ the equations

$$\psi(d_{i+1}(m)) = \overline{\varphi}(m + \operatorname{im} d_i) = \varphi(m)$$

hold, which means $d_{i+1}^*(\psi) = \varphi$. Altogether we have $\ker d_i^* = \operatorname{im} d_{i+1}^*$. $\square$

We give an example which demonstrates the immediate benefit of an injective signal space for the study of linear systems.

Example 4.3.5. Let k be a field, $D = k[\partial_{x_1}, \partial_{x_2}]$ be the commutative polynomial algebra over k and $\mathcal{F}$ a k-vector space of functions. We assume that ∂_{x_1}, ∂_{x_2} act on $\mathcal{F}$ by partial differentiation with respect to x_1 resp. x_2, which turns $\mathcal{F}$ into a D-module. In this example we consider the inhomogeneous linear system

$$(4.7) \quad R_1\,\eta = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \qquad \text{where} \quad R_1 := \begin{pmatrix} \partial_{x_1} \\ \partial_{x_2} \end{pmatrix} \in D^{2\times 1} \quad \text{and} \quad u_1, u_2 \in \mathcal{F}.$$

Obviously, one has the compatibility condition

$$(4.8) \qquad R_2 \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = 0, \qquad \text{where} \quad R_2 = (\partial_{x_2} \quad -\partial_{x_1}) \in D^{1\times 2}.$$

The D-module $M = D/D^{1\times 2}\,R_1$ has the following free resolution (see Def. 4.2.10):

$$0 \longrightarrow D \xrightarrow{.R_2} D^{1\times 2} \xrightarrow{.R_1} D \longrightarrow M \longrightarrow 0.$$

By applying $\hom_D(\,\cdot\,,\mathcal{F})$ to each of these modules we obtain the cocomplex

$$0 \longleftarrow \mathcal{F} \xleftarrow{(R_2).} \mathcal{F}^{2\times 1} \xleftarrow{(R_1).} \mathcal{F} \longleftarrow \hom_D(M,\mathcal{F}) \longleftarrow 0.$$

If $\mathcal{F}$ is an injective D-module, then this complex is exact. In this case we find, as in Prop. 4.1.8, that the solution set $\mathrm{Sol}_{\mathcal{F}}(R_1)$ (being the kernel of the homomorphism $(R_1).$) is isomorphic to $\hom_D(M,\mathcal{F})$, i.e. Malgrange's isomorphism. The exactness at $\mathcal{F}^{2\times 1}$ means that the inhomogeneous system (4.7) is solvable if and only if the compatibility condition (4.8) is fulfilled.

We recall that the algebra $k[[x_1,\ldots,x_n]]$ of formal power series in x_1, …, x_n over the field k can be viewed as the dual vector space of the polynomial algebra $k[x_1,\ldots,x_n]$ over k, i.e. $k[x_1,\ldots,x_n]^* \cong k[[x_1,\ldots,x_n]]$ as k-vector spaces. Now we are going to consider the same construction for an Ore algebra.

Remark 4.3.6. Let $D := k[x_1,\ldots,x_n][\partial_1;\sigma_1,\delta_1]\ldots[\partial_m;\sigma_m,\delta_m]$ be an Ore algebra and set

$$\mathcal{F} := \hom_k(D,k),$$

i.e., $\mathcal{F}$ is the k-vector space of all homomorphisms from the k-vector space D to k. Then $\mathcal{F}$ is a left D-module in virtue of

$$D\times\mathcal{F}\to\mathcal{F} : (d,\lambda)\mapsto (a\mapsto \lambda(a\,d)).$$

The properties of this left D-action are easily checked. For all $d_1,d_2\in D$, $\lambda\in\mathcal{F}$ we have:

$$\begin{aligned} 1\cdot\lambda &= (a\mapsto\lambda(a\cdot 1)) = \lambda,\\ d_2\cdot(d_1\cdot\lambda) &= d_2\cdot(a\mapsto\lambda(a\,d_1)) = (a\mapsto\lambda(a\,d_2\,d_1)) = (d_2\,d_1)\cdot\lambda. \end{aligned}$$

In fact, we have a pairing of D and $\mathcal{F}$, i.e. a k-bilinear form

$$(\;,\;) : D\times\mathcal{F}\to k : (d,\lambda)\mapsto\lambda(d) \tag{4.9}$$

which is non-degenerate in both arguments. With respect to this pairing D and $\mathcal{F}$ can be considered as dual to each other. Moreover, the linear map defined by right multiplication in D by a fixed element $d\in D$ and the linear map given by left multiplication in $\mathcal{F}$ by the same element d are adjoint to each other:

$$(a\cdot d,\lambda) = \lambda(a\cdot d) = (d\cdot\lambda)(a) = (a, d\cdot\lambda), \qquad a\in D, \quad \lambda\in\mathcal{F}. \tag{4.10}$$

Since every homomorphism $\lambda\in\mathcal{F}=\hom_k(D,k)$ is uniquely determined by its values for the elements of the k-basis $(x^\alpha\,\partial^\beta \mid \alpha\in(\mathbb{Z}_{\geq 0})^n, \beta\in(\mathbb{Z}_{\geq 0})^m)$ of D, it has a unique representation as formal power series

$$\sum_{\alpha\in(\mathbb{Z}_{\geq 0})^n,\,\beta\in(\mathbb{Z}_{\geq 0})^m} (x^\alpha\,\partial^\beta,\lambda)\,x^\alpha\,\partial^\beta. \tag{4.11}$$

Due to (4.10), for every $d \in D$ the representation of $d \cdot \lambda$ can be obtained as

$$\sum_{\alpha \in (\mathbb{Z}_{\geq 0})^n,\, \beta \in (\mathbb{Z}_{\geq 0})^m} (x^\alpha\, \partial^\beta, d \cdot \lambda)\, x^\alpha\, \partial^\beta = \sum_{\alpha \in (\mathbb{Z}_{\geq 0})^n,\, \beta \in (\mathbb{Z}_{\geq 0})^m} (x^\alpha\, \partial^\beta \cdot d, \lambda)\, x^\alpha\, \partial^\beta. \tag{4.12}$$

The common case of formal power series over a field k is exemplified below in Remark 4.5.1.

In the following corollary we will apply Janet's algorithm for the class of Ore algebras D treated in Section 2.5 in order to prove that $\hom_k(D, k)$ is an injective left D-module. The crucial property is that the set of monomials of D is partitioned by Janet's algorithm into a set of monomials which are *parametric* for the determination of a linear form $\lambda \in \hom_k(D, k)$ being orthogonal with respect to the pairing in (4.9) to all elements of a given left ideal I of D, and a set of monomials whose λ-values result from the choices of the values for the parametric ones. Note that Janet's algorithm provides both generalized Hilbert series enumerating the respective sets of monomials (see Section 2.3).

Corollary 4.3.7. *Let $D = k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_m; \sigma_m, \delta_m]$ be an Ore algebra as in Section 2.3 and let $\mathcal{F} := \hom_k(D, k)$ be the left D-module defined in Remark 4.3.6. Then $\mathcal{F}$ is injective.*

Proof. We apply Baer's criterion (see Lemma 4.3.3). Let I be a left ideal of D and $\varphi : I \to \mathcal{F}$ a homomorphism of left D-modules. Then φ defines a system of linear equations with coefficients in D for an unknown $\lambda \in \mathcal{F}$:

$$d \cdot \lambda = \varphi(d), \quad d \in I. \tag{4.13}$$

It is easily seen from (4.11) and (4.12) that this system is equivalent to a system of linear equations with coefficients in k for the unknowns $(x^\alpha\, \partial^\beta, \lambda)$, $\alpha \in (\mathbb{Z}_{\geq 0})^n$, $\beta \in (\mathbb{Z}_{\geq 0})^m$.

Since D is left Noetherian, the left ideal I is finitely generated. Let J be a Janet basis for I with respect to a fixed monomial ordering (see Def. 2.2.11 and Section 2.5). Since $\bigcup_{(g,\mu) \in J} \mathrm{Mon}(\mu) g$ is a k-basis of I (see Thm. 2.2.13), for every choice of values for the unknowns (m, λ) with monomials m in the complement of $\mathrm{lm}(I)$ in $\mathrm{Mon}(D)$ a solution of (4.13) is uniquely determined. More precisely, for every map

$$\mathrm{Mon}(D) - [\{\mathrm{lm}(p) \mid (p, \mu) \in J\}] \to k$$

the values for the unknowns in $\{(\mathrm{lm}(p), \lambda) \mid (p, \mu) \in J\}$ are uniquely determined by the equations which are exhibited by the Janet basis. Then we have specified values for all $(x^\alpha\, \partial^\beta, \lambda)$, $\alpha \in (\mathbb{Z}_{\geq 0})^n$, $\beta \in (\mathbb{Z}_{\geq 0})^m$, in such a way that the element $\lambda \in \mathcal{F}$ given by (4.11) is a solution of (4.13). Moreover, the homomorphism $\psi : D \to \mathcal{F}$ defined by $\psi(1) := \lambda$ extends φ. ☐

More examples of injective modules over certain Ore algebras will be given in Example 4.4.5 in the next section.

4.4 Injective Cogenerators

In this section we complement the homological properties of an injective left D-module with properties of a cogenerator for the category of left D-modules to be defined below. If an injective left D-module $\mathcal{F}$ is a cogenerator for the category of left D-modules, then the structural properties of the space of solutions in $\mathcal{F}$ of a linear system are reflected faithfully by the module which is associated with the linear system. For linear systems with constant coefficients, the connection between the module which is defined by the system equations and the solution space was examined in [Obe90] using these notions from homological algebra.

First we consider a (not necessarily commutative) ring D with 1.

Definition 4.4.1. Let D be a ring with 1.

(a) We denote by ${}_DM$ the category of left D-modules.

(b) A left D-module G is called a *generator* for ${}_DM$ if for every left D-module M we have:
$$\langle \operatorname{im} \varphi \mid \varphi \in \hom_D(G, M) \rangle = M.$$

(c) A left D-module C is called a *cogenerator* for ${}_DM$ if for every left D-module M we have:
$$\bigcap_{\varphi \in \hom_D(M,C)} \ker \varphi = \{0\}.$$

Remarks 4.4.2. Let D be a ring with 1.

(a) We show that the free left D-module D of rank 1 is a generator for ${}_DM$. Let M be any left D-module. Then for every $m \in M$ there is a homomorphism $D \to M$ defined by $1 \mapsto m$. Hence $\langle \operatorname{im} \varphi \mid \varphi \in \hom_D(D, M) \rangle \supseteq M$. The inverse inclusion is clear.

(b) A left D-module C is a cogenerator for ${}_DM$ if and only if for every left D-module M and every $0 \neq m \in M$ there exists $\varphi \in \hom_D(M, C)$ such that $\varphi(m) \neq 0$.

The following proposition is of crucial importance for the algebraic approach to linear systems because, if a left D-module $\mathcal{F}$ is a cogenerator for ${}_DM$, then the left D-module which is defined by the system equations reflects structural properties of the $\mathcal{F}$-solutions of the system.

Proposition 4.4.3. *Let $\mathcal{F}$ be a left D-module which is a cogenerator for ${}_DM$. Then for every complex*

$$0 \longrightarrow M_0 \xrightarrow{d_0} M_1 \xrightarrow{d_1} M_2 \xrightarrow{d_2} \cdots \xrightarrow{d_{n-1}} M_n \longrightarrow 0 \tag{4.14}$$

of left D-modules we have: if

$$
\begin{array}{c}
0 \longleftarrow \hom_D(M_0,\mathcal{F}) \xleftarrow{d_0^*} \hom_D(M_1,\mathcal{F}) \xleftarrow{d_1^*} \\
\hom_D(M_2,\mathcal{F}) \xleftarrow{d_2^*} \quad \cdots \quad \xleftarrow{d_{n-1}^*} \hom_D(M_n,\mathcal{F}) \longleftarrow 0
\end{array}
\tag{4.15}
$$

is an exact sequence, where d_i^*, is defined as in Prop. 4.3.4, $i = 0,\dots,n-1$, then (4.14) is exact.

Proof. Let us assume that (4.15) is exact. The exactness of (4.14) is shown in three steps. First we show that d_{n-1} is surjective (see Rem. 4.2.2 (b)). Since d_{n-1}^* is injective (see Rem. 4.2.2 (a)), we have

$$\ker d_{n-1}^* = \{\, \varphi \in \hom_D(M_n,\mathcal{F}) \mid \varphi(\operatorname{im} d_{n-1}) = \{0\}\,\} = \{0\}.$$

There is a one-to-one correspondence between the homomorphisms $M_n \to \mathcal{F}$ which map the elements of $\operatorname{im} d_{n-1}$ to zero and the homomorphisms $M_n/\operatorname{im} d_{n-1} \to \mathcal{F}$. Hence,

$$\hom_D(M_n/\operatorname{im} d_{n-1},\mathcal{F}) = \{0\}.$$

This implies

$$\bigcap_{\overline{\varphi}\in\hom_D(M_n/\operatorname{im} d_{n-1},\mathcal{F})} \ker\overline{\varphi} = M_n/\operatorname{im} d_{n-1},$$

and therefore $M_n/\operatorname{im} d_{n-1} = \{0\}$ because $\mathcal{F}$ is a cogenerator for ${}_DM$. We conclude that $\operatorname{im} d_{n-1} = M_n$.

Next we prove that d_0 is injective. Let $m \in \ker d_0$ be arbitrary. It follows $m \in \ker d_0^*(\varphi)$ for all $\varphi \in \hom_D(M_1,\mathcal{F})$. Since d_0^* is surjective, we have

$$\{\, d_0^*(\varphi) \mid \varphi \in \hom_D(M_1,\mathcal{F})\,\} = \hom_D(M_0,\mathcal{F}).$$

Therefore,

$$m \in \bigcap_{\varphi\in\hom_D(M_0,\mathcal{F})} \ker\varphi = \{0\}$$

because $\mathcal{F}$ is a cogenerator for ${}_DM$. Hence, d_0 is injective.

Finally, we show that $\ker d_{i+1} = \operatorname{im} d_i$ for all $i = 0,\dots,n-2$. The inclusion $\ker d_{i+1} \supseteq \operatorname{im} d_i$ is clear by the assumption that (4.14) is a complex.

We note once more that there is a one-to-one correspondence between the homomorphisms $\varphi : M_{i+1} \to \mathcal{F}$ whose kernel contains $\operatorname{im} d_i$ and the homomorphisms $\overline{\varphi} : M_{i+1}/\operatorname{im} d_i \to \mathcal{F}$. Since $\mathcal{F}$ is a cogenerator for ${}_DM$, we have

$$\bigcap_{\overline{\varphi}\in\hom_D(M_{i+1}/\operatorname{im} d_i,\mathcal{F})} \ker\overline{\varphi} = \{0\}. \tag{4.16}$$

The exactness of (4.15) implies

$$\{ \varphi \in \hom_D(M_{i+1}, \mathcal{F}) \mid \varphi(\operatorname{im} d_i) = \{0\} \} = \ker d_i^* = \operatorname{im} d_{i+1}^*.$$

Therefore, (4.16) shows that

$$\bigcap_{\varphi \in \operatorname{im} d_{i+1}^*} \ker \varphi = \operatorname{im} d_i.$$

Now it is easy to prove $\ker d_{i+1} \subseteq \operatorname{im} d_i$. Let $m \in \ker d_{i+1}$ be arbitrary. Then $\varphi(m) = 0$ for all $\varphi \in \operatorname{im} d_{i+1}^*$, which, due to the previous equation, implies $m \in \operatorname{im} d_i$. Thus we have shown that $\ker d_{i+1} \subseteq \operatorname{im} d_i$. □

We quote the following theorem about the existence of injective cogenerators. However, the constructive proof given in [Rot79] involves modules which are useless for effective computations.

Theorem 4.4.4. *For every ring D with 1 there exists an injective cogenerator $\mathcal{F}$ for the category ${}_D M$ of left D-modules.*

In the following examples we record the most important examples of signal spaces which are injective cogenerators for the category of left D-modules for a few Ore algebras D.

Examples 4.4.5. (a) Let $k \in \{\mathbb{R}, \mathbb{C}\}$ and $D = k[s_1, \ldots, s_r]$ the commutative polynomial algebra over k. Then the following D-modules are injective cogenerators for ${}_D M$ ([Mal63], [Ehr70], [Pal70], [Obe90, Thm. 2.54, Paragraphs 3 and 4]):

(1) the discrete signal space $k^{\mathbb{N}^r}$, whose elements can be represented by sequences $a = (a_n)_{n \in \mathbb{N}^r}$, on which D acts by shifts:

$$(s_i\, a)_n = a_{(n_1, \ldots, n_{i-1}, n_i+1, n_{i+1}, \ldots, n_r)}, \qquad n \in \mathbb{N}^r, \quad 1 \leq i \leq r;$$

$k^{\mathbb{N}^r}$ can also be viewed as the set of formal power series, in which case the action of D is given by left shifts of the coefficient sequences;

(2) the space of convergent power series in r variables with values in k as a submodule of $k^{\mathbb{N}^r}$;

(3) the space $C^\infty(\Omega, k)$ of k-valued smooth functions on an open and convex set $\Omega \subseteq \mathbb{R}^r$, where s_i acts by partial differentiation with respect to the i-th variable, $1 \leq i \leq r$;

(4) the space $\mathcal{D}'(\Omega, k)$ of k-valued distributions on an open and convex set $\Omega \subseteq \mathbb{R}^r$, where D acts again by partial differentiation.

(b) [Zer06] For the localized Weyl algebra $B_1(\mathbb{R})$ (see Ex. 2.4.5 (a), Ex. 4.1.1 (a)) the $\mathbb{R}$-valued functions on $\mathbb{R}$ which are smooth except in finitely many points form an injective cogenerator for ${}_{B_1(\mathbb{R})}M$.

(c) [FO98] Let Ω be an open interval in $\mathbb{R}$ and define the $\mathbb{C}$-algebra $\mathcal{R}(\Omega) := \{f/g \mid f, g \in \mathbb{C}[t],\ g(\lambda) \neq 0 \text{ for all } \lambda \in \Omega\}$. Moreover, let $D := \mathcal{R}[\partial]$ be the skew polynomial ring with commutation rule (cf. Def. 2.4.1)

$$\partial\, a = a\, \partial + \frac{da}{dt}, \qquad a \in \mathcal{R}.$$

Then the space of Sato's hyperfunctions on Ω is an injective cogenerator for ${}_DM$.

We are going to present an injective cogenerator $\mathcal{F}$ for the category of left D-modules, where D is an Ore algebra as in Section 2.5. To this end, the following lemma will be very useful to prove the cogenerator property. The isomorphism described in this lemma is in fact a particular part of a natural equivalence of functors, which is not needed here in the full generality. For more details we refer to [Rot79], [Bou80]. For the case of modules over commutative rings this natural equivalence was also applied in [Obe90] to prove the cogenerator property of certain modules.

Lemma 4.4.6. *Let D be a (not necessarily commutative) ring with 1 which is also a k-algebra and M a left D-module. Then we have*

$$\hom_D(M, \hom_k(D, k)) \cong \hom_k(M, k)$$

as k-vector spaces. Let $0 \neq m \in M$. Then there exists $\varphi \in \hom_D(M, \hom_k(D, k))$ satisfying $\varphi(m) \neq 0$ if and only if there exists $\psi \in \hom_k(M, k)$ which satisfies $\psi(m) \neq 0$.

Proof. First of all, it is verified in the same way as in Remark 4.3.6 that $\hom_k(D, k)$ is a left D-module with action

$$D \times \hom_k(D, k) \to \hom_k(D, k) : (d, \beta) \mapsto (a \mapsto \beta(a\, d)),$$

so that $\hom_D(M, \hom_k(D, k))$ is well-defined. Let us define

$$\begin{array}{rcl} \Phi : \hom_D(M, \hom_k(D, k)) & \longrightarrow & \hom_k(M, k) : \\ \alpha & \longmapsto & (m \mapsto \alpha(m)(1)) \end{array}$$

and

$$\begin{array}{rcl} \Psi : \hom_k(M, k) & \longrightarrow & \hom_D(M, \hom_k(D, k)) : \\ \beta & \longmapsto & (m \mapsto (d \mapsto \beta(d\, m))). \end{array}$$

Then Φ and Ψ are k-linear maps which are inverse to each other. The map Φ is well-defined because every homomorphism of left D-modules is in particular a k-linear map. In order to show that Ψ is well-defined we have to verify that for every $\beta \in \hom_k(M, k)$, $\alpha := \Psi(\beta)$ is a homomorphism of left D-modules. Let m_1, $m_2 \in M$ and d_1, $d_2 \in D$ be arbitrary. Then for all $d \in D$ and $i = 1, 2$ we have:

$$d_i\, \alpha(m_i) = (d \mapsto \alpha((d\, d_i)\, m_i)).$$

Therefore,

$$(d_1\, \alpha(m_1) + d_2\, \alpha(m_2))(d) = \alpha(d\, d_1\, m_1) + \alpha(d\, d_2\, m_2) = \alpha(d\,(d_1\, m_1 + d_2\, m_2)),$$

which shows that $\Psi(\beta)(d_1\, m_1 + d_2\, m_2) = d_1\, \Psi(\beta)(m_1) + d_2\, \Psi(\beta)(m_2)$. The fact that Φ and Ψ are k-linear is obvious. Finally, for all $\beta \in \hom_k(M, k)$ we have

$$\Phi(\Psi(\beta)) = (m \mapsto \beta(m))$$

and for all $\alpha \in \hom_D(M, \hom_k(M, k))$, $m \in M$,

$$\Psi(\Phi(\alpha))(m) = (d \mapsto \alpha(d\, m)(1)) = (d \mapsto d\, \alpha(m)(1)) = (d \mapsto \alpha(m)(1 \cdot d)).$$

Altogether it was proved that Φ and Ψ are isomorphisms which are inverse to each other.

Let us fix $0 \neq m \in M$. If $\varphi \in \hom_D(M, \hom_k(M, k))$ satisfies $\varphi(m) \neq 0$, then there exists $a \in D$ such that $\varphi(m)(a) \neq 0$. We define $\psi \in \hom_k(M, k)$ by $\psi(\widetilde{m}) := \varphi(\widetilde{m})(a)$, $\widetilde{m} \in M$. Then we have $\psi(m) \neq 0$. Conversely, if $\psi \in \hom_k(M, k)$ satisfies $\psi(m) \neq 0$, then for $\varphi \in \hom_D(M, \hom_k(M, k))$ defined by $\varphi(m) := (d \mapsto \psi(d\, m))$ we have $\varphi(m) \neq 0$ because $\varphi(m)(1) \neq 0$. □

Now we are well prepared to present an injective cogenerator $\mathcal{F}$ for the category of left D-modules, where D is an Ore algebra for which Janet bases are defined. However, in general, we can give no analytic interpretation for this injective cogenerator.

Theorem 4.4.7. *Let $D = k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_m; \sigma_m, \delta_m]$ be an Ore algebra as in Section 2.5 and let $\mathcal{F} := \hom_k(D, k)$. Then $\mathcal{F}$ is an injective cogenerator for ${}_D\mathcal{M}$.*

Proof. The fact that $\mathcal{F}$ is injective was shown in Corollary 4.3.7.

We prove that $\mathcal{F}$ is a cogenerator for ${}_D\mathcal{M}$ by showing that for every left D-module M and every $0 \neq m \in M$ there exists $\varphi \in \hom_D(M, \mathcal{F})$ such that $\varphi(m) \neq 0$ (see Remark 4.4.2 (b)). Let M be a left D-module and $0 \neq m \in M$ be arbitrary. By the axiom of choice there exists $\psi \in \hom_k(M, k)$ which satisfies $\psi(m) \neq 0$. Then, according to Lemma 4.4.6, there exists $\varphi \in \hom_D(M, \mathcal{F})$ such that $\varphi(m) \neq 0$. □

4.5 The Generalized Hilbert Series for Partial Differential Equations

In this section we explain the importance of the generalized Hilbert series for systems of linear partial differential equations.

Remark 4.5.1. Let $D = k[\partial_1, \ldots, \partial_n]$ be the commutative polynomial algebra over a field k of characteristic zero, and let $\mathcal{F} := k[[x_1, \ldots, x_n]] \cong \hom_k(D, k)$ be the k-algebra of formal power series. Then every $p \in \mathcal{F}$ can be represented uniquely as

$$\sum_{\alpha \in (\mathbb{Z}_{\geq 0})^n} c_\alpha \frac{x^\alpha}{\alpha!}, \qquad c_\alpha \in k, \qquad \alpha! := \alpha_1! \cdots \alpha_n!. \tag{4.17}$$

We let D act on $\mathcal{F}$ from the left by partial differentiation of the formal power series (4.17) and consider $(x^\alpha/\alpha! \mid \alpha \in (\mathbb{Z}_{\geq 0})^n)$ as the dual basis of $(\partial^\alpha \mid \alpha \in (\mathbb{Z}_{\geq 0})^n)$ with respect to the pairing (4.9), i.e.

$$(\partial^\beta, \textstyle\sum_{\alpha \in (\mathbb{Z}_{\geq 0})^n} c_\alpha \frac{x^\alpha}{\alpha!}) = c_\beta.$$

For a given system of linear partial differential equations with constant coefficients for one unknown function we compute a Janet basis J for the ideal of D which is generated by the left hand sides of these equations. By considering the differential equations as linear equations for $(\partial^\beta, \lambda)$, $\beta \in (\mathbb{Z}_{\geq 0})^n$, where $\lambda \in \mathcal{F}$ is a formal power series solution, Janet's algorithm partitions $\mathrm{Mon}(D)$ into a set of monomials m for which $(m, \lambda) \in k$ can be chosen arbitrarily and the set of monomials $S := \{\mathrm{lm}(p) \mid (p, \mu) \in J\}$ for which $(\mathrm{lm}(p), \lambda) \in k$ is uniquely determined by these choices. The set S is the $\mathrm{Mon}(D)$-multiple closed set generated by the leading monomials of the polynomials in the Janet basis J. In particular, the k-dimension of the space of formal power series solutions can be computed as the number of monomials in the complement C of S in $\mathrm{Mon}(D)$. In fact, the generalized Hilbert series $H_C(\partial_1, \ldots, \partial_n)$ of C enumerates a basis for the Taylor coefficients $(\partial^\beta, \lambda)$ of λ whose values can be assigned freely.

In this context, M. Janet calls the monomials ∂^β in $\mathrm{Mon}(D) - S$ *parametric derivatives* because the corresponding Taylor coefficients $(\partial^\beta, \lambda)$ of a formal power series solution λ can be chosen arbitrarily. The monomials in S are called *principal derivatives*. The Taylor coefficients $(\partial^\beta, \lambda)$ which correspond to principal derivatives ∂^β are uniquely determined by k-linear equations in terms of the Taylor coefficients of parametric derivatives. Of course, the extension of this method to determine the formal power series solutions of a linear system of partial differential equations is extended to the case of more than one unknown function in a straightforward way. The Maple package `Janet` provides procedures `SolSeries` and `PolySol` which compute truncated formal power series solutions and polynomial solutions of linear systems of PDEs up to a given degree.

Remark 4.5.2. The previous remark also applies to linear systems of partial differential equations whose coefficients are rational functions in the independent variables $x_1, \ldots, x_n$, i.e. $D = k[\partial_1, \ldots, \partial_n]$ is replaced by $B_n(k) = k(x_1, \ldots, x_n)[\partial_1, \ldots, \partial_n]$ (see Ex. 2.4.5 (a)). Of course, in this case a formal power series solution is only well-defined if the left submodule M of $B_n(k)^q$ which represents the left hand sides of the equations is also a left submodule of $A[\partial_1, \ldots, \partial_n]^q$, where A is a k-subalgebra of $B_n(k)$ whose elements do not have a pole in $0 \in k^n$ and the Janet basis for M is computed within $A[\partial_1, \ldots, \partial_n]^q$. In other words, a formal power series solution is only well-defined if $0 \in k^n$ is not a zero of any denominator occurring in the course of Janet's algorithm.

The next example illustrates the generalized Hilbert series in very easy cases.

Example 4.5.3.

```
> with(Janet):
```

Let us consider an ordinary differential equation for one unknown function. We are interested in formal power series solutions of $\frac{d^2u}{dt^2} = 0$.

The independent variable is t, the dependent variable is u.

```
> ivar := [t]; dvar := [u];
```

$$ivar := [t]$$
$$dvar := [u]$$

The left hand side is entered in a list:

```
> L := [diff(u(t),t,t)];
```

$$L := [\frac{d^2}{dt^2}\,\mathrm{u}(t)]$$

The Janet basis of the left ideal I in the localized Weyl algebra $B_1(\mathbb{Q})$ generated by ∂^2 is trivially computed as follows:

```
> J := JanetBasis(L, ivar, dvar);
```

$$J := [[\frac{d^2}{dt^2}\,\mathrm{u}(t)],\ [t],\ [u]]$$

The parametric derivatives are given by the following list:

```
> ParamDeriv(ivar, dvar);
```

$$[\mathrm{u}(t),\ \frac{d}{dt}\,\mathrm{u}(t)]$$

In general, `ParamDeriv` returns the generalized Hilbert series of the complement of $\mathrm{lm}(I)$ in $\mathrm{Mon}(B_1(\mathbb{Q}))$ as described in Remark 2.3.3. However, in case this complement consists of finitely many elements only, the result is just the list of these elements.

Let us determine all truncations of formal power series solutions of $\frac{d^2u}{dt^2} = 0$ up to degree 2:

```
>  SolSeries(J, 2);
```

$$[\mathrm{u}(t) = \mathit{C1}_0 + \mathit{C1}_1\, t],\ [\mathit{C1}_0,\ \mathit{C1}_1]$$

In fact, all formal power series solutions are constants or polynomials of degree 1.

Now we assume that we are given the Janet basis which corresponds to the single equation $t\,\frac{d^2u}{dt^2} = 0$. Since the procedure `JanetBasis` divides by the leading coefficient, we have to force the `Janet` package to accept the following input as a Janet basis.

```
>  J := AssertJanetBasis([t*diff(u(t),t,t)], ivar, dvar);
```

$$J := [[t\,(\frac{d^2}{dt^2}\,\mathrm{u}(t))],\ [t],\ [u]]$$

Of course, the parametric derivatives are the same as above:

```
>  ParamDeriv(ivar, dvar);
```

$$[\mathrm{u}(t),\ \frac{d}{dt}\,\mathrm{u}(t)]$$

However, since the procedure `SolSeries` has to divide by t when deriving linear relations for the Taylor coefficients of a formal power series solution from the Janet basis, such formal power series solutions are not well-defined in this case.

```
>  SolSeries(J, 2);

Error, (in Janet/SolSeries) invalid point; solution of, t = 0
```

Let us now exemplify the generalized Hilbert series of the `JanetOre` package.

```
>  with(JanetOre):
```

We define the Weyl algebra $A_1(\mathbb{Q})$ by the variables t and D and the relation $D\,t = t\,D + 1$, which is abbreviated by "weyl(D,t)" when using `JanetOre`.

```
>  var := [D,t]; ops := [weyl(D,t)];
```

$$\mathit{var} := [\mathrm{D},\ t]$$

$$\mathit{ops} := [\mathrm{weyl}(\mathrm{D},\ t)]$$

The left hand side of $t\,\frac{d^2u}{dt^2} = 0$ is represented by the following polynomial in $A_1(\mathbb{Q})$:

```
> L := [t*D^2];
```

$$L := [t\,\mathrm{D}^2]$$

We compute a Janet basis of the left ideal I of $A_1(\mathbb{Q})$ which is generated by $t\,D^2$.

```
> JBasis(L, var, ops);
```

$$[t\,\mathrm{D}^2]$$

The generalized Hilbert series of the complement of I in $\mathrm{Mon}(A_1(\mathbb{Q}))$ is

```
> JFactorModuleBasis(var);
```

$$\frac{1}{1-t} + \frac{\mathrm{D}}{1-t} + \frac{\mathrm{D}^2}{1-\mathrm{D}}$$

However, it should be much more difficult to relate the generalized Hilbert series to a vector space basis of solutions of $t\,\frac{d^2u}{dt^2} = 0$, e.g. in the space of Sato's hyperfunctions (see Ex. 4.4.5 (c)).

4.6 The Bernstein Filtration

We close this chapter by relating the generalized Hilbert series to a well-known concept in the theory of algebraic D-modules [Cou95]: the Bernstein filtration. In this section we generalize the Bernstein filtration to the class of Ore algebras to which we adapted Janet's algorithm in Chapter 2. For a submodule of a finitely generated free module over such an Ore algebra which is endowed with the Bernstein filtration, the Hilbert series of the associated graded module is then easily obtained from the generalized Hilbert series which is provided by Janet's algorithm.

First we need to give the definitions of filtered resp. graded k-algebras and filtered resp. graded modules over filtered resp. graded k-algebras.

Definition 4.6.1. Let D be a (not necessarily commutative) k-algebra.

(a) A family $G = (G_i)_{i \in \mathbb{Z}_{\geq 0}}$ of k-subspaces of D is a *grading of D*, if

$$D = \bigoplus_{i \in \mathbb{Z}_{\geq 0}} G_i \qquad \text{and} \qquad G_i \cdot G_j \subseteq G_{i+j} \qquad \text{for all} \quad i, j \in \mathbb{Z}_{\geq 0}.$$

Then D is said to be *graded*.

(b) Let $G = (G_i)_{i\in\mathbb{Z}_{\geq 0}}$ be a grading of D and let M be a left D-module. A family $\Gamma = (\Gamma_i)_{i\in\mathbb{Z}_{\geq 0}}$ of k-subspaces of M is a *grading of M with respect to G* (or a *G-grading of M*), if

$$M = \bigoplus_{i\in\mathbb{Z}_{\geq 0}} \Gamma_i \qquad \text{and} \qquad G_i \cdot \Gamma_j \subseteq \Gamma_{i+j} \qquad \text{for all} \quad i, j \in \mathbb{Z}_{\geq 0}.$$

Often an additional assumption on Γ is made on which we also agree here, namely that $\dim_k \Gamma_i$ is finite for all $i \in \mathbb{Z}_{\geq 0}$. Then M is said to be *graded with respect to G*.

(c) Let $G = (G_i)_{i\in\mathbb{Z}_{\geq 0}}$ be a grading of D and let M be a left D-module with G-grading $\Gamma = (\Gamma_i)_{i\in\mathbb{Z}_{\geq 0}}$. The *Hilbert series* of M with respect to Γ is defined by

$$H_{M,\Gamma}(\lambda) := \sum_{i\in\mathbb{Z}_{\geq 0}} (\dim_k \Gamma_i)\, \lambda^i \in \mathbb{Z}[[\lambda]].$$

Definition 4.6.2. Let D be a (not necessarily commutative) k-algebra.

(a) A family $F = (F_i)_{i\in\mathbb{Z}_{\geq 0}}$ of k-subspaces of D is a *filtration of D*, if we have for all $i, j \in \mathbb{Z}_{\geq 0}$:

$$F_i \subseteq F_{i+1}, \qquad D = \bigcup_{i\in\mathbb{Z}_{\geq 0}} F_i, \qquad \text{and} \quad F_i \cdot F_j \subseteq F_{i+j}.$$

Then D is said to be *filtered*.

(b) Let $F = (F_i)_{i\in\mathbb{Z}_{\geq 0}}$ be a filtration of D and let M be a left D-module. A family $\Phi = (\Phi_i)_{i\in\mathbb{Z}_{\geq 0}}$ of k-subspaces of M is a *filtration of M with respect to F* (or an *F-filtration of M*), if it satisfies for all $i, j \in \mathbb{Z}_{\geq 0}$:

$$\Phi_i \subseteq \Phi_{i+1}, \qquad \bigcup_{i\in\mathbb{Z}_{\geq 0}} \Phi_i = M, \qquad \text{and} \quad F_i \cdot \Phi_j \subseteq \Phi_{i+j}.$$

Similarly to Def. 4.6.1 we make the additional assumption on Φ that $\dim_k \Phi_i$ is finite for all $i \in \mathbb{Z}_{\geq 0}$. Then M is said to be *filtered with respect to F*.

The next remark shows that every filtration leads to a grading.

Remark 4.6.3. Every filtration $F = (F_i)_{i\in\mathbb{Z}_{\geq 0}}$ of D defines a graded k-algebra

$$\operatorname{gr}_F D := \bigoplus_{i\in\mathbb{Z}_{\geq 0}} F_i/F_{i-1}, \qquad \text{where} \quad F_{-1} := \{0\}.$$

Similarly, if a left D-module M has a filtration $\Phi = (\Phi_i)_{i\in\mathbb{Z}_{\geq 0}}$ with respect to F, then

$$\operatorname{gr}_\Phi M := \bigoplus_{i\in\mathbb{Z}_{\geq 0}} \Phi_i/\Phi_{i-1}, \qquad \text{where} \quad \Phi_{-1} := \{0\},$$

is a left $\operatorname{gr}_F D$-module with F-grading $(\Phi_i/\Phi_{i-1})_{i\in\mathbb{Z}_{\geq 0}}$.

Definition 4.6.4. Let k be a field, $D = k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_m; \sigma_m, \delta_m]$ an Ore algebra and $q \in \mathbb{N}$. Let $0 \neq p \in D^q$ be given in its unique representation

$$p = \sum_{i=1}^{q} \sum_{\substack{a \in (\mathbb{Z}_{\geq 0})^n \\ b \in (\mathbb{Z}_{\geq 0})^m}} c_{i,a,b}\, x^a\, \partial^b\, e_i, \qquad c_{i,a,b} \in k,$$

with respect to the k-basis $\mathrm{Mon}(D^q)$ of D^q (see Remark 2.5.2). Then the *total degree of* p is defined by

$$\deg(p) := \max\{\, |a| + |b| \mid c_{i,a,b} \neq 0 \text{ for some } i \in \{1, \ldots, q\}\,\}.$$

Moreover, we set $\deg(0) := -\infty$.

Motivated by the Bernstein filtration of the Weyl algebras (see [Cou95]) we give the following definition of the Bernstein filtration of Ore algebras for which Janet bases are defined.

Definition 4.6.5. Let k be a field and $D = k[x_1, \ldots, x_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_m; \sigma_m, \delta_m]$ an Ore algebra with

$$\begin{array}{rcll} \sigma_i(x_j) & = & c_{ij} x_j + d_{ij}, & c_{ij} \in k - \{0\}, \quad d_{ij} \in k, \\ \deg(\delta_i(x_j)) & \leq & 1, & 1 \leq i \leq m, \quad 1 \leq j \leq n. \end{array}$$

Then the *Bernstein filtration* $B = (B_i)_{i \in \mathbb{Z}_{\geq 0}}$ of D is defined by

$$B_i := \{\, p \in D \mid \deg(p) \leq i\,\}, \quad i \in \mathbb{Z}_{\geq 0}.$$

Remark 4.6.6. Let D be an Ore algebra as in the previous definition. Then the graded k-algebra $\mathrm{gr}_B\, D$ with respect to the Bernstein filtration B of D is isomorphic to the commutative polynomial algebra over k in $n + m$ variables.

Remark 4.6.7. Let D be an Ore algebra as in Definition 4.6.5, $q \in \mathbb{N}$ and let M be a left submodule of D^q. We endow D with the Bernstein filtration B. Then M has a B-filtration $\Phi = (\Phi_i)_{i \in \mathbb{Z}_{\geq 0}}$ defined by

$$\Phi_i := \{\, p \in M \mid \deg(p) \leq i\,\}, \quad i \in \mathbb{Z}_{\geq 0}.$$

By Remark 4.6.3, $\mathrm{gr}_\Phi\, M$ is a graded $\mathrm{gr}_B\, D$-module, more precisely a graded $k[\xi_1, \ldots, \xi_n, \eta_1, \ldots, \eta_m]$-module with B-grading $\Gamma = (\Gamma_i)_{i \in \mathbb{Z}_{\geq 0}}$ given by

$$\Gamma_i := \Phi_i / \Phi_{i-1}, \quad i \in \mathbb{Z}_{\geq 0}, \qquad \text{where} \quad \Phi_{-1} := \{0\}.$$

Now we fix the a monomial ordering $<$ on $\mathrm{Mon}(D^q)$ which is compatible with the total degree, i.e. for all $m_1, m_2 \in \mathrm{Mon}(D^q)$ we have:

$$\deg(m_1) < \deg(m_2) \quad \Rightarrow \quad m_1 < m_2.$$

Let J be a Janet basis for M with respect to this monomial ordering. According to Theorem 2.2.13 (b), $\bigcup_{(g,\mu)\in J} \mathrm{Mon}(\mu)g$ is a k-basis of M. This k-basis is enumerated by the generalized Hilbert series

$$H_{\mathrm{lm}(M)}(x_1, \ldots, x_n, \partial_1, \ldots, \partial_m),$$

which is easily computed from J by using Remark 2.3.3. Since $<$ is compatible with the total degree,

$$\bigcup_{(g,\mu)\in J} \mathrm{Mon}(\mu)\,(\mathrm{lm}(g) + \Phi_{\deg(g)-1})$$

is a k-basis of $\mathrm{gr}_\Phi M$, so that the coefficient of λ^i in the formal power series $H_{\mathrm{lm}(M)}(\lambda, \ldots, \lambda)$ equals $\dim_k \Gamma_i$ for all $i \in \mathbb{Z}_{\geq 0}$. Therefore, we have

$$H_{\mathrm{lm}(M)}(\lambda, \ldots, \lambda) = H_{\mathrm{gr}_\Phi M, \Gamma}(\lambda).$$

The Hilbert series of the graded module $\mathrm{gr}_\Phi M$ is obtained from the generalized Hilbert series of $\mathrm{lm}(M)$ by substituting the indeterminate λ for all variables x_1, $\ldots$, x_n, ∂_1, $\ldots$, ∂_m.

Chapter 5

Algebraic Systems Theory

In this chapter we study structural properties of control systems. Let us first give a very short introduction to the most important system theoretic notions.

Systems theory comprises the study of systems in a general manner. However, for this thesis it is sufficient to think of a system as modelling certain natural phenomena, engineering or economic processes, mechanical systems, chemical processes etc. We confine ourselves to systems for which certain types of equations determine the relations among quantities of the system which are of interest. The quantities which are related by these equations are called the *system variables*. Some of them are considered to form the *state* of the system, which is supposed to represent the configuration of the system. It depends on the viewer's intention which variables are included in the state. Although a state is very convenient for the interpretation of the observed properties of the system, it will only play a minor role in what follows because the algebraic treatment of these systems does not depend on the agreement on a state.

Among the relevant systems are dynamical systems which are either described by ordinary differential equations in the continuous-time case or by difference equations in the discrete-time case. But the scope of the algebraic approach which is used in this thesis also admits, e.g. systems described by partial differential equations or differential time-delay systems (i.e. systems governed by retarded differential equations). However, only deterministic systems are considered here.

The types of systems are further classified, e.g. depending on whether the state consists of finitely many or infinitely many quantities, whether the equations describing the system are all linear or nonlinear, whether they have constant coefficients or not, etc.

One of the basic questions to ask about a given system is whether it is *controllable*, i.e. whether its state can be influenced by certain degrees of freedom. A system is completely controllable if these degrees of freedom can be manipulated in such a way as to make the state attain desired values. For a dynamical system described by ordinary differential equations, this property can be viewed as the possibility to connect any trajectory of the system which started in the past to any future trajectory which respects the system laws.

As a particular case of controllability, in most applications it is desired to *stabilize* the system, i.e. to steer it into a certain state of rest.

Another interesting property, which is in some sense dual to controllability, is *observability*. Let us suppose that a state is chosen and certain functions of the system variables are fixed which serve as *outputs*, which means that their actual values are available to us. Then the system is said to be observable if the values of the state variables can be recovered from these output variables alone.

The system theoretic philosophy behind the algebraic way of studying these systems presented in this chapter is the *behavioral approach* established by J. C. Willems [PW98], [Wil91]. For this approach a *signal space* is fixed which consists of all functions (e.g. depending on time in the case of ordinary differential equations) which are admissible candidates for the solutions of the system equations. As a subset of this signal space the set of solutions of these equations is determined. The set of solutions is the *behavior* of the system. For continuous-time dynamical systems, the behavior thus consists of all trajectories that are solutions of the ordinary differential equations for some initial conditions.

Depending on the type of the system under consideration and other aims, the signal space is chosen to be e.g. the vector space of smooth functions, the vector space of distributions, etc. As explained in the previous chapter, for the investigation of linear systems a signal space is preferably an injective cogenerator because the structural properties of the solution space can be studied by examining the module which is associated with the linear system.

A classification of certain structural properties of linear systems in terms of the properties of the associated module is recalled in Section 5.1. The most important references are [PQ98, PQ99b, PQ99a, Qua99, Woo00]. In Section 5.2 a straightforward but practicable method for computing presentations of extension groups $\mathrm{ext}^i_D(M, D)$ are given. Controllability is checked in Section 5.3 on two examples. The first is a simple mechanical system, the second one comes from chemical engineering. In Section 5.4 we give a detailed example of how autonomous observables of the linearized system can be lifted to autonomous observables of the given nonlinear one. Finally, a short survey of the `OreModules` package is given in Section 5.5.

For further details about systems theory in general and algebraic methods we refer to [PW98], [Son98], [BY83], [Zer00], [GL01]. For a detailed account on the algebraic approach which is pursued here see [Pom01].

5.1 Structural Properties of Linear Systems

In this section we give a very concise overview of the module-theoretic characterizations of structural properties of linear systems which are needed in the following sections and chapters. Figure 5.1 is essentially due to A. Quadrat and we refer to [Qua99] for the applications of homological algebra to systems theory.

Definition 5.1.1. [Woo00, CQR05] Let D be a left Noetherian Ore algebra, $R \in D^{q\times p}$, $\mathcal{F}$ a left D-module, and $\mathcal{B} = \{\, \eta \in \mathcal{F}^{p\times 1} \mid R\eta = 0 \,\}$ the behavior defined by R and $\mathcal{F}$.

(a) An *observable* of $\mathcal{B}$ is a left D-linear combination of the system variables η_i. An observable ψ is called *autonomous* if there exists $0 \neq P \in D$ such that $P\,\psi = 0$. An observable is said to be *free* if it is not autonomous.

(b) A behavior $\mathcal{B}$ is said to be *controllable* if every observable of $\mathcal{B}$ is free.

Remark 5.1.2. The notion of controllability defined in Def. 5.1.1 is independent of any choice of variables for the system; in particular, no agreement on a state of the system is needed.

For linear systems governed by ordinary differential equations, the equivalence of this notion of controllability and the understanding of controllability as the possibility to connect any past trajectory to any future trajectory was shown in [FG93].

More generally, in [PS98], parametrizability of the behaviors of linear systems of partial differential equations with constant coefficients in the sense of Chapter 6, which is in fact equivalent to controllability, was proved to be equivalent to the possibility to patch any two solutions of the system.

We quote the following important theorem from [CQR05]. The references given in the theorem point to work on particular classes of systems (i.e. for particular Ore algebras D).

Theorem 5.1.3. *Let D be a left Noetherian Ore algebra, $R \in D^{q\times p}$, and $\mathcal{F}$ an injective left D-module which is also a cogenerator for ${}_D M$. Moreover, let $M = D^{1\times p}/D^{1\times q}\,R$ be the left D-module which is associated with the linear system $R\,\eta = 0$ and $\mathcal{B} = \{\, \eta \in \mathcal{F}^{p\times 1} \mid R\,\eta = 0 \,\}$ its behavior.*

(a) [Pom95, Pom01, PQ99b, Woo00] *The observables of $\mathcal{B}$ are in one-to-one correspondence with the elements of the left D-module M.*

(b) [FM98, PS98, Pom95, Pom01, PQ99b, Woo00] *The autonomous observables of $\mathcal{B}$ are in one-to-one correspondence with the torsion elements of M.*

(c) [FM98, Mou95, PS98, Pom95, Pom01, PQ99b] *$\mathcal{B}$ is controllable if and only if M is torsion-free.*

Definition 5.1.4. Let D be a left Noetherian Ore algebra, $R \in D^{q\times p}$ and $M = D^{1\times p}/D^{1\times q}\,R$ a left D-module. The *transposed module* of M is the right D-module

$$M^\top = D^{q\times 1}/R\,D^{p\times 1}.$$

If D is a commutative ring, then we have $M^\top = D^{1\times q}/(D^{1\times p}\,R^T)$, which explains the terminology.

Of course, the module $M^\top$ depends on the presentation of M as the left D-module $D^{1\times p}/D^{1\times q}\,R$ which is not indicated in the notation. When the transposed module $M^\top$ of M will be used, M will always be given by a fixed presentation $M = D^{1\times p}/D^{1\times q}\,R$.

The next theorem is very important for the structural analysis of linear systems over Ore algebras D. It characterizes torsion-free, reflexive and projective left D-modules in terms of certain extension groups. This theorem will be applied many times in the subsequent sections and chapters. For the definition of extension groups we refer to Def. 4.2.12, p. 75.

Theorem 5.1.5. [PQ03] *Let D be a left and right Noetherian domain, $R \in D^{q\times p}$ and consider the left D-module M presented by*

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \longrightarrow M \longrightarrow 0.$$

Let $M^\top$ be the transposed module of M:

$$0 \longleftarrow M^\top \longleftarrow D^{q\times 1} \xleftarrow{R.} D^{p\times 1} \longleftarrow M^* \longleftarrow 0.$$

(a) We have

$$t(M) \cong \mathrm{ext}^1_D(M^\top, D).$$

In particular, M is torsion-free if and only if $\mathrm{ext}^1_D(M^\top, D) = 0$.

(b) M is reflexive if and only if $\mathrm{ext}^i_D(M^\top, D) = 0$ for $i = 1, 2$.

(c) M is projective if and only if $\mathrm{ext}^i_D(M^\top, D) = 0$ for all $i = 1, \ldots, \mathrm{rgld}(D)$, where the right global dimension $\mathrm{rgld}(D)$ of D is defined as the supremum of the lengths of projective resolutions of right D-modules [MR00].

Remark 5.1.6. The right global dimension $\mathrm{rgld}(k[x_1, \ldots, x_n])$ of the polynomial algebra over k equals n. If k has characteristic zero, then for the Weyl algebra $A_n(k)$ we also have $\mathrm{rgld}(A_n(k)) = n$. Hence, in order to check whether a finitely presented left $A_n(k)$-module is projective it is sufficient to compute n extension groups of the transposed module $M^\top$. For more details see [MR00], [CQR05].

The important characterization of structural properties of linear systems in terms of module properties and their counterparts in homological algebra are summarized in the next figure.

Figure 5.1: characterizing system/module properties

system	**module**	**homological algebra**
autonomous observables	$t(M) \neq 0$	$\mathrm{ext}^1_D(M^\top, D) \neq 0$
controllability, parametrizability	$t(M) = 0$	$\mathrm{ext}^1_D(M^\top, D) = 0$
parametrizability of the parametrization	reflexive	$\mathrm{ext}^i_D(M^\top, D) = 0$, $i = 1, 2$
...	...	...
chain of n parametrizations	projective	$\mathrm{ext}^i_D(M^\top, D) = 0$, $1 \leq i \leq n := \mathrm{rgld}(D)$
flatness	free	in general no criteria, but for a principal ideal domain: torsion-free $\iff$ free

5.2 Computation of $\mathrm{ext}^i_D(M, D)$

In the previous section a classification of structural properties of linear systems was recalled. First of all, the structural properties of the system under consideration correspond to properties of its associated module M. The latter properties can be checked by computing extension groups $\mathrm{ext}^i_D(M^\top, D)$ of the transposed module $M^\top$ of M (see Figure 5.1). In this section we describe a straightforward algorithm to compute a presentation of $\mathrm{ext}^i_D(M, D)$, $i \in \mathbb{Z}_{\geq 0}$, for a left Noetherian Ore algebra D. It assumes that generating sets of finitely presented left D-modules (in particular syzygy modules) can be constructed. For the class of Ore algebras considered in Section 2.5, Janet's algorithm lends itself to this purpose. The algorithm presented in this section will be applied in the subsequent sections and chapters to investigate linear systems using the module-theoretic approach.

First let D be a left Noetherian ring. We will confine ourselves to left Noetherian Ore algebras later on. We recall Definition 4.2.12 of $\mathrm{ext}^i_D(M, D)$, $i \in \mathbb{Z}_{\geq 0}$.

Definition 5.2.1. Let D be a left Noetherian Ore algebra, M a finitely generated left D-module, $\mathcal{F}$ a left D-module and

$$\cdots \xrightarrow{d_2} P_1 \xrightarrow{d_1} P_0 \xrightarrow{d_0} M \longrightarrow 0 \tag{5.1}$$

a projective resolution of M. For $i \in \mathbb{Z}_{\geq 0}$, the abelian groups $\mathrm{ext}^i_D(M, \mathcal{F})$ are defined as the defects of exactness of the cochain complex

$$\cdots \longleftarrow \hom_D(P_2, \mathcal{F}) \xleftarrow{d_2^*} \hom_D(P_1, \mathcal{F}) \xleftarrow{d_1^*} \hom_D(P_0, \mathcal{F}) \longleftarrow 0,$$

where the homomorphism d_i^*, $i \geq 1$, is defined by

$$d_i^*(\varphi) = \varphi \circ d_i, \qquad \varphi \in \hom_D(P_{i-1}, \mathcal{F}).$$

In other words, we have:

$$\begin{cases} \mathrm{ext}^0_D(M, \mathcal{F}) &= \ker d_1^* = \hom_D(M, \mathcal{F}), \\ \mathrm{ext}^i_D(M, \mathcal{F}) &= \ker d_{i+1}^* / \operatorname{im} d_i^*, \qquad i \geq 1. \end{cases}$$

In this section we consider the case $\mathcal{F} = D$. For a *left* D-module M the abelian group $\mathrm{ext}^i_D(M, D)$ is a *right* D-module. If effective methods to compute with (finitely generated) left D-modules are available, then it is favourable to deal exclusively with *left* D-modules by turning $\mathrm{ext}^i_D(M, D)$ into a left D-module. This can be achieved with an involution of the ring D which is defined next.

Definition 5.2.2. [Jac85] Let D be a (not necessarily commutative) ring with 1. An automorphism θ of the additive group of D which maps 1 to 1 and satisfies

$$\theta(a_1\, a_2) = \theta(a_2)\,\theta(a_1) \quad \text{for all} \quad a_1, a_2 \in D$$

is called an *anti-automorphism* of D. If moreover θ satisfies $\theta \circ \theta = \mathrm{id}$, then θ is called an *involution* of D.

Examples 5.2.3. (a) Let us consider the Weyl algebra in $2n$ indeterminates $A_n(k) = k[x_1, \ldots, x_n][\partial_1, \ldots, \partial_n]$ (see Ex. 2.4.5 (a), Ex. 4.1.1 (a)). The most common involution θ of $A_n(k)$ is defined by

$$x_i \longmapsto x_i, \qquad \partial_i \longmapsto -\partial_i, \qquad i = 1, \ldots, n.$$

This also defines an involution for $B_n(k)$ in a similar way.

(b) Let $D = S_h$ (see Ex. 2.4.5 (b), Ex. 4.1.1 (b)). Then an involution θ of D is defined by

$$t \longmapsto -t, \qquad \delta_h \longmapsto \delta_h.$$

Remark 5.2.4. Let D be a (not necessarily commutative) ring with 1 which has an involution θ, and let M be a right D-module. Then M can be considered as a left D-module by defining the action of D to be

$$r \cdot m := m \cdot \theta(r), \qquad r \in D, \quad m \in M.$$

Of course, the involution can be used in the same way to turn left D-modules into right D-modules.

The following definition introduces a short notation for the result of applying an involution θ to each entry of R^T for a given matrix $R \in D^{q \times p}$.

Definition 5.2.5. Let D be a (not necessarily commutative) ring with 1 which has an involution θ. For each matrix $R \in D^{q \times p}$ we define the matrix

$$\theta(R) := (\theta(R^T_{ij}))_{1 \le i \le p,\, 1 \le j \le q} \in D^{p \times q}.$$

The next remark shows that $\theta(R)$ is the correct "extension" of $\theta : D \to D$ to matrices.

Remark 5.2.6. In the situation of the previous definition we consider the product $R\,v$ of the matrix R and a column vector $v \in D^{p \times 1}$. It is related to $\theta(v)\,\theta(R)$ as follows:

$$\begin{aligned} \theta(v)\,\theta(R) &= \left(\textstyle\sum_{j=1}^{p} \theta(v)_j\, \theta(R)_{ji}\right)_{i=1,\ldots,q} \\ &= \left(\textstyle\sum_{j=1}^{p} \theta(v_j)\, \theta(R_{ij})\right)_{i=1,\ldots,q} \\ &= \left(\textstyle\sum_{j=1}^{p} \theta(R_{ij}\, v_j)\right)_{i=1,\ldots,q} \\ &= \theta(R\,v). \end{aligned}$$

The matrix R represents a homomorphism $D^{p\times 1} \to D^{q\times 1}$ between free right D-modules. Hence, when considering these modules as left D-modules, the corresponding homomorphism $D^{1\times p} \to D^{1\times q}$ is represented by $\theta(R)$.

Remark 5.2.7. Let D be a (not necessarily commutative) ring with 1 which has an involution θ. We consider a complex

$$\cdots \longrightarrow D^{1\times p_1} \xrightarrow{.R} D^{1\times p_2} \xrightarrow{.S} D^{1\times p_3} \longrightarrow \cdots$$

of free left D-modules, where $R \in D^{p_1\times p_2}$, $S \in D^{p_2\times p_3}$ are matrices with entries in D. We apply $\hom_D(\,\cdot\,, D)$ to this complex and use the isomorphisms $\hom_D(D^{1\times p}, D) \cong D^{p\times 1}$, $p \in \mathbb{N}$. This results in the following cocomplex of free right D-modules

$$\cdots \longleftarrow D^{p_1\times 1} \xleftarrow{R.} D^{p_2\times 1} \xleftarrow{S.} D^{p_3\times 1} \longleftarrow \cdots$$

According to our intention to consider left D-modules only, we want to turn the previous cocomplex of right D-modules into a cocomplex of left D-modules. In particular, homomorphisms between free modules in the new cocomplex are represented by matrices which are applied to row vectors. By using the involution θ of D and Remark 5.2.6 we obtain the following cocomplex of left D-modules

$$\cdots \longleftarrow D^{1\times p_1} \xleftarrow{.\theta(R)} D^{1\times p_2} \xleftarrow{.\theta(S)} D^{1\times p_3} \longleftarrow \cdots$$

Note that if D is commutative, then $\theta(R) = R^T$, $\theta(S) = S^T$.

In what follows, let D be a left Noetherian Ore algebra which has an involution. We assume that we have a procedure at our disposal which computes a finite generating set (e.g. a Janet basis) for the syzygy module of a finite generating set of a left D-module (see Remark 4.2.11, p. 74). The following algorithm is formulated along the lines of the definition of the extension groups $\operatorname{ext}^i_D(M, D)$, $i \in \mathbb{Z}_{\geq 0}$. It constructs a presentation of an extension group $\operatorname{ext}^i_D(M, D)$ for a given finitely presented left D-module M. It uses an involution θ of D which we fix once and for all, so that $\operatorname{ext}^i_D(M, D)$ is considered as left D-module.

Algorithm 5.2.8 (Ext)**.**

Input: (R, i), where $R \in D^{q\times p}$ defines $M = D^{1\times p}/D^{1\times q}R$, and $i \in \mathbb{Z}_{\geq 0}$
Output: (L_1, L_2), where $L_1 \in D^{r\times s}$, $L_2 \in D^{t\times s}$ such that

$$\operatorname{ext}^i_D(M, D) \cong D^{1\times r}L_1/D^{1\times t}L_2$$

Algorithm:

1: Compute the first $i+1$ homomorphisms between free modules in a free resolution of M, i.e. set $S_1 := R$ and find matrices $S_2 \in D^{m_2 \times m_1}$, ..., $S_{i+1} \in D^{m_{i+1} \times m_i}$, where $m_1 := q$, $m_0 := p$, such that the complex

$$D^{1\times m_{i+1}} \xrightarrow{.S_{i+1}} D^{1\times m_i} \xrightarrow{.S_i} \ldots \xrightarrow{.S_3} D^{1\times m_2} \xrightarrow{.S_2} D^{1\times q} \xrightarrow{.S_1} D^{1\times p} \longrightarrow M \longrightarrow 0$$

is exact (iterative computation of syzygy modules)

2: Compute a finite generating set of the syzygy module of the left D-module generated by the rows of $\theta(S_{i+1})$, i.e. find a matrix $K \in D^{r\times m_i}$ such that

$$D^{1\times m_{i+1}} \xleftarrow{.\theta(S_{i+1})} D^{1\times m_i} \xleftarrow{.K} D^{1\times r}$$

is an exact sequence of left D-modules

3: **if** $i = 0$ **then**

4: **return** $(K, 0 \in D^{1\times p})$

5: **else**

6: **return** $(K, \theta(S_i))$

7: **fi**

Remark 5.2.9. Comparing the beginning of a free resolution of M constructed in Algorithm 5.2.8 with the resolution (5.1) (where we assume the P_i to be free) we have $d_i = (.S_i)$ for all $i \geq 1$. Therefore, Algorithm 5.2.8 presents $\operatorname{ext}^i_D(M, D)$ as $\ker d_{i+1} / \operatorname{im} d_i$.

Algorithm 5.2.8 is derived from the definition of the extension groups in a straightforward way. We included the description of this algorithm to clarify the role of left D-modules and the involution θ of D. This algorithm will be applied in the next sections and many times in Chapter 7.

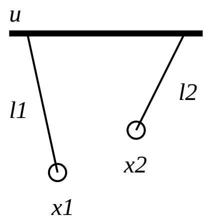
5.3 Controllability

The algorithm described in the previous section, which computes a presentation of an extension group $\operatorname{ext}^1_D(M, D)$, is applied in this section in order to check controllability of two linear systems. First we recall the characterization of controllability by means of $\operatorname{ext}^1_D(M^\top, D) \cong t(M)$, where M is the module associated with the linear system (see Section 5.1). Then we demonstrate the controllability check on a mechanical system and a chemical engineering example.

Remark 5.3.1. By definition, a behavior $\mathcal{B}$ is controllable if and only if every observable of $\mathcal{B}$ is free, i.e. not autonomous (see Section 5.1). Theorem 5.1.3 states that the autonomous observables of $\mathcal{B}$ are in one-to-one correspondence with the torsion elements of the module M which is associated with the linear system. Moreover, torsion-freeness of M is characterized by $\mathrm{ext}^1_D(M^\top, D) = 0$ (see Theorem 5.1.5, Figure 5.1).

The first example of this section can be found in [Pom01, Example V.I.1]. We demonstrate how controllability of a mechanical system can be checked using symbolic computation only. Janet's algorithm (see Chapter 2) is used to compute presentations of extension groups $\mathrm{ext}^1_D(M, D)$. In particular, the next example illustrates the dependence of controllability on configurations of parameters which occur in the equations describing the system (see also Chapter 7).

Example 5.3.2. Let us consider a bipendulum, more precisely a bar on which two pendula of certain lengths $l1$ resp. $l2$ are fixed.



The bar is movable horizontally. Its horizontal position is denoted by u. The horizontal positions of the end points of the two pendula are $x1$ resp. $x2$. Then u, $x1$, $x2$ fulfill the ordinary differential equations

$$(5.2)\qquad \left\{ \begin{array}{rcl} \dfrac{d^2x1}{dt^2} + \dfrac{g}{l1}\,x1 - \dfrac{g}{l1}\,u & = & 0, \\[2ex] \dfrac{d^2x2}{dt^2} + \dfrac{g}{l2}\,x2 - \dfrac{g}{l2}\,u & = & 0, \end{array} \right.$$

where g is the gravitational constant[1].

```
> with(Janet):
```

For the equations which describe the bipendulum, the independent variable is t, the dependent variables are $x1$, $x2$, and u.

```
> ivar := [t]; dvar := [x1,x2,u];
```

$$ivar := [t]$$

$$dvar := [x1,\, x2,\, u]$$

We enter the left hand sides of the equations. The variables g, $l1$, and $l2$ are parameters for the system.

[1] The deduction of the given linear ordinary differential equations which describe the bipendulum relies on the approximation $\sin\theta \approx \theta$ for small angles θ of the pendula to the vertical.

```
> L1 := [g*x1(t)+l1*diff(x1(t),t,t)+g*u(t),
> g*x2(t)+l2*diff(x2(t),t,t)+g*u(t)];
```

$$L1 := [g\,\mathrm{x1}(t) + l1\,(\tfrac{d^2}{dt^2}\,\mathrm{x1}(t)) + g\,\mathrm{u}(t),\ g\,\mathrm{x2}(t) + l2\,(\tfrac{d^2}{dt^2}\,\mathrm{x2}(t)) + g\,\mathrm{u}(t)]$$

Since we deal with ordinary differential equations with constant coefficients, it is appropriate to consider the system over the commutative polynomial algebra

$$D = \mathbb{Q}(g, l1, l2)[\partial],$$

where ∂ represents the differential operator $\frac{d}{dt}$. Then (5.2) can be written as $R\,(x1 \quad x2 \quad u)^T = 0$ with

$$R := \begin{pmatrix} l1\,\partial^2 + g & 0 & g \\ 0 & l2\,\partial^2 + g & g \end{pmatrix} \in D^{2\times 3}.$$

The D-module which is associated with the system is

$$M := D^{1\times 3}/D^{1\times 2}\,R.$$

In order to check controllability of the bipendulum, we compute the extension group $\mathrm{ext}^1_D(M^\top, D)$ of the transposed module $M^\top = D^{2\times 1}/R\,D^{3\times 1}$ of M. Equivalently (see Theorem 5.1.5), we compute a presentation for the torsion submodule $t(M)$ of M. This can be achieved by using the command `Torsion` of the package `Janet` which implements Algorithm 5.2.8 using Janet bases:

```
> Torsion(L1, ivar, dvar);
```

$$[[\_\mathrm{T1}(t) = 0],\ [\_\mathrm{T1}(t)],\ 0]$$

The result means that the torsion submodule $t(M)$ of M is trivial. In general, the first list of the result consists of equations whose right hand sides are generators for the presentation of the torsion submodule and whose left hand sides define names for these generators. The second entry of the result is the list of relations satisfied by the generators for the presentation of $t(M)$ (more details are given below). The result includes the Hilbert series of $t(M)$ as last entry (see Section 2.3). In the present case, we conclude $t(M) = \{0\}$. However, this consequence only holds for the generic configurations of the parameters for the bipendulum, i.e. for a generic choice of g, $l1$, and $l2$. We check by which expressions the computation of $t(M)$ has divided:

```
> ZeroSets(ivar);
```

$$[g,\ l1,\ l2,\ l1 - l2]$$

Of course, g, $l1$, and $l2$ are assumed to be strictly positive. But the algorithm divided by the difference $l1 - l2$ which is zero if and only if the lengths of the two pendula are equal. Therefore, we study this particular case separately:

```
> L2 := subs(l2=l1, L1);
```

$$L2 := [g\,\mathrm{x1}(t) + l1\,(\tfrac{d^2}{dt^2}\,\mathrm{x1}(t)) + g\,\mathrm{u}(t),\ g\,\mathrm{x2}(t) + l1\,(\tfrac{d^2}{dt^2}\,\mathrm{x2}(t)) + g\,\mathrm{u}(t)]$$

By M' we denote the D-module which is associated with this particular system ($l1 = l2$). We compute a presentation of the torsion submodule $t(M')$ of M':

```
> T := Torsion(L2, ivar, dvar);
```

$$T := [[\_\mathrm{T1}(t) = \mathrm{x1}(t) - \mathrm{x2}(t)],\ [\_\mathrm{T1}(t)\,g + (\tfrac{d^2}{dt^2}\,\_\mathrm{T1}(t))\,l1],\ 1 + s]$$

The list of expressions by which the algorithm divided assures that this result is valid in any case:

```
> ZeroSets(ivar);
```

$$[g,\ l1]$$

The first list in T gives one generator $x1 - x2$ for $t(M')$. For the presentation of $t(M')$ this generator is called _T1. The second list in T is a generating set of the differential relations satisfied by _T1. It serves as the list of relations for the presentation of $t(M')$. Finally, the Hilbert series of $t(M')$ given in the last entry of T states that $t(M')$ is two-dimensional as a $\mathbb{Q}(g, l1)$-vector space. More precisely, $t(M')$ has the $\mathbb{Q}(g, l1)$-basis $(\_\mathrm{T1}, \frac{d\,\_\mathrm{T1}}{dt})$.

We conclude that the bipendulum is controllable if and only if the lengths of the two pendula are different. This coincides with the intuition that for the case $l1 = l2$ we consider two copies of a system consisting of one pendulum alone which are joined by the bar. If the initial configurations of the pendula of equal lengths are the same (positions and velocities), then the link of these two systems enforces the motions of the two pendula to be the same. Hence, in case $l1 = l2$ the behavior of the bipendulum is partitioned into two subsets: the set of trajectories with the same motions for the two pendula and the set of trajectories with distinct motions of the two pendula. The membership of an element of the behavior to either of these subsets depends only on the initial conditions and not on any control.

The second example of this section deals with a biological reactor. The dynamics of the underlying system model is investigated in [ALLR82]. We thank the process systems engineering group of Prof. W. Marquardt (RWTH Aachen) for providing us with this model.

Example 5.3.3. A simple model of an isothermal continuous stirred tank biological reactor is described by the following nonlinear system (cf. [ALLR82])

$$\left\{\begin{array}{rcl} \dfrac{dX}{dt} & = & -\dfrac{F}{V}\,X + \mu(S)\,X, \\[2ex] \dfrac{dS}{dt} & = & \dfrac{F\,(SF - S)}{V} - \sigma(S)\,X, \end{array}\right.$$

where X is the concentration of cells in the reactor, S the substrate concentration in the reactor, $\mu(S)$ the specific growth rate, $\sigma(S)$ the specific substrate consumption rate, SF the feed substrate concentration, F the volumetric feed flow rate and V the reactor volume. We consider here the following dimensionless form given in [ALLR82]

$$\begin{cases} \dfrac{d\,x1}{dt} &= -x1 + Da\, M(x2)\, x1\,, \\ \dfrac{d\,x2}{dt} &= -x2 + Da\, \Sigma(x2)\, x1\,, \end{cases}$$

where

$$M(x2) := (1 - x2)\, e^{SF\, x2/K}, \qquad \Sigma(x2) := \frac{M(x2)\,(a + b\, SF)}{a - b\,(SF - 1)\, x2}.$$

According to the given reactor model we set $a = \frac{27}{5}$, $b = 180$. It is convenient to consider $(x1 \quad x2)^T$ as the state of the system and the Damköhler number Da as an input variable.

```
> with(Janet):
```

The independent variable is t, the dependent variables are $x1$, $x2$, Da.

```
> ivar := [t]; dvar := [x1,x2,Da];
```

$$ivar := [t]$$

$$dvar := [x1,\ x2,\ Da]$$

We enter the system equations as a list of left hand sides:

```
> L := [x1[t]+x1+(x2-1)*exp(SF*x2/K)*Da*x1,
> (27/5-180*SF*x2+180*SF)*(x2[t]+x2)+(x2-1)*
> exp(SF*x2/K)*(27/5+180*SF)*Da*x1];
```

$$L := [x1_t + x1 + (x2 - 1)\, e^{(\frac{SF\, x2}{K})}\, Da\, x1,$$
$$(\frac{27}{5} - 180\, SF\, x2 + 180\, SF)\,(x2_t + x2) + (x2 - 1)\, e^{(\frac{SF\, x2}{K})}\,(\frac{27}{5} + 180\, SF)\, Da\, x1]$$

Next we are going to compute the general linearization (see Remark 3.2.10, p. 53) of the system of equations defined by L. The new indeterminates for the general linearization are chosen to be $dx1$, $dx2$, dDa.

```
> Dvar := [dx1,dx2,dDa];
```

$$Dvar := [dx1,\ dx2,\ dDa]$$

The general linearization of L (which is first translated from jet notation into differential expressions) is:

```
> GL := Linearize(Ind2Diff(L, ivar, dvar), ivar, dvar, Dvar);
```

$$
\begin{aligned}
GL := \Bigg[& \left(\tfrac{d}{dt}\,\mathrm{dx1}(t)\right) + \mathrm{dx1}(t) + \mathrm{dx1}(t)\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x2}(t) - \mathrm{dx1}(t)\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t) \\
& + \frac{e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x1}(t)\,(K + SF\,\mathrm{x2}(t) - SF)\,\mathrm{dx2}(t)}{K} \\
& + (\mathrm{x2}(t) - 1)\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{x1}(t)\,\mathrm{dDa}(t), \\
& \frac{9}{5}\,(\mathrm{x2}(t) - 1)\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,(3 + 100\,SF)\,\mathrm{Da}(t)\,\mathrm{dx1}(t) + \frac{9}{5}(3\,(\tfrac{d}{dt}\,\mathrm{dx2}(t))\,K \\
& - 100\,(\tfrac{d}{dt}\,\mathrm{dx2}(t))\,K\,SF\,\mathrm{x2}(t) + 100\,(\tfrac{d}{dt}\,\mathrm{dx2}(t))\,K\,SF - 100\,\mathrm{dx2}(t)\,SF\,K\,(\tfrac{d}{dt}\,\mathrm{x2}(t)) \\
& - 200\,\mathrm{dx2}(t)\,SF\,\mathrm{x2}(t)\,K + 3\,\mathrm{dx2}(t)\,K + 100\,\mathrm{dx2}(t)\,SF\,K \\
& + 3\,\mathrm{dx2}(t)\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x1}(t)\,K + 100\,\mathrm{dx2}(t)\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x1}(t)\,K\,SF \\
& + 3\,\mathrm{dx2}(t)\,SF\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x1}(t)\,\mathrm{x2}(t) \\
& + 100\,\mathrm{dx2}(t)\,SF^2\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x1}(t)\,\mathrm{x2}(t) - 3\,\mathrm{dx2}(t)\,SF\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x1}(t) \\
& - 100\,\mathrm{dx2}(t)\,SF^2\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,\mathrm{Da}(t)\,\mathrm{x1}(t))/K \\
& + \frac{9}{5}\,(\mathrm{x2}(t) - 1)\, e^{\left(\frac{SF\,\mathrm{x2}(t)}{K}\right)}\,(3 + 100\,SF)\,\mathrm{x1}(t)\,\mathrm{dDa}(t) \Bigg]
\end{aligned}
$$

Since the equations involve the exponential function $\exp(SF\ x2/K)$, we deal with the differential ring $\mathbb{Q}(K, SF)\{x1, x2, Da, \exp(SF\ x2/K)\}$. Let I be the differential ideal of this ring which is generated by the differential polynomials in L. Then we consider the general linearization over the Ore algebra

$$D := \mathrm{Quot}(\mathbb{Q}(K, SF)\{x1, x2, Da, \exp(SF\ x2/K)\})[\partial],$$

where ∂ acts by differentiation with respect to t. In order to compute in this differential ring we provide Janet's algorithm with the following rewriting rules for the coefficients of the general linearization, which are jet expressions in $x1$, $x2$, Da:

```
> N := [op(solve({op(L)}, {x1[t],x2[t]}))];
```

$$
\begin{aligned}
N := [& x1_t = -x1 - e^{\left(\frac{SF\,x2}{K}\right)}\,Da\,x1\,x2 + e^{\left(\frac{SF\,x2}{K}\right)}\,Da\,x1,\ x2_t = (-3\,e^{\left(\frac{SF\,x2}{K}\right)}\,Da\,x1 \\
& + 3\,x2 + 3\,e^{\left(\frac{SF\,x2}{K}\right)}\,Da\,x1\,x2 - 100\,SF\,x2^2 + 100\,SF\,x2 + 100\,SF\,e^{\left(\frac{SF\,x2}{K}\right)}\,Da\,x1\,x2 \\
& - 100\,SF\,e^{\left(\frac{SF\,x2}{K}\right)}\,Da\,x1)/(-3 + 100\,SF\,x2 - 100\,SF)]
\end{aligned}
$$

Let us denote by M the left D-module which is associated with the general linearization. In order to check controllability of the biological reactor under consideration we construct a presentation for the extension group $\mathrm{ext}^1_D(M^\top, D)$ or, equivalently, for the torsion submodule $t(M)$ of M (see Theorem 5.1.5). For the computation we consider both $M^\top$ and $\mathrm{ext}^1_D(M^\top, D)$ as left D-modules by means of the involution defined in Ex. 5.2.3 (a).

```
>  T := Torsion(GL, ivar, Dvar, "coeffeqs"=N, "coeffdvar"=dvar);
```

$$T := [[\_\mathrm{T1}(t) = 0], [\_\mathrm{T1}(t)], 0]$$

The first entry of the result T gives a generating set for $t(M)$ which consists of zero only. We conclude that $t(M) = 0$, i.e. the biological reactor is controllable for generic configurations of the parameters.

The command `Torsion` applies Janet's algorithm several times. Altogether the above runs of Janet's algorithm divided by the following expressions:

```
>  ZeroSets(ivar);
```

$$[K,\, (3 + 100\, SF)\, e^{(\frac{SF\, \mathrm{x2}(t)}{K})},\, e^{(-\frac{2\, SF\, \mathrm{x2}(t)}{K})},\, e^{(\frac{3\, SF\, \mathrm{x2}(t)}{K})},\, \mathrm{x2}(t)\, e^{(\frac{3\, SF\, \mathrm{x2}(t)}{K})},$$
$$(\mathrm{x2}(t) - 1)\, e^{(\frac{3\, SF\, \mathrm{x2}(t)}{K})},\, \mathrm{x1}(t)\, e^{(\frac{4\, SF\, \mathrm{x2}(t)}{K})},\, SF\, e^{(\frac{3\, SF\, \mathrm{x2}(t)}{K})},$$
$$(100\, SF\, \mathrm{x2}(t) - 100\, SF - 3)\, e^{(\frac{3\, SF\, \mathrm{x2}(t)}{K})},\, e^{(-\frac{SF\, \mathrm{x2}(t)}{K})}]$$

Controllability of the biological reactor holds if all these expressions are different from zero. Some of these expressions may describe inequalities for the parameters which need to be obeyed in order to avoid abnormal operation of the reactor. It is clear that numerical simulation is not sufficient to find such inequalities so that a preprocessing using symbolic computation could also be useful for the numerics.

The only interesting constant term above is $3 + 100\, SF$. However, the case $3 + 100\, SF = 0$ leads to a negative feed substrate concentration, which makes no sense.

We conclude that the linearization of the biological reactor is completely controllable (for the interesting configurations of the parameters). Since this linearized system has no autonomous observables, we expect that the given nonlinear system has no autonomous observables either. If there existed one, then an autonomous observable of the linearization could be obtained by linearizing the one of the nonlinear system. Hence, for the given nonlinear model of the biological reactor we do not expect any obstructions towards controllability.

In the next section we are going to study a nonlinear system which has an autonomous observable in detail.

5.4 Autonomous Observables

In this section an example for a system with autonomous observables is discussed in detail. Using the general linearization and the differential algebraic methods described in Chapter 3, we find a generating set for the torsion submodule of the module which is associated with the general linearization. Then we show how it is possible to integrate an autonomous observable which corresponds to an element of the torsion submodule in order to find an autonomous observable of the given nonlinear system. For more details about the correspondence between the autonomous observables of the linearized system and the autonomous observables of the nonlinear system we refer to [ABMP95].

Example 5.4.1. The system of ordinary differential equations which we investigate in this example is taken from [Pom94, Example VII.C.3]. It is a nonlinear system of ordinary differential equations for four unknown functions:

$$\left\{\begin{array}{rcl} \dfrac{d\,y1}{dt} - y2 - y3\,u & = & 0, \\ \dfrac{d\,y2}{dt} + y1 & = & 0, \\ \dfrac{d\,y3}{dt} + y1\,u & = & 0. \end{array}\right. \tag{5.3}$$

We are going to study the structural properties of the system in Maple using the packages `Janet`, `jets`, and `diffalg`.

```
>   with(Janet):
>   with(jets):
>   with(diffalg):
```

The independent variable is t, the dependent variables are $y1$, $y2$, $y3$, and u.

```
>   ivar := [t]; dvar := [y1,y2,y3,u];
```

$$ivar := [t]$$

$$dvar := [y1,\, y2,\, y3,\, u]$$

The left hand sides of the homogeneous equations are collected in the list L:

```
>   L := [diff(y1(t),t)-y2(t)-y3(t)*u(t), diff(y2(t),t)+y1(t),
>   diff(y3(t),t)+y1(t)*u(t)];
```

$$L := [(\tfrac{d}{dt}\,\mathrm{y1}(t)) - \mathrm{y2}(t) - \mathrm{y3}(t)\,\mathrm{u}(t),\ (\tfrac{d}{dt}\,\mathrm{y2}(t)) + \mathrm{y1}(t),\ (\tfrac{d}{dt}\,\mathrm{y3}(t)) + \mathrm{y1}(t)\,\mathrm{u}(t)]$$

The equations can be thought of as describing the state $(y1, y2, y3)$ which is possibly influenced by the input u.

We are going to compute the general linearization of this system of ordinary differential equations and use $Y1$, $Y2$, $Y3$, U as new indeterminates for the linearization:

```
> Dvar := [Y1,Y2,Y3,U];
```

$$Dvar := [Y1,\ Y2,\ Y3,\ U]$$

```
> GL := Linearize(L, ivar, dvar, Dvar);
```

$$GL := [(\tfrac{d}{dt}\,\mathrm{Y1}(t)) - \mathrm{Y2}(t) - \mathrm{u}(t)\,\mathrm{Y3}(t) - \mathrm{y3}(t)\,\mathrm{U}(t),\ \mathrm{Y1}(t) + (\tfrac{d}{dt}\,\mathrm{Y2}(t)),\ \mathrm{u}(t)\,\mathrm{Y1}(t) + (\tfrac{d}{dt}\,\mathrm{Y3}(t)) + \mathrm{y1}(t)\,\mathrm{U}(t)]$$

Let I be the differential ideal generated in $\mathbb{Q}(t)\{y1, y2, y3, u\}$ by the differential polynomials in the list L. In order to compute with coefficients in the differential ring $\mathbb{Q}(t)\{y1, y2, y3, u\}/I$, we extract rewriting rules for the jet variables $y1^k_J$, $y2^k_J$, $y3^k_J$, u^k_J from (5.3). First of all we translate L into jet notation:

```
> Diff2Ind(L, ivar, dvar);
```

$$[y1_t - y2 - y3\,u,\ y2_t + y1,\ y3_t + y1\,u]$$

Next we compute a prime decomposition of I (see Thm. 3.3.6 and Rem. 3.3.8). To this end we apply the command `Rosenfeld_Groebner` of the package `diffalg`.

```
> differential_ring(derivations=[t], ranking=[[y1,y2,y3,u]]);
```

$$ODE\_ring$$

```
> P := Rosenfeld_Groebner([y1[t]-y2[]-y3[]*u[], y2[t]+y1[],
> y3[t]+y1[]*u[]], %);
```

$$P := [characterizable]$$

```
> equations(P[1]), inequations(P[1]);
```

$$[y1_t - y2 - y3\,u,\ y2_t + y1,\ y3_t + y1\,u],\ []$$

We conclude that I is a prime differential ideal. Therefore we continue just by compiling L to a list of rewriting rules which do not lead to infinite loops.

```
> N := [y1[t] = y2+y3*u, y2[t] = -y1, y3[t] = -y1*u];
```

$$N := [y1_t = y2 + y3\,u,\ y2_t = -y1,\ y3_t = -y1\,u]$$

The linear system given by the general linearization GL is represented by the matrix

$$R := \begin{pmatrix} \partial & -1 & -u & -y3 \\ 1 & \partial & 0 & 0 \\ u & 0 & \partial & y1 \end{pmatrix} \in D^{3\times 4}$$

over the Ore algebra

$$D = \mathrm{Quot}(\mathbb{Q}(t)\{y1, y2, y3, u\}/I)[\partial],$$

where ∂ represents differentiation with respect to t. The left D-module associated with this system is

$$M = D^{1\times 4}/D^{1\times 3}\, R.$$

We are going to compute the first extension group $\mathrm{ext}^1_D(M^\top, D)$ of the transposed module $M^\top = D^{3\times 1}/R\, D^{4\times 1}$ of M.

A presentation of $t(M) \cong \mathrm{ext}^1_D(M^\top, D)$ (see Theorem 5.1.5) is returned by the command `Torsion`. The following computation takes the rewriting rules for the coefficients in $\mathrm{Quot}(\mathbb{Q}(t)\{y1, y2, y3, u\}/I)$ into account. The involution defined in Ex. 5.2.3 (a) is used to consider $M^\top$ and $\mathrm{ext}^1_D(M^\top, D)$ as left D-modules.

```
> T := Torsion(GL, ivar, Dvar, "coeffeqs"=N, "coeffdvar"=dvar);
```

$$T := [[\_T1(t) = \mathrm{y3}(t)\,\mathrm{y1}(t)\,\mathrm{u}(t)\,\mathrm{Y1}(t) + \mathrm{Y2}(t)\,\mathrm{u}(t)\,\mathrm{y2}(t)\,\mathrm{y3}(t) + \mathrm{u}(t)\,\mathrm{y3}(t)^2\,\mathrm{Y3}(t),$$
$$\_T2(t) = -\mathrm{y1}(t)\,\mathrm{y3}(t)\,\mathrm{Y1}(t) - \mathrm{Y2}(t)\,\mathrm{y2}(t)\,\mathrm{y3}(t) - \mathrm{y3}(t)^2\,\mathrm{Y3}(t)],$$
$$[-\_T1(t) - \_T2(t)\,\mathrm{u}(t),$$
$$(\tfrac{d}{dt}\,\_T1(t))\,\mathrm{y3}(t)\,\mathrm{u}(t) - \_T1(t)\,(\tfrac{d}{dt}\,\mathrm{u}(t))\,\mathrm{y3}(t) - \_T1(t)\,(\tfrac{d}{dt}\,\mathrm{y3}(t))\,\mathrm{u}(t)],\ 1]$$

During the whole computation the procedures divided by the following expressions only:

```
> ZeroSets(ivar);
```

$$[\mathrm{u}(t),\ \mathrm{u}(t),\ \mathrm{y3}(t),\ \mathrm{y1}(t)]$$

Hence, along all solutions $(y1, y2, y3, u)$ of the nonlinear system (5.3) there are autonomous observables of the linearized system which are given by _T1 and _T2 as defined in the first entry of T and all their left D-linear combinations. The second entry of T is a generating set for the relations satisfied by _T1 and _T2. In particular, _T1 satisfies the equation

$$y3\, u\, \frac{d}{dt}\_T1 - \frac{du}{dt}\, y3\, \_T1 - u\, \frac{d\, y3}{dt}\, \_T1 = 0.$$

Moreover, _T1 equals _T2 multiplied by $-u$. The last entry of T is the Hilbert series of $\mathrm{ext}^1_D(M^\top, D) \cong t(M)$. We realize that $t(M)$ is one-dimensional as a $\mathrm{Quot}(\mathbb{Q}(t)\{y1, y2, y3, u\}/I)$-vector space.

Now we want to find autonomous observables of the given nonlinear system which correspond to the autonomous observables that we found for the linearization.

It is possible to decide whether a jet expression is a formal Fréchet derivative (see Def. 3.2.6, p. 52) of some jet expression by applying a certain operator of the variational bicomplex [And, Bar01a] to it. A given jet expression is a formal Fréchet derivative of some jet expression if and only if the procedure `frechetv` of the package `jets` applied to the differential operator in matrix form which represents the jet expression returns the zero matrix.

In order to apply the command `frechetv` we first translate the differential expression defining _T1 into a differential operator in matrix form, i.e. we construct D1 satisfying D1 $(Y1, Y2, Y3, U)^T =$ _T1.

```
> D1 := Diff2Op(Diff2Ind(rhs(T[1][1]), ivar, dvar), ivar, Dvar);
```

$$\mathrm{D1} := \begin{bmatrix} [[y3\ y1\ u,\ []]] & [[u\ y2\ y3,\ []]] & [[u\ y3^2,\ []]] & 0 \end{bmatrix}$$

Each entry of a matrix representing a differential operator is a list of lists. The outer list represents a sum, and each inner list consists of a jet expression and a list of independent variables. The jet expression is the coefficient for the differential operator encoded by the list of independent variables, e.g. $[u, [t, t]]$ is a notation for $u\,\frac{d^2}{dt^2}$. Now it is clear that the first entries of the inner lists in D1 are exactly the coefficients of $Y1$, $Y2$ resp. $Y3$ in _T1.

```
> frechetv(D1, ivar, dvar);
```

$$\begin{bmatrix} 0 & 0 & [[y1\ u\,S,\ []]] & [[y1\ y3\ S,\ []]] \\ 0 & 0 & [[u\ y2\ S,\ []]] & [[y2\ y3\ S,\ []]] \\ [[-y1\ u\,S,\ []]] & [[-u\ y2\ S,\ []]] & 0 & [[y3^2\ S,\ []]] \\ [[-y1\ y3\ S,\ []]] & [[-y2\ y3\ S,\ []]] & [[-y3^2\ S,\ []]] & 0 \end{bmatrix}$$

Since the resulting matrix of operators is not zero, _T1 is not a formal Fréchet derivative of any jet expression.

We try the same for the autonomous observable _T2:

```
> D2 := Diff2Op(Diff2Ind(rhs(T[1][2]), ivar, dvar), ivar, Dvar);
```

$$\mathrm{D2} := \begin{bmatrix} [[-y1\ y3,\ []]] & [[-y2\ y3,\ []]] & [[-y3^2,\ []]] & 0 \end{bmatrix}$$

```
> frechetv(D2, ivar, dvar);
```

$$\begin{bmatrix} 0 & 0 & [[-y1\ S,\ []]] & 0 \\ 0 & 0 & [[-y2\ S,\ []]] & 0 \\ [[y1\ S,\ []]] & [[y2\ S,\ []]] & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Again we conclude that _T2 is not a formal Fréchet derivative.

By inspection we are led to define the following autonomous observable $T3$:

```
> T3 := simplify(-rhs(T[1][2])/y3(t));
```

$$T3 := \mathrm{y1}(t)\,\mathrm{Y1}(t) + \mathrm{y2}(t)\,\mathrm{Y2}(t) + \mathrm{y3}(t)\,\mathrm{Y3}(t)$$

The command `AutonomEq` of the package `Janet` computes a generating set of differential equations which are satisfied by $T3$ modulo the equations given by the general linearization and all its differential consequences. By adding the rewriting rules N for the jet variables $y1^k_J$, $y2^k_J$, $y3^k_J$ we compute with coefficients in the differential field $\mathrm{Quot}(\mathbb{Q}(t)\{y1, y2, y3, u\}/I)$.

```
> AutonomEq(T3, GL, ivar, Dvar, "coeffeqs"=N, "coeffdvar"=dvar);
```

$$[\tfrac{d}{dt}\,\_\mathrm{A}(t)]$$

The equations computed by `AutonomEq` include a new indeterminate $\_\mathrm{A}$ which represents the differential expression given as first argument. Hence, $\_\mathrm{A}$ is to be replaced by $T3$. We conclude that $T3$ satisfies the equation $\frac{d\,T3}{dt} = 0$.

In order to check whether $T3$ is a formal Fréchet derivative of some jet expression, we first translate $T3$ into a differential operator in matrix form:

```
> D3 := Diff2Op(Diff2Ind(T3, ivar, dvar), ivar, Dvar);
```

$$\mathrm{D3} := \begin{bmatrix} [[y1, []]] & [[y2, []]] & [[y3, []]] & 0 \end{bmatrix}$$

```
> frechetv(D3, ivar, dvar);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The resulting differential operator is zero. Hence, there are no formal obstructions towards the existence of a jet expression whose formal Fréchet derivative is $T3$. In fact, such a jet expression can be found using the homotopy formula [Olv93]

$$r(x^i, u^k_J) \mapsto \int_0^1 r(\lambda\, x^i, \lambda\, u^k_J)\, \frac{d\lambda}{\lambda}.$$

We define this homotopy formula as an operator in Maple by using some functions of the package `jets`. (Here it is sufficient to perform the substitution of the jet coordinates by their λ-multiples up to order two only.)

```
> homotopy := r -> int(subs(map(a->a=lambda*a,
> jetcoor([$0..2], ivar, dvar)),
> appmt(r, dvar, ivar, dvar)[1])/lambda, lambda=0..1):
```

When we apply this homotopy operator to D3, we obtain the following jet expression:

```
>  a := homotopy(D3);
```

$$a := \frac{y1^2}{2} + \frac{y2^2}{2} + \frac{y3^2}{2}$$

In fact, the formal Fréchet derivative of the previous jet expression equals $T3$:

```
>  Linearize(Ind2Diff(a, ivar, dvar), ivar, dvar, Dvar);
```

$$\mathrm{y1}(t)\,\mathrm{Y1}(t) + \mathrm{y2}(t)\,\mathrm{Y2}(t) + \mathrm{y3}(t)\,\mathrm{Y3}(t)$$

Moreover, a satisfies a differential equation. We have

$$\frac{da}{dt} = y1\,\frac{d\,y1}{dt} + y2\,\frac{d\,y2}{dt} + y3\,\frac{d\,y3}{dt},$$

and by taking the nonlinear system (5.3) into account, we obtain

$$y1\,\frac{d\,y1}{dt} + y2\,\frac{d\,y2}{dt} + y3\,\frac{d\,y3}{dt} = y1\,(y2 + y3\,u) - y1\,y2 - y1\,y3\,u = 0.$$

Hence, we have $\frac{da}{dt} = 0$, and a is an autonomous observable which corresponds to the autonomous observable $T3$ of the linearized system.

5.5 The Maple package OreModules

In this section we give a short survey of the Maple package OreModules which provides routines for the study of linear systems which are defined over Ore algebras. The OreModules project was started by F. Chyzak and A. Quadrat in 2002 and has been extended continuously since 2003 by A. Quadrat and the author of this thesis.

Several methods for dealing with (left) modules over Ore algebras are implemented in OreModules. Of course, these methods are always restricted to Ore algebras for which Janet or Gröbner bases can be computed in Maple. These include commutative polynomial algebras, the Weyl algebras, algebras of shift operators and combinations of them. By default, the Maple packages Ore_algebra and Groebner written by F. Chyzak are used to perform the Gröbner basis computations which are necessary for the module-theoretic constructions. However, an interface to JanetOre (see Section 2.6) is also provided which allows to switch to Janet bases. In Chapter 7 we apply OreModules using this interface to a linear system which describes a stirred tank model and demonstrate many features of this package.

Among the implemented purely module-theoretic methods are procedures which compute syzygies of the rows of a given matrix, which construct free resolutions of finitely presented (left) modules, which compute extension groups etc.

Several more tools for matrices with entries in Ore algebras are available like the computation of left and right inverses, elimination of variables etc.

For linear systems represented by matrices with entries in one of the above mentioned Ore algebras the following system theoretic problems can be addressed using `OreModules`:

- decide controllability, parametrizability, flatness, and π-freeness [CQR05], [FM98], [Mou95];
- compute parametrizations (see also Chapter 6), autonomous observables, flat outputs, π-polynomials;
- tools for linear quadratic optimal control problems (see Section 7.6);
- some methods from classical control theory: controllability matrices, Brunovský canonical forms.

A more recent extension of the `OreModules` project is the possibility to compute bases of free left modules of rank at least 2 over the Weyl algebras $A_n(k)$, where k has characteristic zero, using a well-known result of J. T. Stafford [Sta78] which states that every left ideal of $A_n(k)$ can be generated by two elements. More precisely, given a left ideal by means of three generators a, b, c, such two generators can be found as $a + \lambda\, c$, $b + \mu\, c$ with appropriate $\lambda, \mu \in A_n(k)$. Exploiting this particular representation of these two generators and using recent algorithmic versions [HS01, Ley04] of Stafford's result, bases of free left $A_n(k)$-modules of rank at least 2 can be computed in `OreModules` for rather small examples. This method is explained in Section 6.6.

For more detailed descriptions of `OreModules` see [CQR06a], [CQR06b]. The `OreModules` web pages [CQR06a] also contain a "Library of Examples" which demonstrate the procedures of `OreModules` on (small) application problems.

Chapter 6

Parametrizing Linear Systems

A system of linear equations with coefficients in a field can be transformed by using Gaussian elimination into a form which allows to determine the solution space of the system very easily. If the system is underdetermined, then the Gauß-reduced matrix associated with it singles out some variables of the system as parameters and specifies how all other variables are expressed (linearly) in terms of the parameters. This procedure can be viewed as identifying the kernel of the linear map induced by the system matrix as the image of another linear map. In particular, every tuple of values assigned to the tuple of parameters yields a solution, i.e. the parameters are not subject to any relation.

In this chapter the problem of parametrizing the solution space of a linear system is described for linear systems with coefficients in an Ore algebra. The most common application is to linear systems of differential equations (Monge's problem [Zer32], [Jan71]). In the behavioral approach to linear systems [PW98], [Zer00], parametrizing the solution set is referred to as the problem of finding an image representation of the behavior. The way of presenting the parametrizability problem in this chapter traces back to J.-F. Pommaret and A. Quadrat [Pom95, Pom01, PQ99a, PQ99b]. A detailed account on parametrizations in the framework of Ore algebras was given in [CQR05]. In [QR05b] a method for parametrizing uncontrollable linear systems was described which is explained in more detail below.

Whereas in the case of linear equations with coefficients in a field it is always achieved that the solution set is parametrized by an injective linear map, the possibility to find an injective parametrization is not given in general. In system theoretic applications, linear and also nonlinear systems of differential equations whose solution sets have injective parametrizations in terms of arbitrary functions are nowadays said to be flat [FLMR95].

We start in the first section by illustrating the notion of parametrizability of linear systems on two examples. In the following section we formally define parametrizations of the solutions sets of linear systems using the module-theoretic approach described in Chapter 5. A well-known characterization of parametrizability expressed in terms of the module which is associated with the linear system is recalled. It signifies that a linear system is parametrizable if and only if it has no autonomous observables. Section 6.3 shows how the notion of parametrization can be extended in such a way that the solutions of certain linear systems having autonomous observables can also be parametrized. For linear systems of ordinary differential equations whose coefficients are constant or polynomials or rational functions this extension is always possible. Some examples of this extension are given in Section 6.4. The notion of flatness of linear systems is discussed in Section 6.5. It it equivalent to the freeness of the module which is associated with the linear system. For certain Ore algebras it is possible to compute bases of free modules effectively. The case of free left modules of rank at least 2 over the Weyl algebras defined over a field of characteristic zero is treated in Section 6.6.

The methods used in this chapter are mainly module-theoretic. The connection to the function spaces containing the solutions of the linear system under consideration is provided by duality methods established by B. Malgrange, L. Ehrenpreis, V. P. Palamodov, and U. Oberst (see Section 4.4).

6.1 Introduction

In this section we first illustrate the notion of parametrization on the well-known de Rham complex. A counter-example to parametrizability is given thereafter.

Throughout this chapter, D will denote an Ore algebra and $\mathcal{F}$ a left D-module which serves as a signal space, i.e. one could think of $\mathcal{F}$ as a function space in which solutions of a given linear system are searched for (see Section 4.1).

We recall (see Convention 4.1.2) that with every matrix $R \in D^{q \times p}$ we associate two homomorphisms:

$$(.R) : D^{1 \times q} \to D^{1 \times p} : m \mapsto m\,R \qquad \text{and} \qquad (R.) : \mathcal{F}^{p \times 1} \to \mathcal{F}^{q \times 1} : \eta \mapsto R\,\eta.$$

Example 6.1.1. Let us consider the linear system

$$\operatorname{div} y = \begin{pmatrix} \partial_{x_1} & \partial_{x_2} & \partial_{x_3} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = 0 \qquad \text{with } y \in \mathcal{F}^{3 \times 1}, \tag{6.1}$$

where $\mathcal{F} := C^\infty(\mathbb{R}^3)$. Since it is a linear system of partial differential equations with constant coefficients, it is appropriate to consider $\mathcal{F}$ as module over the commutative polynomial algebra $D := \mathbb{R}[\partial_{x_1}, \partial_{x_2}, \partial_{x_3}]$. It is well-known that *every solution y of (6.1) can be written as*

$$y = \operatorname{rot} w = \begin{pmatrix} 0 & -\partial_{x_3} & \partial_{x_2} \\ \partial_{x_3} & 0 & -\partial_{x_1} \\ -\partial_{x_2} & \partial_{x_1} & 0 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \qquad \text{for some } w \in \mathcal{F}^{3 \times 1},$$

and *any* $w \in \mathcal{F}^{3\times 1}$ *defines a solution* $y = \operatorname{rot} w$ *of* (6.1), i.e. the solutions $\mathcal{B}_{\mathcal{F}}(R_1)$ of (6.1) are parametrized by the operator rot:

$$\mathcal{B}_{\mathcal{F}}(R_1) = \operatorname{im}((R_2).), \quad R_2 := \operatorname{rot} \in D^{3\times 3}.$$

Of course, this parametrization is not injective: many different $w \in \mathcal{F}^{3\times 1}$ result in the same $y = \operatorname{rot} w$.

One may ask whether $R_2\, w = 0$ can be parametrized in the same way. In fact, *every solution* $w \in \mathcal{F}^{3\times 1}$ *of* $R_2\, w = 0$ *can be written as*

$$w = \operatorname{grad} v = \begin{pmatrix} \partial_{x_1} \\ \partial_{x_2} \\ \partial_{x_3} \end{pmatrix} v \qquad \text{for some } v \in \mathcal{F},$$

and *any* $v \in \mathcal{F}$ *defines a solution* $w = \operatorname{grad} v$ *of* $R_2\, w = 0$, so that we have a parametrization of $\mathcal{B}_{\mathcal{F}}(R_2)$:

$$\mathcal{B}_{\mathcal{F}}(R_2) = \operatorname{im}((R_3).), \quad R_3 := \operatorname{grad} \in D^{3\times 1}.$$

Since all solutions of $R_3\, u = 0$ in $\mathcal{F} = C^\infty(\mathbb{R}^3)$ are constant, altogether we obtain the *de Rham complex*:

$$0 \longrightarrow \mathbb{R} \longrightarrow \mathcal{F} \xrightarrow{\operatorname{grad}} \mathcal{F}^{3\times 1} \xrightarrow{\operatorname{rot}} \mathcal{F}^{3\times 1} \xrightarrow{\operatorname{div}} \mathcal{F} \longrightarrow 0.$$

In this complex grad, rot, div are understood as operators acting on (vectors of) functions.

Example 6.1.2. We consider a system of linear differential time-delay equations which arises in the context of a linearized model of Saint-Venant's equations [DPR99].

$$\text{(6.2)} \qquad \left\{ \begin{array}{rcl} \phi_1(t-2) + \phi_2(t) - 2\,\dot{\phi}_3(t-1) & = & 0, \\[2ex] \phi_1(t) + \phi_2(t-2) - 2\,\dot{\phi}_3(t-1) & = & 0. \end{array} \right.$$

The operators involved in these equations can be taken from a commutative polynomial algebra $D := \mathbb{R}[\partial, \delta]$ because (6.2) is a time-invariant system, i.e. the equations have constant coefficients. We consider $\mathcal{F} := C^\infty(\mathbb{R})$ as a D-module, where ∂ and δ act as follows:

$$\partial : y(t) \mapsto \dot{y}(t), \qquad \delta : y(t) \mapsto y(t-1), \qquad y = y(t) \in \mathcal{F}.$$

We define

$$\text{(6.3)} \qquad R = \begin{pmatrix} \delta^2 & 1 & -2\,\delta\,\partial \\ 1 & \delta^2 & -2\,\delta\,\partial \end{pmatrix} \in D^{2\times 3}.$$

Then the linear system (6.2) can be written as $R\,\phi = 0$.

This system has some solutions. For instance, each 2-periodic function $\theta \in \mathcal{F}$ defines a solution

$$\begin{pmatrix} \phi_1(t) \\ \phi_2(t) \\ \phi_3(t) \end{pmatrix} = \begin{pmatrix} \theta(t) \\ -\theta(t) \\ 0 \end{pmatrix}$$

of the system. We want to analyze now, whether we can find a parametrization of the solutions of (6.2) in the sense of the preceding example. Let $(\phi_1, \phi_2, \phi_3)^T \in \mathcal{F}^{3\times 1}$ be a solution of $R\,\phi = 0$ and let $\tau := \phi_1 - \phi_2$. Taking the equations (6.2) into account we have:

$$\begin{aligned} (\delta^2\,\tau)(t) &= \phi_1(t-2) - \phi_2(t-2) \\ &= -\phi_2(t) + \phi_1(t) + 2\,\dot{\phi}_3(t-1) - 2\,\dot{\phi}_3(t-1) \\ &= -\phi_2(t) + \phi_1(t) \\ &= \tau(t). \end{aligned}$$

Hence, τ is an *autonomous observable* of the system. It satisfies the equation

$$(\delta^2 - 1)\,\tau = 0, \tag{6.4}$$

i.e., τ is a 2-periodic function. Suppose that there exists a parametrization of the behavior of $R\,\phi = 0$, i.e., there exists $P \in D^{3\times m}$ for some $m \in \mathbb{N}$ such that

$$R \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} = 0 \quad \Longleftrightarrow \quad \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} = P\,\xi, \quad \xi \in \mathcal{F}^{m\times 1}.$$

Relation (6.4) implies a relation between the components of ξ: from

$$\phi_j = \sum_{i=1}^{m} P_{ji}\,\xi_i, \qquad j = 1, 2, 3$$

and

$$\tau = \phi_1 - \phi_2 = \sum_{i=1}^{m} (P_{1i} - P_{2i})\,\xi_i$$

we find

$$(\delta^2 - 1)\,\tau = \sum_{i=1}^{m} (P_{1i} - P_{2i})(\delta^2 - 1)\,\xi_i = 0. \tag{6.5}$$

This contradicts the properties of a parametrization in the sense of the previous example: not all $\xi \in \mathcal{F}^{m\times 1}$ define a solution of (6.2), but only those $\xi \in \mathcal{F}^{m\times 1}$ which satisfy (6.5). Hence there exists no parametrization of the solution set of (6.2) in the above sense. This argument holds for every signal space $\mathcal{F}$.

6.2 Parametrizing Linear Systems over Ore Algebras

In this section the notion of parametrization of a behavior as motivated in the previous section is defined formally using the module-theoretic approach of Chapter 5. In particular, both the behavior and the module which is associated with the linear system are considered. The significance of injective cogenerators $\mathcal{F}$ for the parametrizability of the solution space of a linear system as a subset of $\mathcal{F}^{p\times 1}$ is explained. Finally, the well-known necessary and sufficient condition for parametrizability on the module-theoretic side is recalled in Theorem 6.2.8.

Throughout this section let D be a left Noetherian Ore algebra and $\mathcal{F}$ a left D-module. We consider $R \in D^{q\times p}$ defining the linear system $R\,\eta = 0$.

First we recall Definition 4.1.3:

Definition 6.2.1. The *solution set* of $R\,\eta = 0$ in $\mathcal{F}$, where $R \in D^{q\times p}$, is defined as

$$\mathrm{Sol}_{\mathcal{F}}(R) := \ker(R.) = \{\eta \in \mathcal{F}^{p\times 1} \mid R\,\eta = 0\}.$$

A set $\mathcal{B} \subseteq \mathcal{F}^{p\times 1}$ which is a solution set $\mathrm{Sol}_{\mathcal{F}}(R)$ for some $R \in D^{q\times p}$, is called a *behavior*. We also write

$$\mathcal{B}_{\mathcal{F}}(R) := \mathrm{Sol}_{\mathcal{F}}(R).$$

Definition 6.2.2. A behavior $\mathcal{B}$ is *parametrizable*, if there exists a matrix $P \in D^{p\times m}$ such that $\mathcal{B} = P\,\mathcal{F}^{m\times 1}$, i.e., for every $\eta \in \mathcal{B}$, there exists $\xi \in \mathcal{F}^{m\times 1}$ such that $\eta = P\,\xi$, and for all $\xi \in \mathcal{F}^{m\times 1}$ we have $R\,P\,\xi = 0$. In this case, P (and also the induced map $(P.)$) is called a *parametrization of the behavior* $\mathcal{B}$.

Remark 6.2.3. By definition, a behavior $\mathcal{B}$ is represented as the kernel of a homomorphism $(R.) : \mathcal{F}^{p\times 1} \longrightarrow \mathcal{F}^{q\times 1}$ (kernel representation of a behavior). The behavior $\mathcal{B}$ is parametrizable if and only if there exists a matrix $P \in D^{p\times m}$ such that

$$\mathcal{F}^{q\times 1} \xleftarrow{R.} \mathcal{F}^{p\times 1} \xleftarrow{P.} \mathcal{F}^{m\times 1}$$

is an exact sequence of left D-modules (see Def. 4.2.1). A parametrization P of $\mathcal{B}$ constitutes a very convenient representation of $\mathcal{B}$ as the image of the homomorphism $(P.) : \mathcal{F}^{m\times 1} \longrightarrow \mathcal{F}^{p\times 1}$ (image representation of a behavior). Among the applications of image representations of behaviors is e.g. a method to solve linear quadratic optimal control problems (see Section 7.6, [PQ04] and [QR06b]).

Definition 6.2.4. Let $R \in D^{q\times p}$ and $M := D^{1\times p}/D^{1\times q}\,R$. The left D-module M is called *parametrizable*, if there exists a matrix $P \in D^{p\times m}$ such that

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{.P} D^{1\times m}$$

is an exact sequence of left D-modules. In this case, P (and also the induced map $(.P)$) is called a *parametrization of* M.

Remark 6.2.5. In contrast to a parametrization of a behavior, where the image of the map $(P.)$ is considered, for a parametrization of a module the kernel of the map $(.P)$ is relevant.

Remark 6.2.6. Let the left D-module $M = D^{1\times p}/D^{1\times q}R$ be parametrizable, and let $P \in D^{p\times m}$ be a parametrization of M. Then

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{\rho} M \longrightarrow 0$$

is the beginning of a free resolution of M. Moreover, by definition of a parametrization of M, the complex

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{.P} D^{1\times m}$$

is exact. We conclude that M can be considered as a submodule of the free module $D^{1\times m}$:

$$\begin{tikzcd} D^{1\times q} \arrow[r, ".R"] & D^{1\times p} \arrow[r, ".P"] \arrow[dr, "\rho"'] & D^{1\times m} & \\ & & M \arrow[u, "\iota"'] \arrow[dr] & \\ & & 0 \arrow[u] & 0 \end{tikzcd} \tag{6.6}$$

The homomorphism $\iota : M \to D^{1\times m}$ is constructed as follows: For any $m \in M$ there is a preimage $r \in D^{1\times p}$ of m under the canonical projection ρ, namely any representative of the residue class m. Then we define $\iota(m) := r\,P$. This definition is independent of the choice of r because two representatives of m differ by an element $s\,R$, where $s \in D^{1\times q}$. But $s\,R$ is mapped to zero under $(.P)$. Furthermore, ι is injective because, if $m \in M$ satisfies $\iota(m) = 0$, then the residue class m has a representative $s\,R$ for some $s \in D^{1\times q}$ due to the exactness of the upper row in (6.6). But this means, by definition of M, that m is the zero residue class.

The next proposition shows how the notions of parametrization of a behavior resp. a module are related depending on the properties of the signal space $\mathcal{F}$.

Proposition 6.2.7. *Let $R \in D^{q\times p}$ and $M := D^{1\times p}/D^{1\times q}\,R$ be the left D-module associated with the linear system $R\,\eta = 0$. Moreover, let $P \in D^{p\times m}$.*

(a) If $\mathcal{F}$ is a cogenerator for the category of left D-modules (see Def. 4.4.1) and P is a parametrization of the behavior $\mathcal{B} = \mathrm{Sol}_{\mathcal{F}}(R)$, then P is a parametrization of M.

(b) If $\mathcal{F}$ is an injective left D-module (see Def. 4.3.1) and P is a parametrization of M, then P is a parametrization of the behavior $\mathcal{B} = \mathrm{Sol}_{\mathcal{F}}(R)$.

(c) *If $\mathcal{F}$ is an injective cogenerator for the category of left D-modules, then P is a parametrization of the behavior $\mathcal{B} = \mathrm{Sol}_{\mathcal{F}}(R)$ if and only if P is a parametrization of M.*

Proof. (a) We assume that P is a parametrization of $\mathcal{B}$, i.e.

$$\mathcal{F}^{q\times 1} \xleftarrow{R.} \mathcal{F}^{p\times 1} \xleftarrow{P.} \mathcal{F}^{m\times 1} \tag{6.7}$$

is an exact sequence of left D-modules. Since $\mathcal{F}$ is a cogenerator for the category of left D-modules, the complex

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{.P} D^{1\times m}, \tag{6.8}$$

from which (6.7) is obtained by applying $\hom_D(\,\cdot\,, \mathcal{F})$, is exact due to Prop. 4.4.3. Hence, P is a parametrization of M.

(b) By assumption, (6.8) is an exact sequence. Since $\mathcal{F}$ is injective, the complex (6.7) is exact. Then we have:

$$\mathrm{Sol}_{\mathcal{F}}(R) = \ker(R.) = \mathrm{im}(P.) = P\,\mathcal{F}^{m\times 1}.$$

(c) is clear from (a) and (b). □

Theorem 6.2.8. [Pom01] *The left D-module $M = D^{1\times p}/D^{1\times q}\,R$ is parametrizable if and only if $t(M) = 0$.*

Proof. "⇒": By Remark 6.2.6, M can be considered as a submodule of a free module $D^{1\times m}$. Since $D^{1\times m}$ is torsion-free (see Prop. 4.2.8), M does not contain any non-zero torsion element either.

"⇐": Let us assume that $t(M) = 0$. We consider the beginning of a free resolution of the right D-module $M^\top := D^{q\times 1}/(R\,D^{p\times 1})$:

$$0 \longleftarrow M^\top \longleftarrow D^{q\times 1} \xleftarrow{R.} D^{p\times 1} \xleftarrow{P.} D^{m\times 1}.$$

Applying $\hom_D(\,\cdot\,, D)$ to this exact sequence yields the complex

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{.P} D^{1\times m}. \tag{6.9}$$

By definition of $\mathrm{ext}^1_D(M^\top, D)$ (see Def. 4.2.12), the complex (6.9) is exact if and only if

$$\mathrm{ext}^1_D(M^\top, D) = \ker(.P)/\,\mathrm{im}(.R) = 0.$$

Now Theorem 5.1.5 states that we have

$$t(M) \cong \mathrm{ext}^1_D(M^\top, D).$$

Therefore, the assumption implies that $\mathrm{ext}^1_D(M^\top, D) = 0$. Hence, P is a parametrization of M. □

Remark 6.2.9. Let $\mathcal{F}$ be an injective left D-module. By carrying out part "$\Leftarrow$" of the previous proof constructively, we either obtain a parametrization P of $R\,\eta = 0$ or a generating set of $t(M)$. Let us consider the complex (6.9). In order to check whether (6.9) is exact, we construct the syzygy module of the rows of P, i.e. we compute a matrix $R' \in D^{1\times q'}$ such that the lower complex in the diagram

$$\begin{tikzcd} D^{1\times q} \arrow[r, ".R"] & D^{1\times p} \arrow[r, ".P"] & D^{1\times m} \arrow[r] & 0 \\ D^{1\times q'} \arrow[ur, ".R'"'] & & & \end{tikzcd}$$

is exact. Therefore we have $D^{1\times q}\,R = \operatorname{im}(.R) \subseteq \operatorname{im}(.R') = D^{1\times q'}\,R'$ and

$$\operatorname{ext}^1_D(M^\top, D) = \ker(.P)/\operatorname{im}(.R) = \operatorname{im}(.R')/\operatorname{im}(.R).$$

If $\operatorname{im}(.R') = \operatorname{im}(.R)$, then the sequence (6.9) is exact and P is a parametrization of the behavior of $R\,\eta = 0$. Otherwise, the residue classes in $M = D^{1\times p}/D^{1\times q}\,R$ represented by the rows of R' form a generating set for $t(M)$.

Corollary 6.2.10. *Let $R \in D^{q\times p}$ and $M = D^{1\times p}/D^{1\times q}\,R$ the left D-module associated with the linear system $R\,y = 0$. Moreover, let $\mathcal{F}$ be an injective cogenerator for the category of left D-modules. The behavior $\mathcal{B} = \operatorname{Sol}_{\mathcal{F}}(R)$ is parametrizable if and only if $t(M) \neq 0$.*

The *proof* is obtained by combining Proposition 6.2.7 and Theorem 6.2.8.

Example 6.2.11. Returning to Ex. 6.1.2, we have

$$R = \begin{pmatrix} \delta^2 & 1 & -2\,\delta\,\partial \\ 1 & \delta^2 & -2\,\delta\,\partial \end{pmatrix} \in D^{2\times 3},$$

where $D = \mathbb{R}[\partial, \delta]$ is the commutative polynomial algebra. Then the D-module $M = D^{1\times 2}/D^{1\times 3}\,R$ is associated with this linear system. The transposed module $M^\top$ of M is defined by $M^\top := D^{2\times 1}/R\,D^{3\times 1}$. We compute a generating set of the syzygy module of the columns of R and arrive at the following exact sequence:

$$0 \longleftarrow M^\top \longleftarrow D^{2\times 1} \xleftarrow{\begin{pmatrix} \delta^2 & 1 & -2\,\delta\,\partial \\ 1 & \delta^2 & -2\,\delta\,\partial \end{pmatrix}\cdot} D^{3\times 1} \xleftarrow{\begin{pmatrix} 2\,\delta\,\partial \\ 2\,\delta\,\partial \\ 1+\delta^2 \end{pmatrix}\cdot} D^{1\times 1}.$$

By applying $\hom_D(\,\cdot\,, D)$ to this exact sequence, we obtain the complex

$$D^{1\times 2} \xrightarrow{\cdot\begin{pmatrix} \delta^2 & 1 & -2\,\delta\,\partial \\ 1 & \delta^2 & -2\,\delta\,\partial \end{pmatrix}} D^{1\times 3} \xrightarrow{\cdot\begin{pmatrix} 2\,\delta\,\partial \\ 2\,\delta\,\partial \\ 1+\delta^2 \end{pmatrix}} D^{1\times 1}.$$

Now we compute the syzygies of the rows of $P := (2\,\delta\,\partial \quad 2\,\delta\,\partial \quad 1+\delta^2)^T \in D^{3\times 1}$.

$$\begin{tikzcd}
D^{1\times 2} \arrow[r, ".\begin{pmatrix} \delta^2 & 1 & -2\,\delta\,\partial \\ 1 & \delta^2 & -2\,\delta\,\partial \end{pmatrix}"] & D^{1\times 3} \arrow[r, ".\begin{pmatrix} 2\,\delta\,\partial \\ 2\,\delta\,\partial \\ 1+\delta^2 \end{pmatrix}"] & D^{1\times 1} \\
D^{1\times 2} \arrow[ur, ".\begin{pmatrix} 1 & -1 & 0 \\ 0 & -1-\delta^2 & 2\,\delta\,\partial \end{pmatrix} =: R'"'] & &
\end{tikzcd}$$

We are now in position to check the exactness of the upper complex in the previous diagram. Let us denote the first row of R' by $m := (1 \quad -1 \quad 0) \in D^{1\times 3}$. Then we check that $m \notin \operatorname{im}(.R)$, i.e. the residue class $m + \operatorname{im}(.R)$ is not zero in M. Moreover, we verify that

$$(\delta^2 - 1)\,((1 \quad -1 \quad 0) + \operatorname{im}(.R)) = 0 \qquad \text{in } M,$$

and both rows of R' represent the same residue class in M. We conclude

$$t(M) \cong \operatorname{ext}^1_D(M^\top, D) = \operatorname{im}(.R')/\operatorname{im}(.R) \neq 0,$$

and the torsion element $m + \operatorname{im}(.R)$ generates $t(M)$. In particular, the behavior of $R\,\eta = 0$ is not parametrizable, irrespective of which signal space $\mathcal{F}$ is chosen. However, P is a parametrization of $M/t(M)$.

6.3 Parametrizing Linear Systems with Autonomous Observables

As exposed in the previous section, the behavior of a linear system is parametrizable if and only if the linear system is controllable, i.e. if and only if it has no autonomous observables. However, in certain situations the solutions of uncontrollable linear systems can be expressed as a sum of a parametrization of the controllable part of the system and a vector of solutions of the equations fulfilled by autonomous observables of the system. Such a situation is on hand when the torsion submodule $t(M)$ of the module M associated with the linear system has a complement in M. For linear systems of ordinary differential equations whose coefficients are constant, polynomials, or rational functions, this is always the case. In the behavioral setting for the study of linear systems with constant coefficients, constructive solutions to interconnection and decomposition problems were given in [ZL01]. In this section we give several details about the approach presented in [QR05b] for the according problems in the framework of Ore algebras.

We continue to consider a left Noetherian Ore algebra D. Let M be a left D-module. We recall that

$$0 \longrightarrow t(M) \xrightarrow{\iota} M \xrightarrow{\rho} M/t(M) \longrightarrow 0 \tag{6.10}$$

is a short exact sequence of left D-modules, where $t(M)$ is the torsion submodule of M (see Def. 4.2.4). In this complex, ι and ρ denote the canonical injection resp. the canonical projection. As explained in Ex. 4.2.5, ι and ρ are not always written down.

Definition 6.3.1. [Rot79] A short exact sequence of left D-modules

$$0 \longrightarrow M' \xrightarrow{f} M \xrightarrow{g} M'' \longrightarrow 0 \tag{6.11}$$

is *split*, if there exists a homomorphism $h : M'' \to M$ of left D-modules satisfying $g \circ h = \mathrm{id}_{M''}$.

Remarks 6.3.2. (a) The short exact sequence (6.11) is split if and only if there exists a homomorphism $k : M \to M'$ of left D-modules satisfying $k \circ f = \mathrm{id}_{M'}$. In this case, we have $M \cong M' \oplus M''$, and M'' is called a *complement* of M' in M.

(b) The homomorphism k can be constructed from $h : M'' \to M$ in such a way that $\ker(k) = \mathrm{im}(h)$, i.e.

$$0 \longleftarrow M' \xleftarrow{k} M \xleftarrow{h} M'' \longleftarrow 0 \tag{6.12}$$

is a short exact sequence.

Remark 6.3.3. Let (6.11) be a split short exact sequence of left D-modules and let $\mathcal{F}$ be an injective left D-module. Then

$$0 \longleftarrow \hom_D(M', \mathcal{F}) \xleftarrow{f^*} \hom_D(M, \mathcal{F}) \xleftarrow{g^*} \hom_D(M'', \mathcal{F}) \longleftarrow 0 \tag{6.13}$$

is an exact sequence of k-vector spaces, where f^*, g^* are defined as in Prop. 4.3.4. Moreover, since (6.11) is split, we also have the short exact sequence (6.12), which, by the injectivity of $\mathcal{F}$, gives rise to the short exact sequence

$$0 \longrightarrow \hom_D(M', \mathcal{F}) \xrightarrow{k^*} \hom_D(M, \mathcal{F}) \xrightarrow{h^*} \hom_D(M'', \mathcal{F}) \longrightarrow 0, \tag{6.14}$$

where h^*, k^* are defined as in Prop. 4.3.4. As a short exact sequence of k-vector spaces, (6.14) is always split because every k-vector space is free and hence projective. Here we want to stress that h^* in (6.14) actually is a homomorphism satisfying $h^* \circ g^* = \mathrm{id}_{\hom_D(M'', \mathcal{F})}$: for every $\varphi \in \hom_D(M'', \mathcal{F})$ we have

$$(h^* \circ g^*)(\varphi) = (g \circ h)^*(\varphi) = \varphi \circ g \circ h = \varphi$$

because $g \circ h = \mathrm{id}_{M''}$ holds by assumption.

We apply the notion of a split short exact sequence to the case $M' = t(M)$ and $M'' = M/t(M)$, i.e. to the short exact sequence (6.10), where M is the left D-module associated with a linear system.

Remark 6.3.4. Let $\mathcal{F}$ be an injective left D-module. For a given matrix $R \in D^{q\times p}$ we consider the left D-module $M = D^{1\times p}/D^{1\times q} R$ which is associated with $R\eta = 0$. Moreover, let $R' \in D^{q'\times p}$ be a matrix such that $M/t(M) = D^{1\times p}/D^{1\times q'} R'$. We consider the short exact sequence

$$0 \longrightarrow t(M) \xrightarrow{\iota} M \xrightarrow{\rho} M/t(M) \longrightarrow 0. \tag{6.15}$$

Since $\mathcal{F}$ is injective,

$$0 \longleftarrow \hom_D(t(M), \mathcal{F}) \longleftarrow \hom_D(M, \mathcal{F}) \longleftarrow \hom_D(M/t(M), \mathcal{F}) \longleftarrow 0$$

is an exact sequence of k-vector spaces. Using Malgrange's isomorphism (see Prop. 4.1.8), we have

$$\hom_D(M, \mathcal{F}) \cong \mathrm{Sol}_{\mathcal{F}}(R), \qquad \hom_D(M/t(M), \mathcal{F}) \cong \mathrm{Sol}_{\mathcal{F}}(R') \tag{6.16}$$

as k-vector spaces, so that the previous short exact sequence can also be written (up to isomorphism) as

$$0 \longleftarrow \hom_D(t(M), \mathcal{F}) \xleftarrow{\lambda} \mathrm{Sol}_{\mathcal{F}}(R) \xleftarrow{\kappa} \mathrm{Sol}_{\mathcal{F}}(R') \longleftarrow 0.$$

It is clear that we have the following isomorphism of k-vector spaces:

$$\mathrm{Sol}_{\mathcal{F}}(R) \cong \mathrm{Sol}_{\mathcal{F}}(R') \oplus \hom_D(t(M), \mathcal{F}).$$

Let us assume that the short exact sequence (6.15) is split. Then, according to Remark 6.3.3, a projection π from $\mathrm{Sol}_{\mathcal{F}}(R)$ onto $\mathrm{Sol}_{\mathcal{F}}(R')$ can be obtained as ρ^* (up to the isomorphisms (6.16)). In order to turn this remark into a useful tool for solving certain linear systems, we will translate the present situation into the language of matrices below.

Remark 6.3.5. By the definition of a projective module (Def. 4.2.6), if M'' is projective, then (6.11) is split. For the Ore algebras

$$D \in \{\, k[\partial],\ A_1(k) = k[t][\partial],\ B_1(k) = k(t)[\partial] \,\},$$

every finitely generated torsion-free left D-module is projective (see Theorem 4.2.9 (a) and (b), p. 74). If the linear system is given by ordinary differential equations with coefficients which are constant, polynomials, or rational functions in t, then the associated module M can be defined over one of these Ore algebras. In this case the torsion-free module $M'' = M/t(M)$ is therefore projective. Hence, every such linear system of ordinary differential equations gives rise to a split short exact sequence (6.15), so that we can always draw the conclusion from Remark 6.3.4 about the set of solutions for these linear systems.

We consider the following commutative diagram of left D-modules:

$$
\begin{tikzcd}
 & & 0 \arrow[d] & \\
 & 0 \arrow[d] & t(M) \arrow[d] & \\
D^{1\times q} \arrow[r, ".R"] \arrow[d] & D^{1\times p} \arrow[r] \arrow[d, "\mathrm{id}"] & M \arrow[r] \arrow[d, "\rho"] & 0 \\
D^{1\times q'} \arrow[r, ".R'"] & D^{1\times p} \arrow[r] \arrow[d] & M/t(M) \arrow[r] \arrow[d] & 0 \\
 & 0 & 0 &
\end{tikzcd}
\tag{6.17}
$$

By applying $\hom_D(\,\cdot\,,\mathcal{F})$ to the complexes of left D-modules which are contained in this diagram we obtain:

$$
\begin{tikzcd}
 & & 0 & \\
 & 0 & \hom_D(t(M),\mathcal{F}) \arrow[u] & \\
\mathcal{F}^{q\times 1} & \mathcal{F}^{p\times 1} \arrow[l, "R."'] \arrow[u] & \mathrm{Sol}_{\mathcal{F}}(R) \arrow[l] \arrow[u] & 0 \arrow[l] \\
\mathcal{F}^{q'\times 1} \arrow[u] & \mathcal{F}^{p\times 1} \arrow[l, "R'."'] \arrow[u, "\mathrm{id}"'] & \mathrm{Sol}_{\mathcal{F}}(R') \arrow[l] \arrow[u] & 0 \arrow[l] \\
 & 0 \arrow[u] & 0 \arrow[u] &
\end{tikzcd}
\tag{6.18}
$$

For a visualization of the maps used in the next lemma, we refer to the diagram (6.17).

Lemma 6.3.6. *Let $R \in D^{q\times p}$, $M = D^{1\times p}/D^{1\times q}\,R$ and $R' \in D^{q'\times p}$ such that $M/t(M) = D^{1\times p}/D^{1\times q'}\,R'$. The short exact sequence*

$$0 \longrightarrow t(M) \longrightarrow M \longrightarrow M/t(M) \longrightarrow 0 \tag{6.19}$$

is split if and only if there exist matrices $S \in D^{p\times q'}$ and $V \in D^{q'\times q}$ such that

$$R' - R'\,S\,R' = V\,R. \tag{6.20}$$

Proof. Let $\rho : M \to M/t(M)$ be the canonical projection. By Def. 6.3.1, (6.19) is split if and only if there exists a homomorphism $\sigma : M/t(M) \to M$ such that $\rho\circ\sigma = \mathrm{id}_{M/t(M)}$.

Every homomorphism $M \to M/t(M)$ can be defined by assigning its images on the elements of a generating set of M in such a way that all left D-linear relations satisfied by these generators are fulfilled for the corresponding images. More precisely, in the present context we have $M = D^{1\times p}/D^{1\times q}\, R$ so that the residue classes in M of the standard basis vectors of $D^{1\times p}$ form a generating set of M. All left D-linear relations for this generating set are generated by the rows of R. Similarly, the target $M/t(M) = D^{1\times p}/D^{1\times q'}\, R'$ is generated by the residue classes in $M/t(M)$ of the standard basis vectors of $D^{1\times p}$ whose left D-linear relations are generated by the rows of R'. Hence, every homomorphism $M \to M/t(M)$ can be defined on representatives of residue classes by an endomorphism $\Delta : D^{1\times p} \to D^{1\times p}$ which satisfies

$$\Delta(D^{1\times q}\, R) \subseteq D^{1\times q'}\, R'.$$

For $\rho : M \to M/t(M)$ the endomorphism Δ can be chosen to be the identity. Analogously, $\sigma : M/t(M) \to M$ is induced by an endomorphism $\Sigma : D^{1\times p} \to D^{1\times p}$ which satisfies

$$\Sigma(D^{1\times q'}\, R') \subseteq D^{1\times q}\, R. \tag{6.21}$$

We represent Σ with respect to the standard basis of $D^{1\times p}$ by a matrix $U \in D^{p\times p}$. Then (6.21) is equivalent to the existence of a matrix $V \in D^{q'\times q}$ such that

$$R'\, U = V\, R. \tag{6.22}$$

Let us now assume that (6.19) is split. Then, by the preceding remarks, there exist $U \in D^{p\times p}$ and $V \in D^{q'\times q}$ such that (6.22) holds. The property $\rho \circ \sigma = \mathrm{id}_{M/t(M)}$ translates into the existence of a matrix $S \in D^{p\times q'}$ such that

$$U\, I_p + S\, R' = I_p, \tag{6.23}$$

where I_p is the $(p \times p)$-identity matrix. By multiplying by R' from the left and using (6.22) we find (6.20).

Conversely, if (6.20) holds, then we define $U := I_p - S\, R'$. Then we have

$$R'\, U = R' - R'\, S\, R' = V\, R,$$

which means that U represents a well-defined homomorphism $\sigma : M/t(M) \to M$ which satisfies $\rho \circ \sigma = \mathrm{id}_{M/t(M)}$ because (6.23) holds. □

Lemma 6.3.7. *Let $R \in D^{q\times p}$, $M = D^{1\times p}/D^{1\times q}\, R$ and $R' \in D^{q'\times p}$, $M/t(M) = D^{1\times p}/D^{1\times q'}\, R'$. Then there exists a matrix $R'' \in D^{q\times q'}$ such that $R = R''\, R'$.*

Proof. In the terminology of the proof of the previous lemma, the canonical projection $\rho : M \to M/t(M)$ is represented by $\Delta = \mathrm{id}_{D^{1\times p}} : D^{1\times p} \to D^{1\times p}$ which satisfies $\Delta(D^{1\times q}\, R) \subseteq D^{1\times q'}\, R'$. Hence there exists a matrix $R'' \in D^{q\times q'}$ satisfying $R\, I_p = R''\, R'$. □

Theorem 6.3.8. *Let $\mathcal{F}$ be an injective left D-module, $R \in D^{q \times p}$ and $M = D^{1 \times p} / D^{1 \times q} R$ the left D-module associated with $R\,\eta = 0$. Moreover, let $R' \in D^{q' \times p}$ such that $M/t(M) = D^{1 \times p} / D^{1 \times q'} R'$ and*

$$(\underbrace{L'}_{\in D^{r \times q'}} \quad \underbrace{L}_{\in D^{r \times q}}) \in D^{r \times (q'+q)} \quad \text{such that} \quad \ker\left(. \begin{pmatrix} R' \\ R \end{pmatrix} \right) = D^{1 \times r} \, (L' \quad L).$$

If there exist matrices $S \in D^{p \times q'}$ and $V \in D^{q' \times q}$ satisfying

$$R' - R'\,S\,R' = V\,R, \tag{6.24}$$

then we have

$$R\,\eta = 0 \qquad \Longleftrightarrow \qquad \eta = P\,\xi + S\,\tau, \quad \xi \in \mathcal{F}^{m \times 1},$$

where $P \in D^{p \times m}$ is a parametrization of $M/t(M)$ and τ runs through all solutions of

$$L'\,\tau = 0.$$

Remarks 6.3.9. (a) The matrix composed of L and L' in the assumption of Thm. 6.3.8 can be obtained from a computation of syzygies of the rows of the matrix $(R'^T \quad R^T)^T \in D^{(q'+q) \times p}$ (see Remark 4.2.11).

(b) By Lemma 6.3.6, the existence of matrices $S \in D^{p \times q'}$ and $V \in D^{q' \times q}$ satisfying (6.24) is equivalent to the fact that (6.19) is split.

(c) Since $P \in D^{p \times m}$ in the assertion of the theorem is a parametrization of $M/t(M)$, we have $P\,\xi \in \mathrm{Sol}_{\mathcal{F}}(R')$ for all $\xi \in \mathcal{F}^{m \times 1}$.

Proof of Thm. 6.3.8. "$\Rightarrow$": We show that every solution η of $R\,\eta = 0$ is of the form $\eta = P\,\xi + S\,\tau$ for some $\xi \in \mathcal{F}^{m \times 1}$ and some solution τ of $L'\,\tau = 0$.

Let $\eta \in \mathrm{Sol}_{\mathcal{F}}(R)$ be arbitrary. Set $U := I_p - S\,R' \in D^{p \times p}$. By applying U to η we obtain

$$\eta = U\,\eta + S\,R'\,\eta.$$

We have $U\,\eta \in \mathrm{Sol}_{\mathcal{F}}(R')$ because

$$R'\,U\,\eta = (R' - R'\,S\,R')\,\eta = V\,R\,\eta = 0.$$

Since $\mathcal{F}$ is injective, the parametrization P of $M/t(M)$ is also a parametrization of $\mathrm{Sol}_{\mathcal{F}}(R')$ (see Prop. 6.2.7 (b)). Hence, there exists some $\xi \in \mathcal{F}^{m \times 1}$ such that $U\,\eta = P\,\xi$. Moreover, $\tau := R'\,\eta$ is a solution of $L'\,\tau = 0$ because

$$L'\,\tau = L'\,R'\,\eta = -L\,R\,\eta = 0$$

holds by assumption. Altogether we have shown that η can be written as $\eta = P\,\xi + S\,\tau$ as claimed.

"$\Leftarrow$": Let τ be a solution of $L'\tau = 0$ and let $\xi \in \mathcal{F}^{m\times 1}$ be arbitrary. Since $\mathcal{F}$ is an injective left D-module, the inhomogeneous linear system

$$\begin{pmatrix} R' \\ R \end{pmatrix} \eta = \begin{pmatrix} \tau \\ 0 \end{pmatrix}$$

is solvable if and only if the compatibility conditions

$$(L' \quad L) \begin{pmatrix} \tau \\ 0 \end{pmatrix} = L'\tau = 0$$

are fulfilled (compare to Ex. 4.3.5, p. 80). Since this is the case by assumption, there exists $\eta \in \mathrm{Sol}_{\mathcal{F}}(R)$ such that $R'\eta = \tau$.

By Lemma 6.3.7 there exists a matrix $R'' \in D^{q\times q'}$ such that $R = R''R'$. Using such a matrix R'' and (6.24) we are able to show that $\eta := P\xi + S\tau$ is a solution of $R\eta = 0$:

$$\begin{aligned} R\eta &= R\,P\,\xi + R\,S\,\tau \\ &= R''\,(\underbrace{R'\,P}_{0})\,\xi + R''\,(R'\,S\,R')\eta \\ &= R''\,(R' - V\,R)\,\eta = 0. \end{aligned}$$

□

This theorem holds, in particular, in the case of linear systems of partial differential equations with constant coefficients for the $\mathbb{R}[\partial_1, \ldots, \partial_n]$-modules $\mathcal{F} = C^\infty(\Omega)$ und $\mathcal{D}'(\Omega)$ for open convex sets $\Omega \subseteq \mathbb{R}^n$, which are injective cogenerators for the category of $\mathbb{R}[\partial_1, \ldots, \partial_n]$-modules (see Ex. 4.4.5 (a) (3), (4)). Examples are given in the next section.

An important application of the possibility to express all solutions of a linear system as in Theorem 6.3.8 is to linear quadratic optimal control problems, where a quadratic cost functional is to be minimized subject to a linear system. If Theorem 6.3.8 applies in this situation, then substituting $P\xi + S\tau$ for η in the cost functional turns the given problem into a variational problem without constraints. For more details, we refer to Section 7.6, [PQ04] and [QR06b].

6.4 Applications

In this section we demonstrate Theorem 6.3.8 on two examples. The first one is the bipendulum which is described by a system of ordinary differential equations in Ex. 5.3.2. As a second example the linear system of differential time-delay equations which was introduced in Ex. 6.1.2 is investigated again. Both systems are not controllable in the sense of Section 6.2. The associated modules M are not torsion-free. However, in both examples the torsion submodules have a complement in M so that Theorem 6.3.8 can be applied.

Example 6.4.1. We reconsider the bipendulum [Pom01] of Ex. 5.3.2 for a configuration of the parameters for which the bipendulum is not controllable, i.e. the lengths of the two pendula are equal. In this example we apply the Maple package `OreModules` (see Section 5.5) in combination with `JanetOre` (see Section 2.6).

```
> with(OreModules):
> with(JanetOre):
```

First we set an option which makes `OreModules` use the package `JanetOre` for the necessary Janet basis computations.

```
> OreModulesOptions("GroebnerBasis", "JanetOre"):
```

Since we deal with a linear system of ordinary differential equations we define the Weyl algebra $A_1(\mathbb{Q}(g,l))$ in Maple. However, it is important to notice that, since the equations have constant coefficients, we actually deal with modules over the commutative ring

$$\mathcal{O} := \mathbb{Q}(g,l)[\mathrm{D}]$$

of polynomials in D with coefficients that are rational functions in the parameters g and l. Here D represents differentiation with respect to time t. We define the $\mathcal{O}$-module $\mathcal{F} = C^\infty(\mathbb{R})$ of smooth functions on $\mathbb{R}$ and consider solutions of the system in $\mathcal{F}$. Note that $\mathcal{F}$ is an injective cogenerator for ${}_{\mathcal{O}}M$ (see Ex. 4.4.5 (a) (3)).

```
> Alg := DefineOreAlgebra(diff=[D,t], polynom=[t], comm=[g,l]):
```

The system of equations (5.2), p. 104, is entered for the case $l1 = l2 = l$.

```
> R := matrix([[D^2+g/l, 0, -g/l], [0, D^2+g/l, -g/l]]);
```

$$R := \begin{bmatrix} \mathrm{D}^2 + \frac{g}{l} & 0 & -\frac{g}{l} \\ 0 & \mathrm{D}^2 + \frac{g}{l} & -\frac{g}{l} \end{bmatrix}$$

Let us denote the $\mathcal{O}$-module which is associated with the system by M. We compute the extension group $\mathrm{ext}^1_{\mathcal{O}}(M^\top, \mathcal{O})$.

```
> Ext1 := Exti(Involution(R, Alg), Alg, 1);
```

$$Ext1 := \left[\begin{bmatrix} -g - \mathrm{D}^2\, l & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & 0 \\ 0 & \mathrm{D}^2\, l + g & -g \end{bmatrix}, \begin{bmatrix} g \\ g \\ \mathrm{D}^2\, l + g \end{bmatrix} \right]$$

The result of `Exti` is a list with three entries. The second entry of *Ext1* is a matrix whose rows represent residue classes in M which form a generating set for $\mathrm{ext}^1_{\mathcal{O}}(M^\top, \mathcal{O})$. The entries of the i-th column of the first matrix in *Ext1* generate the annihilator in $\mathcal{O}$ of the residue class represented by the i-th row in *Ext1*[2]. In particular, we have $(l1\,\mathrm{D}^2 + g)\,m1 = 0$ in M, where $m1$ is the residue class represented by $(-1 \quad 1 \quad 0)$. This coincides with the corresponding result in Ex. 5.3.2. We conclude that the bipendulum is not parametrizable for this configuration of the parameters.

The third matrix in *Ext1* is a parametrization of $M/t(M)$. It was computed exactly as in the proof of Theorem 6.2.8. Let us denote this parametrization by P.

```
> P := Ext1[3];
```

$$P := \begin{bmatrix} g \\ g \\ \mathrm{D}^2\, l + g \end{bmatrix}$$

A generating set of torsion elements of $t(M) \cong \mathrm{ext}^1_{\mathcal{O}}(M^\top, \mathcal{O})$ in terms of the system variables and autonomous equations satisfied by the corresponding autonomous observables can also be computed as follows:

```
> TorsionElements(R, [x1(t),x2(t),u(t)], Alg);
```

$$[[\ -g\,\theta_1(t) - l\,(\tfrac{d^2}{dt^2}\,\theta_1(t)) = 0\ ],\ [\ \theta_1(t) = \mathrm{x1}(t) - \mathrm{x2}(t)\ ]]$$

As the bipendulum is described by ordinary differential equations, we know that the torsion submodule $t(M)$ has a complement in M (see Remark 6.3.5). Therefore, we can apply Theorem 6.3.8 to find a parametrization of the behavior of the bipendulum nevertheless.

Since the residue classes in M represented by the rows of *Ext1*[2] generate $t(M)$, we can also consider *Ext1*[2] as a presentation of $M/t(M)$, i.e. $M/t(M) \cong \mathcal{O}^{1\times 3}/\mathcal{O}^{1\times 2}\, \mathit{Ext1}[2]$. Hence, we have constructed a matrix R' in $\mathcal{O}^{2\times 3}$ which fulfills a part of the assumptions in Theorem 6.3.8. Let us denote this matrix by T in Maple.

```
> T := Ext1[2];
```

$$T := \begin{bmatrix} 1 & -1 & 0 \\ 0 & \mathrm{D}^2\, l + g & -g \end{bmatrix}$$

As was proved in Lemma 6.3.7 we can find a matrix $R'' \in \mathcal{O}^{2\times 2}$ which satisfies $R''\, R' = R$. Such a matrix can be found using the command `Factorize`.

```
> Factorize(R, T, Alg);
```

$$\begin{bmatrix} \dfrac{\mathrm{D}^2\, l + g}{l} & \dfrac{1}{l} \\ 0 & \dfrac{1}{l} \end{bmatrix}$$

In order to be able to apply Theorem 6.3.8 we still have to meet a few requirements. The matrix equation (6.24) is solved by a Janet basis computation (using the Kronecker product for matrices, this matrix equation is written as a system of affine equations for the unknown entries; for more details see [QR05b]). This is accomplished by the command `Complement` which returns, if possible, a list of matrices U, V, S satisfying the relations $T - T\,S\,T = V\,R$ and $U = I_3 - S\,T$.

```
> C := Complement(T, R, Alg);
```

$$C := \left[\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & l \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right]$$

In order to complete the requirements of Theorem 6.3.8 we compute the syzygy module of the rows of the matrix formed by stacking R' and R:

```
> L := SyzygyModule(linalg[stackmatrix](T, R), Alg);
```

$$L := \begin{bmatrix} \mathrm{D}^2\, l + g & 0 & -l & l \\ 0 & 1 & 0 & -l \end{bmatrix}$$

The submatrix L' of L obtained by selecting the first two columns defines a system of two linear ordinary differential equations for autonomous observables $\tau 1$, $\tau 2$ of the bipendulum. The second equation is $\tau 2 = 0$. Let us integrate the first equation:

```
> dsolve(l*diff(tau1(t), t, t) + g*tau1(t), tau1(t));
```

$$\tau 1(t) = \_C1 \sin(\frac{\sqrt{g}\, t}{\sqrt{l}}) + \_C2 \cos(\frac{\sqrt{g}\, t}{\sqrt{l}})$$

Hence, the general solution of the equation $L'\,\tau = 0$ in Theorem 6.3.8 is given by the following vector:

```
> tau := matrix(2, 1, [rhs(%), 0]);
```

$$\tau := \begin{bmatrix} \_C1 \sin(\frac{\sqrt{g}\, t}{\sqrt{l}}) + \_C2 \cos(\frac{\sqrt{g}\, t}{\sqrt{l}}) \\ 0 \end{bmatrix}$$

Now we can write down a vector $P\,\xi + S\,\tau$ depending on one function and two constants which can be chosen arbitrarily such that all solutions of R are given by this vector.

```
> eta := ApplyMatrix(P, [xi(t)], Alg) +
> ApplyMatrix(C[3], tau, Alg);
```

$$\eta := \begin{bmatrix} g\,\xi(t) \\ g\,\xi(t) \\ g\,\xi(t) + l\,(\frac{d^2}{dt^2}\,\xi(t)) \end{bmatrix} + \begin{bmatrix} \_C1\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l}}) + \_C2\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l}}) \\ 0 \\ 0 \end{bmatrix}$$

Hence, we have found a kind of parametrization of the behavior of the bipendulum in the uncontrollable case. The above vector η can be computed directly using the command `Parametrization` in `OreModules`, which uses `dsolve` to find the general solution of $L'\,\tau = 0$:

```
> Parametrization(R, Alg);
```

$$\begin{bmatrix} \_C1\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l}}) + \_C2\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l}}) + g\,\xi_1(t) \\ g\,\xi_1(t) \\ g\,\xi_1(t) + l\,(\frac{d^2}{dt^2}\,\xi_1(t)) \end{bmatrix}$$

Every solution $(x1, x2, u)^T \in C^\infty(\mathbb{R})^{3\times 1}$ of (5.2) is of the form

$$\begin{cases} x1(t) &= \_C1\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l}}) + \_C2\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l}}) + g\,\xi_1(t), \\ x2(t) &= g\,\xi_1(t), \\ u(t) &= g\,\xi_1(t) + l\,(\frac{d^2}{dt^2}\,\xi_1(t)) \end{cases}$$

for some constants $\_C1, \_C2 \in \mathbb{R}$ and some $\xi_1 \in C^\infty(\mathbb{R})$.

Example 6.4.2. We resume Ex. 6.1.2 in which a system of linear differential time-delay equations was studied.

```
> with(OreModules):
> with(JanetOre):
```

We select the interface of `OreModules` to `JanetOre`.

```
> OreModulesOptions("GroebnerBasis", "JanetOre"):
```

In order to handle the linear system (6.2) using `OreModules` we define the Ore algebra $Alg := \mathbb{Q}[t,s][D,\delta]$ which consists of polynomials in D and δ with coefficients that are polynomials in t and s.

```
> Alg := DefineOreAlgebra(diff=[D,t], dual_shift=[delta,s],
> polynom=[t,s], shift_action=[delta,t]):
```

Since the equations have constant coefficients, only the action of D and δ on functions is relevant in what follows. All modules will be considered over the commutative subring $\mathcal{O} := \mathbb{Q}[D,\delta]$ of Alg. According to the definition of Alg, D acts by differentiation with respect to t and δ acts as a shift of "length" 1 on t.

Next we define the matrix R given in (6.3).

```
> R := matrix([[delta^2, 1, -2*D*delta],
> [1, delta^2, -2*D*delta]]);
```

$$R := \begin{bmatrix} \delta^2 & 1 & -2\,\mathrm{D}\,\delta \\ 1 & \delta^2 & -2\,\mathrm{D}\,\delta \end{bmatrix}$$

We can verify the action of D and δ on functions by applying the matrix R to the vector $(\phi 1(t), \phi 2(t), \phi 3(t))^T$:

```
> ApplyMatrix(R, [phi1(t),phi2(t),phi3(t)], Alg);
```

$$\begin{bmatrix} \phi 1(t-2) + \phi 2(t) - 2\,\mathrm{D}(\phi 3)(t-1) \\ \phi 1(t) + \phi 2(t-2) - 2\,\mathrm{D}(\phi 3)(t-1) \end{bmatrix}$$

Let us denote the $\mathcal{O}$-module which is associated with the linear system by M. Then we compute the extension group $\mathrm{ext}_{\mathcal{O}}(M^{\top}, \mathcal{O})$.

```
> Ext1 := Exti(Involution(R, Alg), Alg, 1);
```

$$\mathit{Ext1} := \left[\begin{bmatrix} \delta^2 - 1 & 0 \\ 0 & \delta^2 - 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & 0 \\ 0 & -\delta^2 - 1 & 2\,\mathrm{D}\,\delta \end{bmatrix}, \begin{bmatrix} 2\,\mathrm{D}\,\delta \\ 2\,\mathrm{D}\,\delta \\ 1 + \delta^2 \end{bmatrix} \right]$$

The rows of $\mathit{Ext1}[2]$ represent residue classes which form a generating set for the torsion submodule $t(M)$ of M. The entries of the i-th column of $\mathit{Ext1}[1]$ annihilate the generator m_i of $t(M)$ represented by the i-th row in $\mathit{Ext1}[2]$ so that we have $(\delta^2 - 1)\, m_i = 0$ in M. Therefore, we have found non-trivial torsion elements of M. Moreover $P := \mathit{Ext1}[3]$ is parametrization of $M/t(M)$.

```
> P := Ext1[3];
```

$$P := \begin{bmatrix} 2\,\mathrm{D}\,\delta \\ 2\,\mathrm{D}\,\delta \\ 1 + \delta^2 \end{bmatrix}$$

The generators for $t(M)$ can also be obtained in terms of the system variables together with the equations represented by *Ext1*[1] as follows:

```
> TorsionElements(R, [phi1(t),phi2(t),phi3(t)], Alg);
```

$$\left[\begin{bmatrix} \theta_1(t-2)-\theta_1(t)=0 \\ \theta_2(t-2)-\theta_2(t)=0 \end{bmatrix}, \begin{bmatrix} \theta_1(t)=\phi 1(t)-\phi 2(t) \\ \theta_2(t)=-\phi 2(t-2)-\phi 2(t)+2\,\mathrm{D}(\phi 3)(t-1) \end{bmatrix}\right]$$

The matrix $R' := \mathit{Ext1}[2]$ satisfies $M/t(M) \cong \mathcal{O}^{1\times 3}/\mathcal{O}^{1\times 2}\,R'$. We denote this matrix by T in Maple:

```
> T := Ext1[2];
```

$$T := \begin{bmatrix} 1 & -1 & 0 \\ 0 & -\delta^2-1 & 2\,\mathrm{D}\,\delta \end{bmatrix}$$

Lemma 6.3.7 is easily verified: we find a matrix R'' in $\mathcal{O}^{2\times 2}$ such that $R = R''\,R'$.

```
> Factorize(R, T, Alg);
```

$$\begin{bmatrix} \delta^2 & -1 \\ 1 & -1 \end{bmatrix}$$

Let us check whether the short exact sequence

$$0 \longrightarrow t(M) \longrightarrow M \longrightarrow M/t(M) \longrightarrow 0$$

is split. By Lemma 6.3.6 this is equivalent to the existence of matrices $S \in \mathcal{O}^{3\times 2}$ and $V \in \mathcal{O}^{2\times 2}$ which fulfill $R' - R'\,S\,R' = V\,R$. The command `Complement` computes such matrices if they exist. More precisely, `Complement` returns a list of matrices U, V, S such that $U = I_3 - S\,R'$ and V and S satisfy $R' - R'\,S\,R' = V\,R$.

```
> C := Complement(T, R, Alg);
```

$$C := \left[\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ \frac{-1}{2} & \frac{-1}{2} \end{bmatrix}, \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{-1}{2} & 0 \\ 0 & 0 \end{bmatrix}\right]$$

Finally, we need to compute the syzygy module of the matrix $(T^T \quad R^T)^T$:

```
> L := SyzygyModule(linalg[stackmatrix](T, R), Alg);
```

$$L := \begin{bmatrix} 1 & -1 & 0 & -1 \\ 0 & \delta^2-1 & -1 & \delta^2 \end{bmatrix}$$

Let us denote by L' the submatrix of L formed by the first two columns. Then the application of Theorem 6.3.8 requires the solutions $\tau = (\tau 1 \quad \tau 2)^T$ of $L'\tau = 0$. The first equation in $L'\tau$ states that $\tau 1 = \tau 2$, and the second equation implies that $\tau 1$ is a 2-periodic function of t.

```
> tau := matrix(2, 1, [tau1(t),tau1(t)]);
```

$$\tau := \begin{bmatrix} \tau 1(t) \\ \tau 1(t) \end{bmatrix}$$

According to Theorem 6.3.8 we have a vector $\eta := P\,\xi + S\,\tau$ which gives all solutions of (6.2) in an injective signal space, when $\tau 1$ is a 2-periodic function and ξ is an arbitrary function.

```
> eta := ApplyMatrix(P, [xi(t)], Alg) +
> ApplyMatrix(C[3], tau, Alg);
```

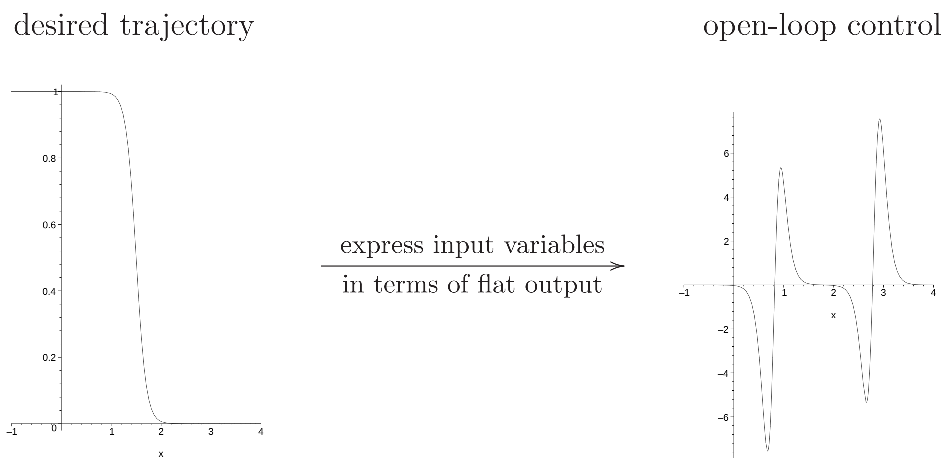
$$\eta := \begin{bmatrix} 2\,\mathrm{D}(\xi)(t-1) \\ 2\,\mathrm{D}(\xi)(t-1) \\ \xi(t) + \xi(t-2) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{2}\,\tau 1(t) \\ -\dfrac{1}{2}\,\tau 1(t) \\ 0 \end{bmatrix}$$

The command `Parametrization` as applied in the previous example uses `pdsolve` in Maple to solve partial differential equations, which arise as $L'\tau = 0$ in Theorem 6.3.8, symbolically. In the present case, `Parametrization` is not yet applicable because `pdsolve` does not handle retarded differential equations.

6.5 Flatness

The notion of flatness of linear systems is investigated more closely in this section. Stated in the framework of this chapter a linear system is flat if its behavior has an injective parametrization (see Def. 6.2.2). The problem of expressing the solutions of an underdetermined system of (nonlinear) partial differential equations in terms of arbitrary functions has already been studied in the 19th century and is known as Monge's problem (see [Zer32], [Jan71]). Important contributions were made by D. Hilbert [Hil12] and E. Cartan [Car14] for nonlinear systems of ordinary differential equations.

In the context of systems theory, the notion of flatness was introduced for continuous-time nonlinear systems in the beginning of the 1990s [FG93, FLMR95]. A given system is flat, if there exists a *flat output*, which is a (vector) function of the system variables (and of finitely many of their derivatives) such that conversely every system variable can (locally) be expressed as a function of the components of the flat output in a unique way. In particular, the components of a flat output are differentially independent. If a given system is flat and a flat output of the system is known, then it is particularly easy to design controls for the system. The problem of steering the system in such a way that the values of the flat output follow given trajectories is then solved as follows: the chosen inputs of the system are expressed in terms of the flat output which implies that the input trajectory is (locally) uniquely determined by the desired trajectory for the flat output. Sometimes this control strategy is sufficient and one renounces a feedback to improve the accuracy of the actual output trajectory in comparison to the desired trajectory ("open-loop control").



Many systems emerging naturally in applications are flat and even have flat outputs with a physical meaning so that the open-loop control strategy can be implemented in a straightforward way. Many applications are treated in the literature (e.g. [MRFR98], [Rot97], [OM02], [FPA03]).

In the rest of this section we incorporate the notion of flatness for linear systems into the present Ore algebra framework.

Let D be a left Noetherian Ore algebra and $\mathcal{F}$ a left D-module.

Definition 6.5.1. [FM98, Mou95] A behavior $\mathcal{B}$ is said to be *flat* if an injective parametrization $P \in D^{p \times m}$ of $\mathcal{B}$ exists, i.e. if there exist a parametrization $P \in D^{p \times m}$ of $\mathcal{B}$ and a matrix $T \in D^{m \times p}$ such that $T\,P = I_m$. In this case, we say that T defines a flat output ξ of $\mathcal{B}$ in the sense that $\eta = P\,\xi$ is equivalent to $\xi = T\,\eta$.

For an example we refer to Section 7.2.

Theorem 6.5.2. [FM98, Mou95, PQ99b] *Let D be a left Noetherian Ore algebra, $\mathcal{F}$ a left D-module which is an injective cogenerator for ${}_DM$, and $\mathcal{B}$ the behavior of a linear system $R\,\eta = 0$, where $R \in D^{q \times p}$. Let M be the left D-module associated with the system. The behavior $\mathcal{B}$ is flat if and only if the module M is free. Then the D-bases of M are in one-to-one correspondence with the flat outputs of $\mathcal{B}$.*

Proof. By definition of M the complex

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \longrightarrow M \longrightarrow 0 \tag{6.25}$$

is an exact sequence of left D-modules. Now $\mathcal{B}$ is flat if and only if there exists $P \in D^{p\times m}$ which induces an injective homomorphism $(P.) : \mathcal{F}^{m\times 1} \to \mathcal{F}^{p\times 1}$ such that

$$\mathcal{F}^{q\times 1} \xleftarrow{R.} \mathcal{F}^{p\times 1} \xleftarrow{P.} \mathcal{F}^{m\times 1} \longleftarrow 0 \tag{6.26}$$

is exact (see Remark 6.2.3). We recall what the property of $\mathcal{F}$ being an injective cogenerator for ${}_DM$ means for the application of $\hom_D(\,\cdot\,, \mathcal{F})$ to the complex

$$D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{.P} D^{1\times m} \longrightarrow 0. \tag{6.27}$$

This complex is exact if and only if (6.26) is exact. If $\mathcal{B}$ is flat, then (6.26) is exact and it follows $M \cong D^{1\times m}$, i.e. M is free. Conversely, if $M \cong D^{1\times m}$, then there exists a matrix $P \in D^{p\times m}$ such that (6.25) reads as the short exact sequence (6.27). Since $\mathcal{F}$ is an injective cogenerator, (6.26) is then exact. Hence, $\mathcal{B}$ is flat. The flat outputs are in one-to-one correspondence with the left inverses of P, which on the other hand are in bijection with the D-bases of M. □

Remark 6.5.3. In general, no necessary and sufficient condition expressed in terms of homological algebra is known for the freeness of a left D-module (see also Figure 5.1, p. 99). For specific k-algebras D, Theorem 4.2.9 (b) and (c) provide (in combination with Prop. 4.2.8) necessary and sufficient conditions for a finitely generated left D-module to be free.

Of course, in order to exploit the flatness of a behavior $\mathcal{B}$, it is not sufficient to prove that the associated left D-module is free, but a flat output of $\mathcal{B}$ also needs to be computed. According to the previous theorem, the flat outputs of $\mathcal{B}$ are in one-to-one correspondence with the D-bases of M. In general, a parametrization which is constructed as explained in Remark 6.2.9 is not minimal, i.e., the solutions of the given linear system are expressed in terms of arbitrary functions whose number is not minimal. Obviously, a parametrization is not injective, if it is not minimal. For more details about minimal parametrizations we refer to [CQR05], [QR06a].

For specific k-algebras D there are particular methods to find bases of finitely generated free modules:

- In the case of the localized Weyl algebra $B_1(k)$, where k is a field of characteristic zero, the Jacobson normal form (see Remark 2.4.11) can be used to find a basis.

- For commutative polynomial algebras Quillen-Suslin's theorem (see Theorem 4.2.9 (c)) states that projective modules are free. A constructive version of Quillen-Suslin's theorem is developed at Lehrstuhl B für Mathematik, RWTH Aachen [FQ06].

- In the next section we describe a method to compute bases of finitely generated free left modules over $A_n(k)$ or $B_n(k)$ of rank at least 2, where k is a field of characteristic zero.

We only mention here another characterization of controllable time-varying linear systems [QR05a]: every controllable time-varying linear system is a projection of a flat system.

6.6 Computing Bases of Free Modules over the Weyl Algebras

In this final section a method to compute bases of free left D-modules of rank at least 2 is presented, where D is either $A_n(k)$ or $B_n(k)$ for some $n \in \mathbb{N}$ and a field k of characteristic zero. This joint work with A. Quadrat [QR06a] relies on algorithmic versions of a theorem of J. T. Stafford developed in [HS01] and [Ley04]. First the notion of a stable unimodular column vector is defined. For such a column vector there exists a matrix representing elementary row operations such that the product of this matrix by the column vector is the first standard basis vector. Then we prove that every projective left D-module M which is the cokernel of an injective homomorphism $(.R) : D^{1\times p} \to D^{1\times q}$ with $p - q$ large enough is a free module. The proofs of these facts are combined in a procedure to compute a basis of M. Finally, an example demonstrates this procedure.

For the first part of this section we let D be any left Noetherian ring.

Definition 6.6.1. (a) A column vector $v = (v_1, \ldots, v_m)^T \in D^{m\times 1}$ is said to be *unimodular*, if there exists a row vector $w = (w_1, \ldots, w_m) \in D^{1\times m}$ such that $\sum_{i=1}^{m} w_i\, v_i = 1$. The set of unimodular column vectors in $D^{m\times 1}$ is denoted by $U_c(m, D)$.

(b) A unimodular column vector $v = (v_1, \ldots, v_m)^T \in U_c(m, D)$ is said to be *stable*, if there exist $a_1, \ldots, a_{m-1} \in D$ such that

$$v' := (v_1 + a_1\, v_m, \ldots, v_{m-1} + a_{m-1}\, v_m)^T \in U_c(m-1, D). \tag{6.28}$$

(c) The positive integer l is said to be *in the stable range of* D, if every unimodular column in $U_c(m, D)$ is stable for all $m \geq l$.

(d) The least positive integer l in the stable range of D is denoted by $\mathrm{sr}(D)$. If no such integer exists, then we set $\mathrm{sr}(D) = \infty$.

Remark 6.6.2. More generally than in Definition 6.6.1, the stable range is defined for left and right D-modules. The stable ranges of D considered either as left D-module or as right D-module in this more general sense are the same.

Example 6.6.3. (a) [MR00, Prop. 11.5.3] We have $\mathrm{sr}(\mathbb{Z}) = 2$ and $\mathrm{sr}(k[x]) = 2$ for any field k.

(b) [MR00, Cor. 11.5.10 (i)] For $D = \mathbb{R}[x_1, \ldots, x_n]$ we have $\mathrm{sr}(D) = n + 1$.

(c) [Sta78] If k is a field of characteristic zero, then we have $\mathrm{sr}(A_n(k)) = 2$ and $\mathrm{sr}(B_n(k)) = 2$.

Definition 6.6.4. [MR00] The *elementary group* $E(m, D)$ is the subgroup of $\mathrm{GL}(m, D)$ which is generated by all matrices of the form $I_m + r\, E_{i,j}$, $i \neq j$, $r \in D$, where $E_{i,j}$ denotes the $(m \times m)$-matrix whose only non-zero entry is at position (i, j) and this entry equals 1.

The next proposition states that every stable unimodular column vector in $D^{m\times 1}$ can be transformed to the first standard basis vector in $D^{m\times 1}$ by multiplying an appropriate matrix $E \in E(m, D)$ from the left. A constructive version of this proposition for the Weyl algebras $A_n(k)$ is an important ingredient for the algorithm which computes bases of free modules over $A_n(k)$ of rank at least 2.

Proposition 6.6.5. [QR06a] *If $v \in U_c(m, D)$ is stable, then there exists $E \in E(m, D)$ such that $E\, v = (1, 0, \ldots, 0)^T$.*

Proof. Since v is stable, there exist $a_1, \ldots, a_{m-1} \in D$ such that (6.28) holds. If we denote the entries of v' by v'_i, $1 \leq i \leq m - 1$, then the matrix

$$E_1 := \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & a_1 \\ 0 & 1 & 0 & \ldots & 0 & a_2 \\ 0 & 0 & 1 & \ldots & 0 & a_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & a_{m-1} \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix} \in E(m, D)$$

satisfies $E_1\, v = (v'_1, \ldots, v'_{m-1}, v_m)^T$. Since v' is a unimodular column vector, there exist $b_1, \ldots, b_{m-1} \in D$ such that $\sum_{i=1}^{m-1} b_i\, v'_i = 1$. We define $c_i := (v'_1 - 1 - v_m)\, b_i$, $i = 1, \ldots, m - 1$. Then the matrix

$$E_2 := \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ c_1 & c_2 & c_3 & \ldots & c_{m-1} & 1 \end{pmatrix} \in E(m, D)$$

satisfies $E_2\,E_1\,v = (v'_1, \ldots, v'_{m-1}, v'_1 - 1)^T$. The first entry of the first standard basis vector in $D^{m\times 1}$ is now produced by the matrix

$$E_3 := \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & -1 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix} \in E(m, D),$$

namely we have $E_3\,E_2\,E_1\,v = (1, v'_2, \ldots, v'_{m-1}, v'_1 - 1)^T$. Finally, using the matrix

$$E_4 := \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ -v'_2 & 1 & 0 & \ldots & 0 & 0 \\ -v'_3 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -v'_{m-1} & 0 & 0 & \ldots & 1 & 0 \\ -v'_1 + 1 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix} \in E(m, D)$$

we obtain $E := E_4\,E_3\,E_2\,E_1 \in E(m, D)$ which fulfills the assertion. □

As soon as $a_1, \ldots, a_{m-1} \in D$ can be found constructively such that (6.28) holds, the proof of the previous proposition can be translated into an algorithm. We are going to show below how this can be done for the Weyl algebras $A_n(k)$ and $B_n(k)$ for a field k of characteristic zero. Bases of free left D-modules of rank at least $\mathrm{sr}(D)$ can be computed along the lines of the proof of the following theorem. The previous proposition is applied iteratively.

Theorem 6.6.6. [QR06a] *Let k be a field and D a (not necessarily commutative) k-algebra with involution θ. Then every projective left D-module M having a free resolution of the form*

$$0 \longrightarrow D^{1\times q} \xrightarrow{.R} D^{1\times p} \longrightarrow M \longrightarrow 0 \tag{6.29}$$

with $p - q \geq \mathrm{sr}(D)$ is free.

Remark 6.6.7. In fact, the assumption on the left D-module M in the theorem states that M is *stably-free*, which by definition means that the direct sum of M and some free left D-module is free (cf. Remark 6.3.2 (a)). In the present situation we have $M \oplus D^{1\times q} \cong D^{1\times p}$. Of course, every free module is stably-free, and every stably-free module is projective. If M is a finitely generated stably-free left module over an Ore algebra of the type considered e.g. in Section 2.4, then a free resolution of M as in (6.29) can be constructed from any given finite presentation of M (cf. [QR06a]).

Proof of Thm. 6.6.6. Since M is projective, the short exact sequence (6.29) is split (see Remark 6.3.5), i.e. there exists $S \in D^{p\times q}$ such that $R\,S = I_q$. In order to be able to apply Proposition 6.6.5, we consider $\theta(R) \in D^{p\times q}$ (cf. Def. 5.2.5) and the short exact sequence

$$0 \longleftarrow D^{1\times q} \xleftarrow{.\theta(R)} D^{1\times p} \longleftarrow \ker(.\theta(R)) \longleftarrow 0,$$

which is split because $\theta(S)\,\theta(R) = \theta(R\,S) = I_q$. Therefore, the first column v of $\theta(R)$ is a unimodular column vector with p entries. Now v is stable because $p > \mathrm{sr}(D)$. By Proposition 6.6.5 there exists a matrix $E \in E(p, D)$ such that $E\,v = (1, 0, \ldots, 0)^T$, and hence

$$E\,\theta(R) = \begin{pmatrix} 1 & \star \\ 0 & \\ \vdots & T \\ 0 & \end{pmatrix} \tag{6.30}$$

for some matrix $T \in D^{(p-1)\times(q-1)}$, where $\star$ denotes an appropriate number of elements in D. We have $(\theta(S)\,E^{-1})\,(E\,\theta(R)) = I_q$, and obviously every left inverse of $E\,\theta(R)$ is of the form

$$\begin{pmatrix} 1 & \star \\ 0 & \\ \vdots & L \\ 0 & \end{pmatrix} \in D^{(q-1)\times(p-1)}.$$

Hence, by removing the first row and the first column of $\theta(S)\,E^{-1}$ we obtain a left inverse of T. Therefore, the first column of T is a unimodular column vector with $p-1$ entries. Since $p - 1 \geq \mathrm{sr}(D)$, there exists a matrix $E' \in E(p-1, D)$ such that $E'\,T$ has the same shape as the matrix in (6.30), but having only $p-1$ rows and $q-1$ columns. The above argument can be repeated as long as the number of entries is greater than or equal to $\mathrm{sr}(D)$. Iteratively we conclude that there exist matrices $E_1 := E$, $E_2 := \mathrm{diag}(1, E')$, $\ldots$, $E_q \in E(p, D)$ such that $F := E_q \cdots E_1$ satisfies

$$F\,\theta(R) = \begin{pmatrix} 1 & \star & \ldots & \star \\ 0 & 1 & \star & \vdots \\ \vdots & \vdots & \ddots & \star \\ 0 & 0 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}.$$

Now it is clear that $\ker(.(F\,\theta(R))) = \operatorname{im}(.P)$ with $P := (0 \quad I_{p-q}) \in D^{(p-q)\times p}$. So we have $\ker(.\theta(R)) = \operatorname{im}(.(P\,F))$ because F is invertible, i.e.

$$0 \longleftarrow D^{1\times q} \xleftarrow{.\theta(R)} D^{1\times p} \xleftarrow{.(P\,F)} D^{1\times(p-q)} \longleftarrow 0$$

is a short exact sequence. If we define $Q := \theta(P\,F) \in D^{p\times(p-q)}$, then we have the short exact sequence:

$$0 \longrightarrow D^{1\times q} \xrightarrow{.R} D^{1\times p} \xrightarrow{.Q} D^{1\times(p-q)} \longrightarrow 0,$$

which is split because $R\,S = I_q$ (see Remark 6.3.2 (a)). Hence, there exists a matrix $L \in D^{(p-q)\times p}$ such that $L\,Q = I_{p-q}$. We conclude that

$$M = D^{1\times p}/D^{1\times q}\,R \cong D^{1\times p}\,Q = D^{1\times(p-q)}$$

is free, and a basis of M is given by the residue classes in M of the rows of L. $\square$

In the preceding proof Proposition 6.6.5 is applied. We quote the following theorem of J. T. Stafford which will allow us to turn Proposition 6.6.5 into an effective procedure. Then the proof of Theorem 6.6.6 describes a procedure to construct bases of free modules over the Weyl algebras of rank at least 2.

Theorem 6.6.8 (Stafford)**.** [Sta78] *Let k be a field of characteristic zero and $D \in \{A_n(k),\, B_n(k)\}$. Let $a, b, c \in D$ generate the left ideal I of D, i.e.*

$$I = D\,a + D\,b + D\,c,$$

and let $d_1, d_2 \in D - \{0\}$ be arbitrary. Then there exist $f, g \in D$ such that

$$I = D\,(a + d_1\,f\,c) + D\,(b + d_2\,g\,c). \tag{6.31}$$

Remark 6.6.9. Most interesting in the present context is the special case $d_1 = d_2 = 1$. Then the theorem simply states that every left ideal of $A_n(k)$ (and of $B_n(k)$) can be generated by two elements and, moreover, given three generators of the left ideal, a generating set exists that consists of only two appropriate linear combinations of the three generators as formed in (6.31).

Remark 6.6.10. Let $D \in \{A_n(k),\, B_n(k)\}$, where k is a field of characteristic zero. Since $\operatorname{sr}(D) = 2$ (see Ex. 6.6.3 (c)), every unimodular column vector $v = (v_1, \ldots, v_m)^T \in U_c(m, D)$ is stable, i.e. there exist $a_1, \ldots, a_{m-1} \in D$ such that $v' = (v_1 + a_1\,v_m, \ldots, v_{m-1} + a_{m-1}\,v_m)^T$ is unimodular. Appropriate coefficients $a_i \in D$ can be computed by applying a constructive version of Theorem 6.6.8 (see the next remark). To this end, choose three distinct entries of v, say v_1, v_2, v_m, and consider the left ideal $I = D\,v_1 + D\,v_2 + D\,v_m$. Then a constructive version of Theorem 6.6.8 (for the case $d_1 = d_2 = 1$) yields a_1, $a_2 \in D$ such that

$I = D\,(v_1 + a_1\, v_m) + D\,(v_2 + a_2\, v_m)$. Now, $v' = (v_1 + a_1\, v_m, v_2 + a_2\, v_m, v_3, \ldots, v_{m-1})^T$ is unimodular because by assumption

$$D = \sum_{i=1}^{m} D\, v_i = I + \sum_{i=3}^{m-1} D\, v_i.$$

Remark 6.6.11. Algorithmic versions of Theorem 6.6.8 have been developed recently in [HS01] and [Ley04]. Using these constructive methods, an implementation of the described procedure to compute bases of free left modules over $A_n(k)$ or $B_n(k)$ of rank at least 2 has been included in the Maple package `OreModules` (see Section 5.5) by A. Quadrat and the author of this thesis. It is based on the proofs of Prop. 6.6.5 and Theorem 6.6.6 and Remark 6.6.10. The implementation has been applied successfully to small examples. However, the bottleneck are the constructive versions of Stafford's theorem which require a lot of Gröbner / Janet basis computations.

Let us demonstrate the procedure to compute bases of free left modules over the Weyl algebras of rank at least 2 on an example.

Example 6.6.12. [QR06a] We consider $D = A_3(\mathbb{Q})$ with the involution θ defined in Ex. 5.2.3 (a),

$$R = (-\partial_1 + x_3 \quad -\partial_2 \quad -\partial_3) \in D^{p \times q}, \qquad p = 1, \quad q = 3,$$

and $M = D^{1\times 3}/D\,R$. Then $S = (\partial_3 \quad 0 \quad \partial_1 - x_3)^T$ satisfies $R\,S = 1$. Therefore the short exact sequence

$$0 \longrightarrow D \xrightarrow{.R} D^{1\times 3} \longrightarrow M \longrightarrow 0$$

splits and M is projective. Moreover, $p - q = 2 = \mathrm{sr}(D)$. By Theorem 6.6.6, M is free.

Now $\theta(R) = (\partial_1 + x_3 \quad \partial_2 \quad \partial_3)^T$ is a unimodular column vector because $\theta(S)\,\theta(R) = 1$. Without describing the details of the algorithmic versions of Stafford's theorem, we show that

$$D\,(\partial_1 + x_3) + D\,\partial_2 + D\,\partial_3 = D\,(\partial_1 + x_3) + D\,(\partial_2 + \partial_3). \tag{6.32}$$

In fact, the left ideal in D generated by the entries a, b, c of $\theta(R)$ equals D because $\partial_3\,(\partial_1 + x_3) + (-\partial_1 - x_3)\,\partial_3 = 1$. On the other hand, we have

$$(\partial_2 + \partial_3)\,(\partial_1 + x_3) + (-\partial_1 - x_3)\,(\partial_2 + \partial_3) = 1, \tag{6.33}$$

which shows that (6.32) holds. Therefore, we can choose $f = 0$, $g = 1$ in Stafford's theorem (in the case $d_1 = d_2 = 1$). Following the proof of Prop. 6.6.5, we define the matrix

$$E_1 := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \in E(3, D).$$

We obtain $E_1\,\theta(R) = (\partial_1 + x_3 \quad \partial_2 + \partial_3 \quad \partial_3)^T$. By (6.33) we are led to define (see the proof of Prop. 6.6.5)

$$c_1 := (\partial_1 + x_3 - 1 - \partial_3)\,(\partial_2 + \partial_3), \qquad c_2 := (\partial_1 + x_3 - 1 - \partial_3)\,(-\partial_1 - x_3)$$

and

$$E_2 := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_1 & c_2 & 1 \end{pmatrix} \in E(3, D).$$

We obtain $E_2\,E_1\,\theta(R) = (\partial_1 + x_3 \quad \partial_2 + \partial_3 \quad \partial_1 + x_3 - 1)^T$. Finally, we define

$$E_3 := \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad E_4 := \begin{pmatrix} 1 & 0 & 0 \\ -\partial_2 - x_3 & 1 & 0 \\ -\partial_1 - x_3 + 1 & 0 & 1 \end{pmatrix} \in E(3, D)$$

and $F := E_4\,E_3\,E_2\,E_1$ which satisfies $F\,\theta(R) = (1, 0, 0)^T$. Now we compute a left inverse $L \in D^{2\times 3}$ of the matrix Q which is formed by the last two columns of $\theta(F)$. The procedure `LeftInverse` of the `OreModules` package yields:

$$L = \left(\begin{matrix} 0 & 1 + x_3 + 2\,\partial_1\,x_3 - x_3^2 - \partial_1^2 - \partial_3\,x_3 - \partial_1 + \partial_1\,\partial_3 \\ 1 & -2 - \partial_2\,x_3 - \partial_3\,x_3 + \partial_2 + \partial_3 + \partial_1\,\partial_3 - \partial_3\,\partial_2 - \partial_3^2 + \partial_1\,\partial_2 \end{matrix}\right.$$
$$\left.\begin{matrix} -x_3 - 2\,\partial_1\,x_3 + x_3^2 + \partial_1^2 + \partial_3\,x_3 + \partial_1 - \partial_1\,\partial_3 \\ 2 - \partial_1\,\partial_2 - \partial_1\,\partial_3 + \partial_2\,x_3 + \partial_3\,x_3 - \partial_2 - \partial_3 + \partial_3\,\partial_2 + \partial_3^2 \end{matrix}\right)$$

The residue classes of the rows of L form a basis of M.

Chapter 7

A Stirred Tank Model

In this chapter we demonstrate the algebraic approach to the structural analysis of linear systems on an example of a stirred tank model. The example is taken from [KS72] (see Example 1.2). We apply the Maple packages `OreModules` (see Section 5.5) and `JanetOre` (see Section 2.6) in order to perform the necessary computations. In the following sections more system theoretic concepts are addressed than have been introduced in the previous chapters.

First a linearization of the system of ordinary differential equations modelling the stirred tank is introduced in Section 7.1. Controllability of this linear system is checked in Section 7.2 by computing a presentation of the torsion submodule of the module which is associated with the system. In [KS72] (Ex. 1.21) the controllability is checked by computing the rank of the controllability matrix, which can also be done with `OreModules` (end of Section 7.3). In addition, we study parametrizability and flatness of the stirred tank in Section 7.2.

For certain configurations of the parameters of the stirred tank model the system is not completely controllable. An autonomous observable of the stirred tank for a particular configuration of the parameters is found in [KS72] (Ex. 1.19) by precise inspection of the system equations and by physical intuition, whereas the same autonomous observable is found in Section 7.3 automatically by applying Algorithm 5.2.8 for the computation of extension groups. The controllable part of the system is examined for these configurations of the parameters.

Section 7.4 first defines output variables for the system. Then observability of the stirred tank and the relations between the input and the output variables are investigated.

As explained in Section 6.5, flatness is a particularly useful property for controlling a system in such a way that interesting quantities follow a prescribed trajectory. This concept is illustrated in Section 7.5.

In Section 7.6 a linear quadratic optimal control problem as posed in [KS72], Example 3.9, is treated. Whereas the problem is solved in [KS72] by deriving Riccati equations whose solutions define the optimal feedback, we apply a method proposed by J.-F. Pommaret and A. Quadrat [PQ04] which amounts to substituting a parametrization of the system into the cost functional and thereby obtaining a variational problem without constraints. The Euler-Lagrange equations for this variational problem then give necessary conditions for optimality.

Finally, the stirred tank model is modified in two ways as shown in [KS72]. First a discrete-time description of the stirred tank is investigated in Section 7.7. We study controllability, parametrizability and flatness for this discrete linear system. In the final section a delay is incorporated into the stirred tank model. The structural properties of the resulting differential time-delay system are examined in a similar manner as previously, but obstructions towards flatness of the system are considered more closely.

7.1 Introduction

The tank under consideration is a container of some fluid, fed with two incoming flows of fluids and having one outgoing flow. We denote by $F_1(t)$, $F_2(t)$ the flow rates of the incoming flows and by $F(t)$ the flow rate of the outgoing one. The concentrations c_1, c_2 of certain dissolved materials in the incoming flows are assumed to be constant. The concentration of interest in the outgoing flow is denoted by $c(t)$. Inside the tank the fluid is supposed to be stirred well so that $c(t)$ equals the according concentration in the tank. Finally, the tank is assumed to have a constant cross-sectional area S.

In [KS72] the following mass balance equations are derived:

$$\left\{\begin{array}{rcl} \dot{V}(t) & = & F_1(t) + F_2(t) - k\sqrt{\frac{V(t)}{S}}, \\ \dot{\overline{c(t)\,V(t)}} & = & c_1\,F_1(t) + c_2\,F_2(t) - c(t)\,k\sqrt{\frac{V(t)}{S}}, \end{array}\right.$$

where V is the volume of the fluid in the tank and k is an experimental constant. Linearization around a steady-state with fluid volume V_0 and concentration c_0 of the outgoing flow yields the following model of linear ordinary differential equations for the stirred tank:

$$\dot{x}(t) = \begin{pmatrix} -\frac{1}{2\theta} & 0 \\ 0 & -\frac{1}{\theta} \end{pmatrix} x(t) + \begin{pmatrix} 1 & 1 \\ \frac{c_1-c_0}{V_0} & \frac{c_2-c_0}{V_0} \end{pmatrix} u(t). \tag{7.1}$$

The components of x respectively u are the deviations of $(V(t), c(t))$ respectively $(F_1(t), F_2(t))$ from the steady-state and $\theta := V_0/(k\sqrt{\frac{V_0}{S}})$ is the holdup time of the tank.

7.2 Controllability, parametrizability, flatness

```
> with(OreModules):
> with(JanetOre):
```

We select the Maple package `JanetOre` to perform the necessary Janet basis computations.

```
> OreModulesOptions("GroebnerBasis", "JanetOre"):
```

First we study the structural properties of the system describing the stirred tank for generic parameters, i.e. in this section we check whether the system is controllable, parametrizable and flat.

The equations describing the behavior of the stirred tank have constant coefficients, i.e. we deal with a time-invariant system. Therefore we consider modules over the commutative polynomial algebra

$$\mathcal{O} := \mathbb{Q}(\theta, V0, c0, c1, c2)[\mathrm{D}],$$

where D represents differentiation with respect to t. In order to apply `OreModules` to the given system, we declare the Ore algebra $Alg := \mathbb{Q}(\theta, V0, c0, c1, c2)[t][\mathrm{D}]$ of ordinary differential operators with polynomial coefficients in t which have coefficients that are rational functions in θ, $V0$, $c0$, $c1$, $c2$. Obviously, Alg contains $\mathcal{O}$ as subring.

```
> Alg := DefineOreAlgebra(diff=[D,t], polynom=[t],
> comm=[theta,V0,c0,c1,c2]):
```

The linear system (7.1) of ordinary differential equations is entered as follows:

```
> R := matrix([[D+1/(2*theta),0,-1,-1],
> [0,D+1/theta,-(c1-c0)/V0,-(c2-c0)/V0]]);
```

$$R := \begin{bmatrix} \mathrm{D} + \dfrac{1}{2\,\theta} & 0 & -1 & -1 \\ 0 & \mathrm{D} + \dfrac{1}{\theta} & -\dfrac{c1 - c0}{V0} & -\dfrac{c2 - c0}{V0} \end{bmatrix}$$

The equations are retrieved by applying the matrix R of operators to the column vector of system variables:

```
> ApplyMatrix(R, vector([x1(t),x2(t),u1(t),u2(t)]), Alg)=
> matrix([[0]$2]);
```

$$\left[\begin{array}{c} -\dfrac{1}{2}\,\dfrac{-\mathrm{x1}(t)-2\,(\frac{d}{dt}\,\mathrm{x1}(t))\,\theta+2\,\mathrm{u1}(t)\,\theta+2\,\mathrm{u2}(t)\,\theta}{\theta} \\ \dfrac{\mathrm{x2}(t)\ \mathit{V0}+(\frac{d}{dt}\,\mathrm{x2}(t))\,\theta\ \mathit{V0}-\mathrm{u1}(t)\,\theta\,\mathit{c1}+\mathrm{u1}(t)\,\theta\,\mathit{c0}-\mathrm{u2}(t)\,\theta\,\mathit{c2}+\mathrm{u2}(t)\,\theta\,\mathit{c0}}{\theta\ \mathit{V0}} \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \end{array}\right]$$

Let $M = \mathcal{O}^{1\times 4}/\mathcal{O}^{1\times 2}\,R$ be the $\mathcal{O}$-module which is associated with the linear system. Since we consider a linear system with constant coefficients, the transposed module $M^\top$ is presented by the transposed matrix of R (see Def. 5.1.4), i.e. $M^\top \cong \mathcal{O}^{1\times 2}/\mathcal{O}^{1\times 4}\,R^T$. For matrices with entries in commutative Ore algebras the standard involution is just transposition. Therefore, the transposed module $M^\top$ of M is presented by:

```
> R_adj := linalg[transpose](R);
```

$$\mathit{R\_adj} := \left[\begin{array}{cc} \mathrm{D}+\dfrac{1}{2\,\theta} & 0 \\ 0 & \mathrm{D}+\dfrac{1}{\theta} \\ -1 & -\dfrac{\mathit{c1}-\mathit{c0}}{\mathit{V0}} \\ -1 & -\dfrac{\mathit{c2}-\mathit{c0}}{\mathit{V0}} \end{array}\right]$$

We are going to check whether the system which describes the stirred tank is parametrizable. Since the system is given by ordinary differential equations with constant coefficients, parametrizability is equivalent to flatness (Prop. 4.2.8 and Thm. 4.2.9 (b)). The system is parametrizable if and only if $\mathrm{ext}^1_{\mathcal{O}}(M^\top,\mathcal{O}) \cong t(M)$ (see Thm. 5.1.5) is zero. This can be checked by computing the extension group $\mathrm{ext}^1_{\mathcal{O}}(M^\top,\mathcal{O})$ of the transposed module $M^\top$:

```
> Ext := Exti(R_adj, Alg, 1);
```

$$\mathit{Ext} := \left[\left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right], \left[\begin{array}{cccc} 2\,\mathrm{D}\,\theta+1 & 0 & -2\,\theta & -2\,\theta \\ 0 & -\theta\ \mathit{V0}\,\mathrm{D}-\mathit{V0} & -\theta\,(\mathit{c0}-\mathit{c1}) & -\theta\,(\mathit{c0}-\mathit{c2}) \end{array}\right],\right.$$
$$\left.\left[\begin{array}{c} -2\,\theta\ \mathit{V0}\,(\mathit{c1}-\mathit{c2}),\ 2\,\theta\ \mathit{V0}\,(\mathit{c1}-\mathit{c2}) \\ 2\,\theta\,(\mathit{c0}-\mathit{c2})\,(\mathit{c1}-\mathit{c2}),\ -2\,\theta\,(-\mathit{c2}\,\mathit{c0}+\mathit{c2}\,\mathit{c1}+\mathit{c1}\,\mathit{c0}-\mathit{c1}^2) \\ \mathit{V0}\,(\mathit{c0}-\mathit{c2}),\ -\mathit{V0}\,(\mathit{c2}+\mathit{c0}-2\,\mathit{c1})+2\,(\mathit{c1}-\mathit{c2})\,\theta\ \mathit{V0}\,\mathrm{D} \\ -\mathit{V0}\,(-2\,\mathit{c2}+\mathit{c0}+\mathit{c1})-2\,(\mathit{c1}-\mathit{c2})\,\theta\ \mathit{V0}\,\mathrm{D},\ \mathit{V0}\,(\mathit{c0}-\mathit{c1}) \end{array}\right]\right]$$

The residue classes in M represented by the rows of the second matrix in Ext generate $t(M)$. For $i = 1, 2$, the entries of the i-th column of the first matrix in Ext generate the ideal of $\mathcal{O}$ whose elements annihilate the i-th generator given in $\mathit{Ext}[2]$, i.e., if we denote the residue class in M of the i-th row in $\mathit{Ext}[2]$ by m_i, then we have $1 \cdot m_1 = 0$ and $1 \cdot m_2 = 0$ in M. Therefore, $t(M) = 0$. Hence, the stirred tank is generically parametrizable, and thus, generically flat. A generic parametrization of the module M is given by $P := Ext[3]$. In the notation of Algorithm 5.2.8 we have $Ext[2] = K$ and $Ext[3] = \theta(S_{i+1})$ for $i = 1$.

```
> P := evalm(Ext[3]):
```

When using the command `Parametrization` of `OreModules` this parametrization is applied to the vector $(\xi_1 \quad \xi_2)^T$ whose entries represent arbitrary (smooth) functions of t, and the result is returned:

```
> matrix([[x1(t)],[x2(t)],[u1(t)],[u2(t)]])=
> Parametrization(R, Alg);
```

$$\begin{bmatrix} \mathrm{x1}(t) \\ \mathrm{x2}(t) \\ \mathrm{u1}(t) \\ \mathrm{u2}(t) \end{bmatrix} =$$

$$[2\,\theta\, V0\, \xi_1(t)\, c2 - 2\,\theta\, V0\, \xi_1(t)\, c1 - 2\,\theta\, V0\, \xi_2(t)\, c2 + 2\,\theta\, V0\, \xi_2(t)\, c1]$$

$$\Big[2\,\theta\, \xi_1(t)\, c2^2 - 2\,\theta\, \xi_1(t)\, c2\, c1 - 2\,\theta\, \xi_1(t)\, c2\, c0 + 2\,\theta\, \xi_1(t)\, c1\, c0 + 2\,\theta\, \xi_2(t)\, c2\, c0$$
$$- 2\,\theta\, \xi_2(t)\, c2\, c1 - 2\,\theta\, \xi_2(t)\, c1\, c0 + 2\,\theta\, \xi_2(t)\, c1^2\Big]$$

$$\Big[V0(-\xi_1(t)\, c2 + \xi_1(t)\, c0 - \xi_2(t)\, c2 - \xi_2(t)\, c0 + 2\,\xi_2(t)\, c1 - 2\,\theta\, (\tfrac{d}{dt}\, \xi_2(t))\, c2$$
$$+ 2\,\theta\, (\tfrac{d}{dt}\, \xi_2(t))\, c1)\Big]$$

$$\Big[-\, V0(\xi_1(t)\, c0 - 2\,\xi_1(t)\, c2 + \xi_1(t)\, c1 - 2\,\theta\, (\tfrac{d}{dt}\, \xi_1(t))\, c2 + 2\,\theta\, (\tfrac{d}{dt}\, \xi_1(t))\, c1$$
$$+ \xi_2(t)\, c1 - \xi_2(t)\, c0)\Big]$$

We have just proved that M is torsion-free. Since this result was obtained by a generic computation, it may be wrong for certain specializations of the parameters of the system. We are going to investigate this below.

The algebra $\mathcal{O}$ is a principal ideal domain. By Theorem 4.2.9 (b) every torsion-free $\mathcal{O}$-module is free and, in particular, projective. Let us demonstrate another way to check whether M is projective or not. First we note that the matrix R which presents M induces an injective homomorphism $.R$:

```
> SyzygyModule(R, Alg);
```

$$\mathrm{INJ}(2)$$

We see that there is no non-trivial relation between the rows of R, i.e. R has full row rank. Now, M is projective if and only if the short exact sequence

$$0 \longrightarrow \mathcal{O}^{1\times 2} \xrightarrow{.R} \mathcal{O}^{1\times 4} \longrightarrow M \longrightarrow 0 \tag{7.2}$$

is split (see Def. 4.2.6 and Def. 6.3.1). Since $.R$ is injective, (7.2) is split if and only if there exists a matrix T in $\mathcal{O}^{4\times 2}$ such that $R\,T = I_2$ (see Rem. 6.3.2 (a)).

```
> RightInverse(R, Alg);
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\dfrac{c0 - c2}{c1 - c2} & -\dfrac{V0}{c1 - c2} \\ \dfrac{c0 - c1}{c1 - c2} & \dfrac{V0}{c1 - c2} \end{bmatrix}$$

Hence, there exists T in $\mathcal{O}^{4\times 2}$ satisfying $R\,T = I_2$. Therefore, M is projective. If values are assigned to the parameters of the system, then the associated module is projective if and only if $c1 \neq c2$.

As we have already seen above, M is free. This also follows from the Quillen-Suslin Theorem (see Thm. 4.2.9 (c)), because $\mathcal{O}$ is a commutative polynomial algebra over a field. Hence, the stirred tank is a flat system (see Thm. 6.5.2), and all specializations of the parameters lead to flat systems except for the case $c1 = c2$.

We can compute a generic flat output of the behavior of the stirred tank by computing a matrix S in $\mathcal{O}^{2\times 4}$ such that $S\,P = I_2$, where P is the parametrization of the module M obtained above.

```
> S := map(factor, LeftInverse(P, Alg));
```

$$S := \begin{bmatrix} \dfrac{c0 - c1}{2\,\theta\,V0\,(c1 - c2)^2} & \dfrac{1}{2\,\theta\,(c1 - c2)^2} & 0 & 0 \\ \dfrac{c0 - c2}{2\,\theta\,V0\,(c1 - c2)^2} & \dfrac{1}{2\,\theta\,(c1 - c2)^2} & 0 & 0 \end{bmatrix}$$

A flat output of the behavior of the stirred tank is defined by

$$(\xi 1 \quad \xi 2)^T = S\,(x1 \quad x2 \quad u1 \quad u2)^T,$$

and $(\xi 1 \quad \xi 2)^T$ satisfies $(x1 \quad x2 \quad u1 \quad u2)^T = P\,(\xi 1 \quad \xi 2)^T$. Again, this is a result which holds for generic configurations of θ, $V0$, $c0$, $c1$, $c2$ only. More precisely, we have found a flat output for every specialization of the parameters except when $c1 = c2$. (Of course, we assume $\theta \neq 0$ and $V0 \neq 0$.) The flat output can be displayed in a more readable way as follows:

```
> matrix([[xi1(t)],[xi2(t)]])=
> ApplyMatrix(S, [x1(t),x2(t),u1(t),u2(t)], Alg);
```

$$\begin{bmatrix} \xi 1(t) \\ \xi 2(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\,\frac{\mathrm{x1}(t)\,c0 - \mathrm{x1}(t)\,c1 + \mathrm{x2}(t)\,V0}{\theta\,V0\,(c1-c2)^2} \\ \frac{1}{2}\,\frac{\mathrm{x1}(t)\,c0 - \mathrm{x1}(t)\,c2 + \mathrm{x2}(t)\,V0}{\theta\,V0\,(c1-c2)^2} \end{bmatrix}$$

If we apply the parametrization P to the flat output, we get the following matrix:

```
> Flat := Mult(P, S, Alg);
```

$$\mathit{Flat} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{(2\,\mathrm{D}\,\theta+1)\,(c0-c2)}{2\,\theta\,(c1-c2)} & \frac{(1+\mathrm{D}\,\theta)\,V0}{(c1-c2)\,\theta} & 0 & 0 \\ -\frac{(2\,\mathrm{D}\,\theta+1)\,(c0-c1)}{2\,\theta\,(c1-c2)} & -\frac{(1+\mathrm{D}\,\theta)\,V0}{(c1-c2)\,\theta} & 0 & 0 \end{bmatrix}$$

This matrix shows that we can also consider $(x1 \quad x2)^T$ as a flat output of the system. (The system variables $u1$ and $u2$ are expressed as linear combinations of $x1$ and $x2$ with coefficients given in the third resp. fourth row of *Flat*.) We have the following parametrization:

```
> matrix([[x1(t)],[x2(t)],[u1(t)],[u2(t)]])=
> ApplyMatrix(Flat, [x1(t),x2(t),u1(t),u2(t)], Alg);
```

$$\begin{bmatrix} \mathrm{x1}(t) \\ \mathrm{x2}(t) \\ \mathrm{u1}(t) \\ \mathrm{u2}(t) \end{bmatrix} =$$

$$[\mathrm{x1}(t)]$$

$$[\mathrm{x2}(t)]$$

$$\left[\frac{1}{2}\,\frac{\mathrm{x1}(t)\,c0 - \mathrm{x1}(t)\,c2 - 2\,(\frac{d}{dt}\,\mathrm{x1}(t))\,\theta\,c2 + 2\,(\frac{d}{dt}\,\mathrm{x1}(t))\,\theta\,c0 + 2\,\mathrm{x2}(t)\,V0 + 2\,V0\,(\frac{d}{dt}\,\mathrm{x2}(t))\,\theta}{(c1-c2)\,\theta}\right]$$

$$\left[-\frac{1}{2}\,\frac{\mathrm{x1}(t)\,c0 - \mathrm{x1}(t)\,c1 - 2\,(\frac{d}{dt}\,\mathrm{x1}(t))\,\theta\,c1 + 2\,(\frac{d}{dt}\,\mathrm{x1}(t))\,\theta\,c0 + 2\,\mathrm{x2}(t)\,V0 + 2\,V0\,(\frac{d}{dt}\,\mathrm{x2}(t))\,\theta}{(c1-c2)\,\theta}\right]$$

Let us compute a change of variables which transforms R into Brunovský canonical form [Son98]:

```
> Br := Brunovsky(R, Alg);
```

$$Br := \begin{bmatrix} \frac{c0 - c1}{2\,\theta\,V0\,(c1-c2)^2} & \frac{1}{2\,\theta\,(c1-c2)^2} & 0 & 0 \\ -\frac{c0-c1}{4\,\theta^2\,V0\,(c1-c2)^2} & -\frac{1}{2\,\theta^2\,(c1-c2)^2} & 0 & -\frac{1}{2\,\theta\,V0\,(c1-c2)} \\ \frac{c0-c2}{2\,\theta\,V0\,(c1-c2)^2} & \frac{1}{2\,\theta\,(c1-c2)^2} & 0 & 0 \\ -\frac{c0-c2}{4\,\theta^2\,V0\,(c1-c2)^2} & -\frac{1}{2\,\theta^2\,(c1-c2)^2} & \frac{1}{2\,\theta\,V0\,(c1-c2)} & 0 \end{bmatrix}$$

The matrix Br defines the following transformation of the variables:

```
> matrix([[z1(t)],[v1(t)],[z2(t)],[v2(t)]])=
> ApplyMatrix(Br, [x1(t),x2(t),u1(t),u2(t)], Alg);
```

$$\begin{bmatrix} \mathrm{z1}(t) \\ \mathrm{v1}(t) \\ \mathrm{z2}(t) \\ \mathrm{v2}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\,\frac{-\mathrm{x1}(t)\,c1 + \mathrm{x1}(t)\,c0 + \mathrm{x2}(t)\,V0}{\theta\,V0\,(c1-c2)^2} \\ -\frac{1}{4}\,\frac{-\mathrm{x1}(t)\,c1 + \mathrm{x1}(t)\,c0 + 2\,\mathrm{x2}(t)\,V0 - 2\,\mathrm{u2}(t)\,\theta\,c2 + 2\,\mathrm{u2}(t)\,\theta\,c1}{\theta^2\,V0\,(c1-c2)^2} \\ \frac{1}{2}\,\frac{-\mathrm{x1}(t)\,c2 + \mathrm{x1}(t)\,c0 + \mathrm{x2}(t)\,V0}{\theta\,V0\,(c1-c2)^2} \\ \frac{1}{4}\,\frac{\mathrm{x1}(t)\,c2 - \mathrm{x1}(t)\,c0 - 2\,\mathrm{x2}(t)\,V0 - 2\,\mathrm{u1}(t)\,\theta\,c2 + 2\,\mathrm{u1}(t)\,\theta\,c1}{\theta^2\,V0\,(c1-c2)^2} \end{bmatrix}$$

We note that this transformation is well-defined if and only if $c1 \neq c2$.

Let us check that the system matrix for the variables $z1$, $v1$, $z2$, $v2$ is in Brunovský canonical form. To this end we solve

$$Br \begin{pmatrix} x1 \\ x2 \\ u1 \\ u2 \end{pmatrix} = \begin{pmatrix} z1 \\ v1 \\ z2 \\ v2 \end{pmatrix}$$

for $(x1 \quad x2 \quad u1 \quad u2)^T$ modulo the equations $R\,(x1 \quad x2 \quad u1 \quad u2)^T = 0$, which allows to eliminate the latter variables. The command `Elimination` computes a Janet basis with respect to a monomial ordering as described in Remark 2.2.3 (c) to find relations among the variables $z1$, $v1$, $z2$, $v2$ as just explained.

```
> F := Elimination(linalg[stackmatrix](Br, R), [x1,x2,u1,u2],
> [z1,v1,z2,v2,0,0], Alg):

> ApplyMatrix(F[1], [x1(t),x2(t),u1(t),u2(t)], Alg)=
> ApplyMatrix(F[2], [z1(t),v1(t),z2(t),v2(t)], Alg);
```

$$\begin{bmatrix} 0 \\ 0 \\ \mathrm{u2}(t) \\ \mathrm{u1}(t) \\ \mathrm{x2}(t) \\ \mathrm{x1}(t) \end{bmatrix} =$$

$$\left[-(\tfrac{d}{dt}\,\mathrm{z2}(t)) + \mathrm{v2}(t)\right]$$
$$\left[-(\tfrac{d}{dt}\,\mathrm{z1}(t)) + \mathrm{v1}(t)\right]$$
$$[-\mathrm{z1}(t)\ \mathit{V0}\ \mathit{c1} - \mathrm{z1}(t)\ \mathit{V0}\ \mathit{c0} + 2\,\mathrm{z1}(t)\ \mathit{V0}\ \mathit{c2} + 2\,\mathrm{v1}(t)\,\theta\ \mathit{V0}\ \mathit{c2} - 2\,\mathrm{v1}(t)\,\theta\ \mathit{V0}\ \mathit{c1}$$
$$- \mathrm{z2}(t)\ \mathit{V0}\ \mathit{c1} + \mathrm{z2}(t)\ \mathit{V0}\ \mathit{c0}]$$
$$[-\mathrm{z1}(t)\ \mathit{V0}\ \mathit{c2} + \mathrm{z1}(t)\ \mathit{V0}\ \mathit{c0} - \mathrm{z2}(t)\ \mathit{V0}\ \mathit{c2} - \mathrm{z2}(t)\ \mathit{V0}\ \mathit{c0} + 2\,\mathrm{z2}(t)\ \mathit{V0}\ \mathit{c1}$$
$$- 2\,\mathrm{v2}(t)\,\theta\ \mathit{V0}\ \mathit{c2} + 2\,\mathrm{v2}(t)\,\theta\ \mathit{V0}\ \mathit{c1}]$$
$$[2\,\theta\,\mathrm{z1}(t)\ \mathit{c2}^2 - 2\,\theta\,\mathrm{z1}(t)\ \mathit{c2}\ \mathit{c1} - 2\,\theta\,\mathrm{z1}(t)\ \mathit{c2}\ \mathit{c0} + 2\,\theta\,\mathrm{z1}(t)\ \mathit{c1}\ \mathit{c0} + 2\,\theta\,\mathrm{z2}(t)\ \mathit{c2}\ \mathit{c0}$$
$$- 2\,\theta\,\mathrm{z2}(t)\ \mathit{c2}\ \mathit{c1} - 2\,\theta\,\mathrm{z2}(t)\ \mathit{c1}\ \mathit{c0} + 2\,\theta\,\mathrm{z2}(t)\ \mathit{c1}^2]$$
$$[2\,\mathrm{z1}(t)\,\theta\ \mathit{V0}\ \mathit{c2} - 2\,\mathrm{z1}(t)\,\theta\ \mathit{V0}\ \mathit{c1} - 2\,\mathrm{z2}(t)\,\theta\ \mathit{V0}\ \mathit{c2} + 2\,\mathrm{z2}(t)\,\theta\ \mathit{V0}\ \mathit{c1}]$$

The first two equations show that we have transformed the system matrix into Brunovský canonical form, i.e. when considering $(\mathit{z1} \quad \mathit{z2})^T$ as state and $(\mathit{v1} \quad \mathit{v2})^T$ as input of the system, then the state is determined from the input simply by integration. Moreover, the remaining equations in the previous result express $\mathit{x1}$, $\mathit{x2}$, $\mathit{u1}$, and $\mathit{u2}$ in terms of $\mathit{z1}$, $\mathit{v1}$, $\mathit{z2}$, $\mathit{v2}$.

7.3 Autonomous observables

The concentrations $\mathit{c0}$, $\mathit{c1}$, and $\mathit{c2}$ satisfy a linear equation resulting from the linearization of the nonlinear model around the steady-state.

By taking another relation for the values of the steady-state into account, one derives that $\mathit{c1} = \mathit{c2}$ implies $\mathit{c0} = \mathit{c1} = \mathit{c2}$ (for more details see [KS72]). We are going to study the case $\mathit{c0} = \mathit{c1} = \mathit{c2}$ separately. The matrix which represents the system equations is in this case:

```
>  R1mod := subs([c2=c0,c1=c0], evalm(R));
```

$$\mathit{R1mod} := \begin{bmatrix} \mathrm{D} + \dfrac{1}{2\,\theta} & 0 & -1 & -1 \\ 0 & \mathrm{D} + \dfrac{1}{\theta} & 0 & 0 \end{bmatrix}$$

Let us denote by M' the $\mathcal{O}$-module which is associated with this particular system. The matrix which presents the transposed module of M' is $\mathit{R1mod}^T$ because $\mathcal{O}$ is a commutative ring (see Def. 5.1.4).

```
> R1mod_adj := linalg[transpose](R1mod);
```

$$R1mod\_adj := \begin{bmatrix} \mathrm{D} + \frac{1}{2\,\theta} & 0 \\ 0 & \mathrm{D} + \frac{1}{\theta} \\ -1 & 0 \\ -1 & 0 \end{bmatrix}$$

In order to check controllability and parametrizability, we compute the extension group $\mathrm{ext}^1_{\mathcal{O}}(M'^{\top}, \mathcal{O})$ of the transposed module of M'.

```
> Ext1mod := Exti(R1mod_adj, Alg, 1);
```

$$Ext1mod := \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 + \mathrm{D}\,\theta \end{bmatrix}, \begin{bmatrix} 2\,\mathrm{D}\,\theta + 1 & 0 & -2\,\theta & -2\,\theta \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} -2\,\theta & 0 \\ 0 & 0 \\ 0 & 1 \\ -2\,\mathrm{D}\,\theta - 1 & -1 \end{bmatrix} \right]$$

The residue classes in M' represented by the rows of the second matrix in *Ext1mod* generate $t(M')$, and the residue class represented by the i-th row is annihilated by the entries in the i-th column of the first matrix in *Ext1mod*. Hence, we find a non-trivial torsion element m in M', namely the residue class represented by *x2*. It satisfies the equation $(\theta\,\mathrm{D} + 1)\,m = 0$.

Therefore, the system is not (completely) controllable.

Note that *Ext1mod*[2] in $\mathcal{O}^{2\times 4}$ can also be interpreted as a matrix which presents $M'/t(M')$, i.e. we have $M'/t(M') = \mathrm{coker}(.Ext1mod[2])$. Up to isomorphism, $M'/t(M')$ is the $\mathcal{O}$-module which is associated with the controllable part of the system. Accordingly, *Ext1mod*[3] is a parametrization of $M'/t(M')$.

We can find directly a generating set of autonomous observables of the system by using the command `AutonomousElements`:

```
> AutonomousElements(R1mod, [x1(t),x2(t),u1(t),u2(t)], Alg);
```

$$[[\ \theta_1(t) + \theta\,(\tfrac{d}{dt}\,\theta_1(t)) = 0\ ],\ [\ \theta_1 = \_C1\ e^{(-\frac{t}{\theta})}\ ],\ [\ \theta_1 = \mathrm{x2}(t)\ ]]$$

The first entry of the result is an autonomous equation for the autonomous observable θ_1 which is defined in terms of the system variables by the third entry. By solving the autonomous equation, θ_1 is obtained as a function of t which is given by the second entry.

It was proved in [PQ99b] that the autonomous observables of a linear system are in one-to-one correspondence with the first integrals of motion of the system. The autonomous observable given above corresponds to the following first integral of motion:

```
> V := FirstIntegral(R1mod, [x1(t),x2(t),u1(t),u2(t)], Alg);
```

$$V := \_C1\, e^{(\frac{t}{\theta})}\, \mathrm{x2}(t)$$

By definition of the first integral of motion, the time derivative of V is zero modulo the system equations.

```
> Vdot := diff(V, t);
```

$$Vdot := \frac{\_C1\, e^{(\frac{t}{\theta})}\, \mathrm{x2}(t)}{\theta} + \_C1\, e^{(\frac{t}{\theta})}\, (\tfrac{d}{dt}\, \mathrm{x2}(t))$$

We recall the left hand sides of the system equations:

```
> Sys := ApplyMatrix(R1mod, [x1(t),x2(t),u1(t),u2(t)], Alg);
```

$$Sys := \left[\begin{array}{c} \frac{1}{2} \frac{\mathrm{x1}(t) + 2\, (\frac{d}{dt}\, \mathrm{x1}(t))\, \theta - 2\, \mathrm{u1}(t)\, \theta - 2\, \mathrm{u2}(t)\, \theta}{\theta} \\ \frac{\mathrm{x2}(t) + (\frac{d}{dt}\, \mathrm{x2}(t))\, \theta}{\theta} \end{array} \right]$$

It is possible to solve the second equation for the first derivative of *x2*:

```
> sol := solve(Sys[2,1], diff(x2(t), t));
```

$$sol := -\frac{\mathrm{x2}(t)}{\theta}$$

By substituting *sol* for the first derivative of *x2* in $Vdot$, we can verify that the time derivative of V is zero:

```
> subs(diff(x2(t), t)=sol, Vdot);
```

$$0$$

Now we pass over to the controllable part of the system:

```
> ApplyMatrix(Ext1mod[2], [x1(t),x2(t),u1(t),u2(t)], Alg)=
> matrix([[0],[0]]);
```

$$\left[\begin{array}{c} \mathrm{x1}(t) + 2\, (\frac{d}{dt}\, \mathrm{x1}(t))\, \theta - 2\, \mathrm{u1}(t)\, \theta - 2\, \mathrm{u2}(t)\, \theta \\ \mathrm{x2}(t) \end{array} \right] = \left[\begin{array}{c} 0 \\ 0 \end{array} \right]$$

As $\mathcal{O}$ is a principal ideal domain, the torsion-free module $M'/t(M')$ is free. A flat output of the behavior of the controllable part is obtained as follows:

```
> S2 := LeftInverse(Ext1mod[3], Alg);
```

$$S2 := \begin{bmatrix} -\dfrac{1}{2\,\theta} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Hence, a flat output is defined by $\xi = (\xi_1 \quad \xi_2)^T = S2\,(x1 \quad x2 \quad u1 \quad u2)^T$. We have $(x1 \quad x2 \quad u1 \quad u2)^T = Ext1mod[3]\,(\xi_1 \quad \xi_2)^T$.

Although the stirred tank has autonomous observables for the given configuration of the parameters, we can construct a parametrization of its behavior along the lines of Section 6.3. Due to Remark 6.3.5, the torsion submodule $t(M')$ has a complement in M'. This is taken into account by the command `Parametrization` which, using the notation of Theorem 6.3.8, computes a parametrization $P1mod := P\,\xi + S\,\tau$ of the solution set of $R1mod\,\eta = 0$, where ξ is a vector of arbitrary (smooth) functions of t and the vector τ depends on certain functions and constants which can be chosen arbitrarily.

```
> P1mod := Parametrization(R1mod, Alg);
```

$$P1mod := \begin{bmatrix} -2\,\theta\,\xi_1(t) \\ \_C1\,e^{(-\frac{t}{\theta})} \\ \xi_2(t) \\ -\xi_1(t) - 2\,\theta\,(\frac{d}{dt}\,\xi_1(t)) - \xi_2(t) \end{bmatrix}$$

We can easily verify that $P1mod$ parametrizes some solutions of the system:

```
> ApplyMatrix(R1mod, P1mod, Alg);
```

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Since we deal with a linear system with constant coefficients, we can choose the space of smooth functions on $\mathbb{R}$ as an injective $\mathcal{O}$-module $\mathcal{F}$ which is also a cogenerator for ${}_{\mathcal{O}}M$ (see Ex. 4.4.5 (a) (3)). Therefore we actually have

$$\mathrm{Sol}_{\mathcal{F}}(R) = \{P1mod \mid (\xi_1, \xi_2)^T \in \mathcal{F}^{2\times 1},\ \_C1 \in \mathbb{R}\}.$$

Finally we demonstrate that our controllability result agrees with the conclusion drawn from the standard check using the controllability matrix for a Kalman system. The system equations can be written as $\dot{x} = A\,x + B\,u$, where A and B are defined as follows:

```
> A := evalm([[-1/(2*theta),0],[0,-1/theta]]);
```

$$A := \begin{bmatrix} -\dfrac{1}{2\,\theta} & 0 \\ 0 & -\dfrac{1}{\theta} \end{bmatrix}$$

```
> B := matrix([[1,1],[(c1-c0)/V0,(c2-c0)/V0]]);
```

$$B := \begin{bmatrix} 1 & 1 \\ \dfrac{c1 - c0}{V0} & \dfrac{c2 - c0}{V0} \end{bmatrix}$$

Let us compute the controllability matrix:

```
> C := ControllabilityMatrix(A, B, 2, Alg);
```

$$C := \begin{bmatrix} 1 & 1 & -\dfrac{1}{2\,\theta} & -\dfrac{1}{2\,\theta} \\ \dfrac{c1 - c0}{V0} & \dfrac{c2 - c0}{V0} & \dfrac{c0 - c1}{\theta\, V0} & \dfrac{c0 - c2}{\theta\, V0} \end{bmatrix}$$

```
> linalg[rank](C);
```

$$2$$

Generically the controllability matrix C has full rank. Hence, the stirred tank is controllable for generic configurations of the parameters. We compute all principal minors of C:

```
> col := combinat[choose]([1,2,3,4], 2);
```

$$col := [[1,\, 2],\, [1,\, 3],\, [1,\, 4],\, [2,\, 3],\, [2,\, 4],\, [3,\, 4]]$$

```
> d := map(c->numer(linalg[det](linalg[submatrix](C,
> 1..2, c))), col);
```

$$d := [c2 - c1,\, c0 - c1,\, -2\, c2 + c0 + c1,\, c2 + c0 - 2\, c1,\, c0 - c2,\, c2 - c1]$$

```
> solve({op(d)});
```

$$\{c1 = c2,\, c0 = c2,\, c2 = c2\}$$

The controllability matrix has rank 1 if and only if $c0 = c1 = c2$. According to the remark at the beginning of this section, the system is controllable if and only if $c1 \neq c2$.

7.4 Observability, input-output behavior

Let us introduce variables which we consider as outputs for the system: $y1 = x1/(2\,\theta)$ and $y2 = x2$ (see [KS72], p. 9). We are going to check whether the system is observable and study its input-output behavior. First of all we define the following matrix $Y \in \mathcal{O}^{2\times 4}$ which expresses the output variables in terms of the system variables $x1$, $x2$, $u1$, $u2$:

```
> Y := matrix([[1/(2*theta),0,0,0],[0,1,0,0]]);
```

$$Y := \begin{bmatrix} \frac{1}{2\,\theta} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Therefore, we consider the system described by

$$\begin{pmatrix} x1 \\ x2 \\ y1 \\ y2 \end{pmatrix} = \begin{pmatrix} R \\ Y \end{pmatrix} \begin{pmatrix} x1 \\ x2 \\ u1 \\ u2 \end{pmatrix}. \tag{7.3}$$

```
> RY := linalg[stackmatrix](R, Y);
```

$$RY := \begin{bmatrix} \mathrm{D} + \frac{1}{2\,\theta} & 0 & -1 & -1 \\ 0 & \mathrm{D} + \frac{1}{\theta} & -\frac{c1 - c0}{V0} & -\frac{c2 - c0}{V0} \\ \frac{1}{2\,\theta} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

In these equations we try to eliminate *x1* and *x2*:

```
> E := Elimination(RY,[x1,x2,u1,u2],[0,0,y1,y2],Alg,[u1,u2]):
> ApplyMatrix(E[1], [x1(t),x2(t)], Alg)=
> ApplyMatrix(E[2], [y1(t),y2(t),u1(t),u2(t)], Alg);
```

$$\begin{bmatrix} 0 \\ 0 \\ \mathrm{x2}(t) \\ \mathrm{x1}(t) \end{bmatrix} = \begin{bmatrix} -V0\,\mathrm{y2}(t) - \theta\,V0\,(\frac{d}{dt}\,\mathrm{y2}(t)) + \mathrm{u1}(t)\,\theta\,c1 - \mathrm{u1}(t)\,\theta\,c0 + \mathrm{u2}(t)\,\theta\,c2 - \mathrm{u2}(t)\,\theta\,c0 \\ -\mathrm{y1}(t) - 2\,\theta\,(\frac{d}{dt}\,\mathrm{y1}(t)) + \mathrm{u1}(t) + \mathrm{u2}(t) \\ \mathrm{y2}(t) \\ 2\,\theta\,\mathrm{y1}(t) \end{bmatrix}$$

The result shows that the system variables *x1* and *x2* can be expressed without derivatives in terms of the system variables *y1*, *y2*. Hence, the state $(x1 \quad x2)^T$ of the system can be computed from the output $(y1 \quad y2)^T$, which means that the stirred tank with output $(y1 \quad y2)^T$ is observable. Moreover, the first two of the previous equations describe the input-output behavior of the system, i.e. we have

$$\begin{cases} V0\;y2 + \theta\;V0\,\frac{d}{dt}y2 &= (\theta\;c1 - \theta\;c0)\;u1 + (\theta\;c2 - \theta\;c0)\;u2, \\ y1 + 2\,\theta\,\frac{d}{dt}y1 &= u1 + u2. \end{cases}$$

We demonstrate another possibility to prove the observability of the system. It is easily seen that the variables *x1*, *x2* on the one hand and the variables *u1*, *u2*,

$y1$, $y2$ on the other hand can be separated in the equation (7.3) in such a way that (7.3) is equivalent to

$$RX \begin{pmatrix} x1 \\ x2 \end{pmatrix} = RUY \begin{pmatrix} u1 \\ u2 \\ y1 \\ y2 \end{pmatrix} \tag{7.4}$$

with appropriate matrices $RX \in \mathcal{O}^{4\times 2}$, $RUY \in \mathcal{O}^{4\times 4}$. These matrices are defined as follows:

```
> RX := linalg[submatrix](RY, 1..4, 1..2);
```

$$RX := \begin{bmatrix} \mathrm{D} + \frac{1}{2\,\theta} & 0 \\ 0 & \mathrm{D} + \frac{1}{\theta} \\ \frac{1}{2\,\theta} & 0 \\ 0 & 1 \end{bmatrix}$$

```
> RUY := linalg[diag](linalg[submatrix](-RY, 1..2, 3..4),
> linalg[diag](1,1));
```

$$RUY := \begin{bmatrix} \frac{1}{\frac{c1-c0}{V0}} & \frac{1}{\frac{c2-c0}{V0}} & 0 & 0 \\ & & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In order to solve (7.4) for $(x1 \quad x2)^T$, we compute a matrix $S \in \mathcal{O}^{2\times 4}$ which satisfies $S \cdot RX = I_2$.

```
> S := LeftInverse(RX, Alg);
```

$$S := \begin{bmatrix} 0 & 0 & 2\,\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplying S on both sides of (7.4) from the left reveals that the system is observable because the state variables $x1$, $x2$ are expressed in terms of the output variables $y1$, $y2$:

```
> matrix([[x1(t)],[x2(t)]])=ApplyMatrix(Mult(S, RUY, Alg),
> [u1(t),u2(t),y1(t),y2(t)], Alg);
```

$$\begin{bmatrix} \mathrm{x1}(t) \\ \mathrm{x2}(t) \end{bmatrix} = \begin{bmatrix} 2\,\theta\,\mathrm{y1}(t) \\ \mathrm{y2}(t) \end{bmatrix}$$

7.5 Motion planning

Let us recall that in case $c1 \neq c2$ the system is flat, i.e. its behavior has an injective parametrization. In Section 7.2 we have computed matrices $P \in \mathcal{O}^{4\times 2}$ and $S \in \mathcal{O}^{2\times 4}$ such that

$$\begin{pmatrix} x1 \\ x2 \\ u1 \\ u2 \end{pmatrix} = P \begin{pmatrix} \xi 1 \\ \xi 2 \end{pmatrix}, \quad \begin{pmatrix} \xi 1 \\ \xi 2 \end{pmatrix} = S \begin{pmatrix} x1 \\ x2 \\ u1 \\ u2 \end{pmatrix}.$$

Then we defined $Flat := P\,S \in \mathcal{O}^{4\times 4}$ and noticed that we can choose $(x1 \quad x2)^T$ as a flat output of the behavior of the stirred tank. The input variables $u1$, $u2$ are expressed in terms of the flat output $(x1 \quad x2)^T$ as $(u1 \quad u2)^T = F\,(x1 \quad x2)^T$, where F is defined as follows:

```
> F := linalg[submatrix](Flat, 3..4, 1..2);
```

$$F := \begin{bmatrix} \dfrac{(2\,\mathrm{D}\,\theta + 1)\,(c0 - c2)}{2\,\theta\,(c1 - c2)} & \dfrac{(1 + \mathrm{D}\,\theta)\;V0}{(c1 - c2)\,\theta} \\ -\dfrac{(2\,\mathrm{D}\,\theta + 1)\,(c0 - c1)}{2\,\theta\,(c1 - c2)} & -\dfrac{(1 + \mathrm{D}\,\theta)\;V0}{(c1 - c2)\,\theta} \end{bmatrix}$$

For given reference trajectories $x1_{\mathrm{ref}}$, $x2_{\mathrm{ref}}$ for $x1$ resp. $x2$ the open-loop controls $u1_{\mathrm{ref}}$, $u2_{\mathrm{ref}}$ which lead to the desired trajectories for $x1$ and $x2$ are obtained as

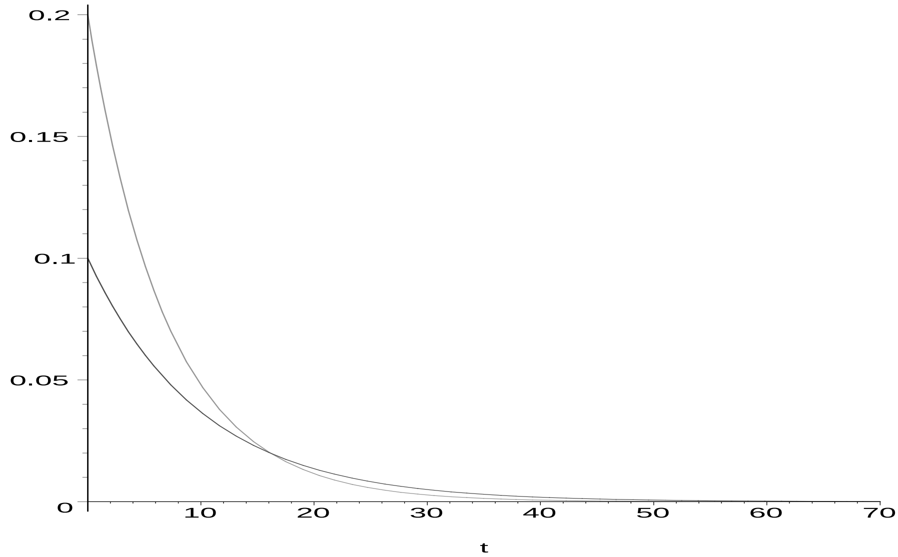
$$\begin{pmatrix} u1_{\mathrm{ref}} \\ u2_{\mathrm{ref}} \end{pmatrix} = F \begin{pmatrix} x1_{\mathrm{ref}} \\ x2_{\mathrm{ref}} \end{pmatrix}.$$

In this section we use the parameter values given on p. 10 of [KS72]:

```
> conf := [c1=1,c2=2,c0=1.25,V0=1,theta=50]:
```

Let us consider the following reference trajectories:

```
> x1ref := 0.1*exp(-alpha*t):  x2ref := 0.2*exp(-beta*t):
> alpha := 1/10:  beta := 1/7:
> plot([x1ref,x2ref], t=0..70, color=[red,green]);
```



The corresponding open-loop controls are then given by:

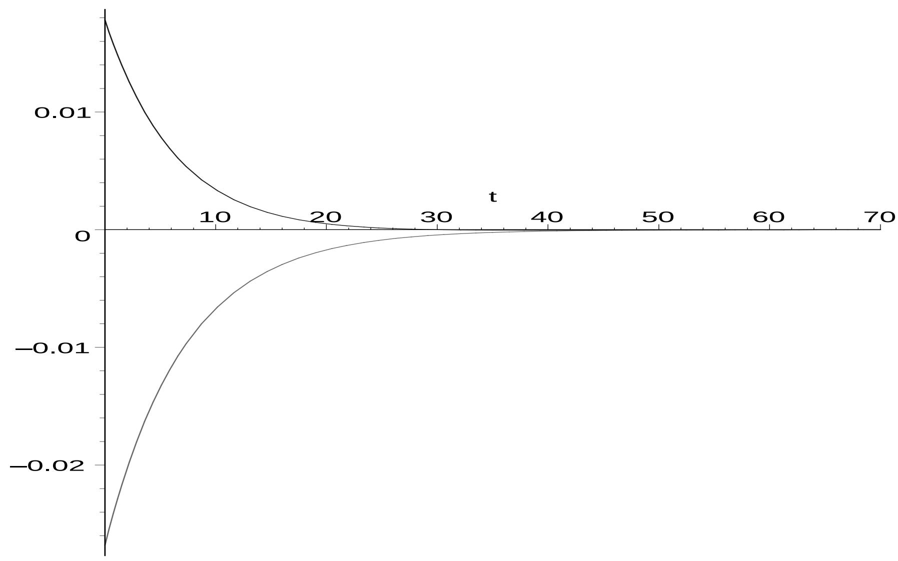
```
> u1ref := subs(conf, ApplyMatrix(F, [x1ref,x2ref], Alg)[1,1]);
```

$$u1ref := -0.006750000000\, e^{(-0.1000000000\, t)} + 0.02457142856\, e^{(-0.1428571429\, t)}$$

```
> u2ref := subs(conf, ApplyMatrix(F, [x1ref,x2ref], Alg)[2,1]);
```

$$u2ref := -0.002250000000\, e^{(-0.1000000000\, t)} - 0.02457142856\, e^{(-0.1428571429\, t)}$$

```
> plot([u1ref,u2ref], t=0..70, color=[blue,magenta]);
```



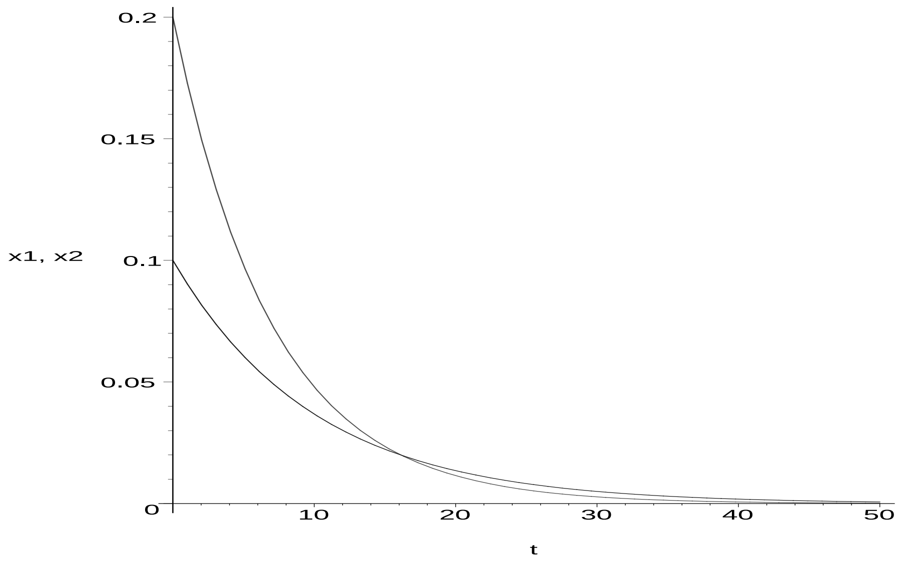
We are going to simulate the system for the given input $(u1_{\mathrm{ref}} \quad u2_{\mathrm{ref}})^T$.

```
> ODE1 := {seq(ApplyMatrix(subs(conf, evalm(R)),
> [x1(t),x2(t),u1ref,u2ref], Alg)[i,1]=0,i=1..2)};
```

$$ODE1 := \{\frac{1}{100}\,\mathrm{x1}(t) + (\tfrac{d}{dt}\,\mathrm{x1}(t)) + 0.009000000000\,e^{(-0.1000000000\,t)} = 0,$$
$$\frac{1}{50}\,\mathrm{x2}(t) + (\tfrac{d}{dt}\,\mathrm{x2}(t)) + 0.02457142856\,e^{(-0.1428571429\,t)} = 0\}$$

The initial conditions are:

```
> IC := {x1(0)=0.1, x2(0)=0.2}:
> dsol1 := dsolve(ODE1 union IC, {x1(t),x2(t)}, type=numeric,
> stiff=true):
> plots[odeplot](dsol1, [[t,x1(t), color=blue],
> [t,x2(t),color=red]], 0..50);
```



Due to model errors and noises, the system needs to be stabilized around the trajectory $(x1_{\mathrm{ref}} \quad x2_{\mathrm{ref}} \quad u1_{\mathrm{ref}} \quad u2_{\mathrm{ref}})^T$.

In fact, we have $\dot{x} = A\,x + B\,u$, where A and B are defined as in Section 7.3. By linearity the error $e = x - x_{\mathrm{ref}}$ satisfies $\dot{e} = A\,e + B\,(u - u_{\mathrm{ref}})$. If we choose the feedback $u = u_{\mathrm{ref}} + K\,e$, then the dynamics of the error are described by

$$\dot{e} = (A + B\,K)\,e.$$

Since the system is controllable, K can be chosen in such a way that the error e tends to zero (pole placement). The closed-loop system defined by the above feedback is then described by

$$\dot{x} = A\,x + B\,(u_{\mathrm{ref}} + K\,e) = (A + B\,K)\,x + B\,u_{\mathrm{ref}} - B\,K\,x_{\mathrm{ref}}. \tag{7.5}$$

Let us recall the matrices A and B from Section 7.3 and choose the feedback gain K as follows:

```
> A := evalm([[-1/(2*theta),0],[0,-1/theta]]):
> B := matrix([[1,1],[(c1-c0)/V0,(c2-c0)/V0]]):
> K := matrix([[-0.7425,7.92],[-0.2475,0]]);
```

$$K := \begin{bmatrix} -0.7425 & 7.92 \\ -0.2475 & 0 \end{bmatrix}$$

Then we define $L := A + BK$.

```
> L := evalm(subs(conf, evalm(A + B &* K)));
```

$$L := \begin{bmatrix} -1.000000000 & 7.92 \\ 0. & -2.000000000 \end{bmatrix}$$

The characteristic polynomial of L is:

```
> chi := collect(linalg[charpoly](L, lambda), lambda);
```

$$\chi := 2.000000000 + \lambda^2 + 3.000000000\,\lambda$$

```
> factor(chi);
```

$$(\lambda + 1.000000000)\,(\lambda + 2.000000000)$$

Hence, the eigenvalues of L are -1 and -2. Since the (real parts of the) eigenvalues are negative, the closed-loop system is asymptotically stable.

The equations (7.5) are entered as follows:

```
> lhseq := subs(conf, evalm(ApplyMatrix(evalm(D-L),
> [x1(t),x2(t)], Alg)-ApplyMatrix(B, [u1ref,u2ref], Alg)+
> ApplyMatrix(Mult(B, K, Alg), [x1ref,x2ref], Alg)));
```

$$lhseq :=$$

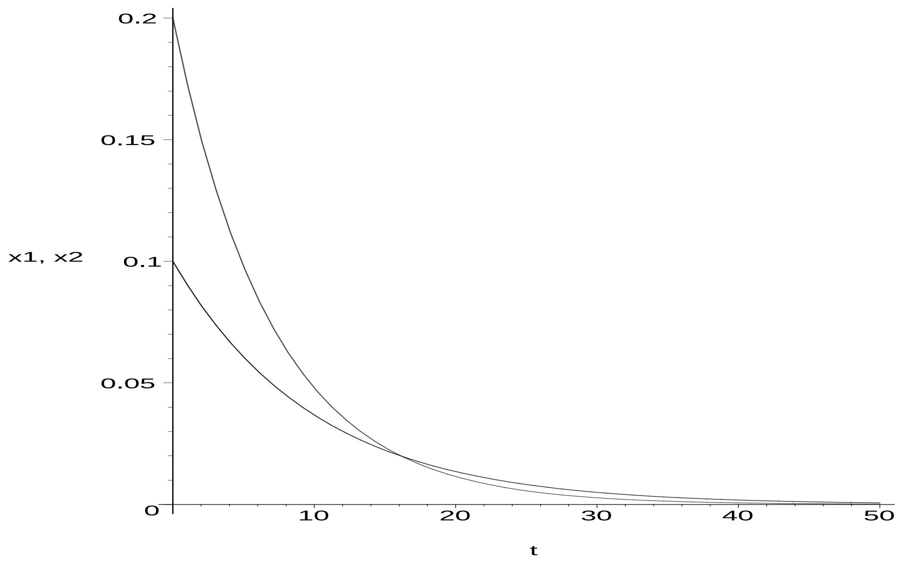
$$\left[1.000000000\,\mathrm{x1}(t) + (\tfrac{d}{dt}\,\mathrm{x1}(t)) - 7.92\,\mathrm{x2}(t) - 0.09000000000\,e^{(-0.1000000000\,t)}\right.$$
$$\left. + 1.584000000\,e^{(-0.1428571429\,t)}\right]$$
$$\left[2.000000000\,\mathrm{x2}(t) + (\tfrac{d}{dt}\,\mathrm{x2}(t)) - 0.3714285714\,e^{(-0.1428571429\,t)}\right]$$

```
> ODE2 := convert(lhseq, set);
```

$$ODE2 := \{1.000000000\,\mathrm{x1}(t) + (\tfrac{d}{dt}\,\mathrm{x1}(t)) - 7.92\,\mathrm{x2}(t)$$
$$- 0.09000000000\,e^{(-0.1000000000\,t)} + 1.584000000\,e^{(-0.1428571429\,t)},$$
$$2.000000000\,\mathrm{x2}(t) + (\tfrac{d}{dt}\,\mathrm{x2}(t)) - 0.3714285714\,e^{(-0.1428571429\,t)}\}$$

```
> dsol2 := dsolve(ODE2 union IC, {x1(t),x2(t)}, type=numeric,
> stiff=true):
```

```
> plots[odeplot](dsol2, [[t,x1(t), color=blue],
> [t,x2(t),color=red]], 0..50);
```



Let us consider the case where there are some perturbations in the system represented by terms $\exp(-10\,t)\,\sin(10\,t)^2$ on the right hand side of the equations:

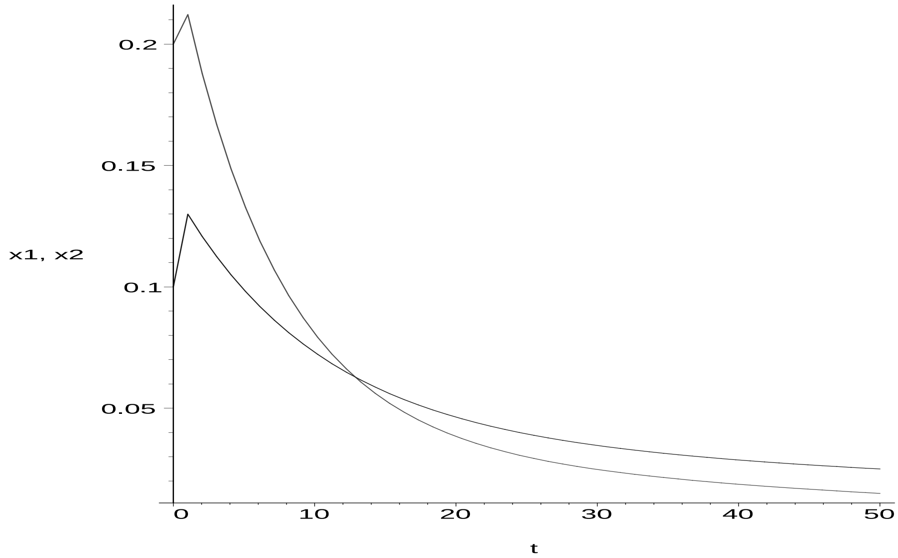
```
> ODE3 := {seq(ApplyMatrix(subs(conf, evalm(R)),
> [x1(t),x2(t),u1ref,u2ref],
> Alg)[i,1]=exp(-10*t)*sin(10*t)^2,i=1..2)};
```

$$ODE3 := \{\frac{1}{50}\,\mathrm{x2}(t) + (\tfrac{d}{dt}\,\mathrm{x2}(t)) + 0.02457142856\,e^{(-0.1428571429\,t)} = e^{(-10\,t)}\sin(10\,t)^2,$$
$$\frac{1}{100}\,\mathrm{x1}(t) + (\tfrac{d}{dt}\,\mathrm{x1}(t)) + 0.009000000000\,e^{(-0.1000000000\,t)} = e^{(-10\,t)}\sin(10\,t)^2\}$$

```
> dsol3 := dsolve(ODE3 union IC, {x1(t),x2(t)}, type=numeric,
> stiff=true):
```

For the open-loop system the trajectories of *x1* and *x2* are the following:

```
> plots[odeplot](dsol3, [[t,x1(t),color=blue],
> [t,x2(t),color=red]], 0..50);
```

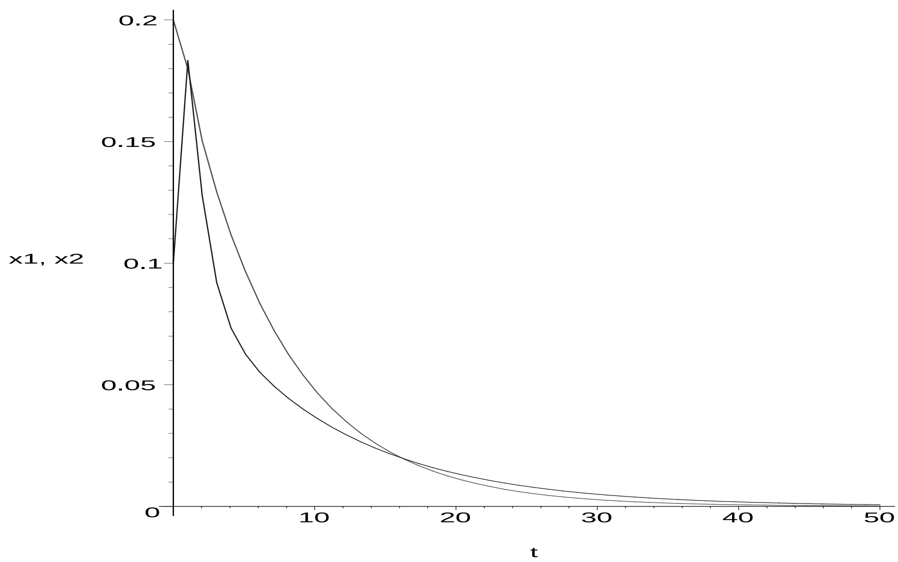


We determine the trajectories of *x1* and *x2* for the closed-loop system:

```
> ODE4 := {lhseq[1,1]=exp(-10*t)*sin(10*t)^2,
> lhseq[2,1]=exp(-10*t)*sin(10*t)^2};
```

$$ODE4 := \{$$
$$2.000000000\,\mathrm{x2}(t) + (\tfrac{d}{dt}\,\mathrm{x2}(t)) - 0.3714285714\,e^{(-0.1428571429\,t)} = e^{(-10\,t)}\sin(10\,t)^2,$$
$$1.000000000\,\mathrm{x1}(t) + (\tfrac{d}{dt}\,\mathrm{x1}(t)) - 7.92\,\mathrm{x2}(t) - 0.09000000000\,e^{(-0.1000000000\,t)}$$
$$+\,1.584000000\,e^{(-0.1428571429\,t)} = e^{(-10\,t)}\sin(10\,t)^2\}$$

```
> dsol4 := dsolve(ODE4 union IC, {x1(t),x2(t)}, type=numeric,
> stiff=true):
> plots[odeplot](dsol4, [[t,x1(t), color=blue],
> [t,x2(t),color=red]], 0..50);
```



From the plots we can see that now the trajectories of $x1$ and $x2$ are again closer to the ones in the unperturbed situation. In particular, the state $(x1 \quad x2)^T$ tends to zero more rapidly than without feedback.

7.6 Optimal control problems

Most of the time the demands on the dynamics of a system cannot be fulfilled simultaneously. For instance, the wish that the state variables tend to zero very rapidly can be met when the system is controllable by assigning values with very small negative real parts to the spectrum of the system matrix describing the closed-loop system. However, great influences on the system dynamics require a great amount of energy, which is always restricted in practice. This leads to a formulation of an optimal control problem [Son98, KS72]: the control for the system is determined in such a way that it minimizes a given cost functional which is designed according to the demands on the dynamics. Of course, the original equations which describe the system still have to be satisfied by the system variables. The simplest non-trivial situation is given for a linear system of ordinary differential equations and a cost functional of the form

$$\int_{t_0}^{t_1} \left(x^T\, C1\, x + u^T\, C2\, u\right)\, dt, \tag{7.6}$$

where $C1$ and $C2$ are positive definite symmetric matrices. Such problems are referred to as linear quadratic optimal control problems. The first term of the functional measures the deviation of the state from zero during the time interval $[t_0, t_1]$. Similarly, the second summand reflects the size of the input values for $t \in [t_0, t_1]$. Other variants of the cost functional include a term $x(t_1)^T\, C3\, x(t_1)$ (without integration) which measures the distance of the state to zero at the final time instant t_1 with a positive definite symmetric matrix $C3$. Of course, the demands on the system dynamics can be encoded by appropriate weighting of these terms, i.e. by the choice of the matrices $C1$, $C2$, and $C3$.

In this section we consider a linear quadratic optimal control problem for the stirred tank. We first deal with a cost functional of the form (7.6) which we define along the lines of [KS72], p. 227.

First of all, we define the matrix

```
> Q := matrix([[sigma1,0],[0,sigma2]]);
```

$$Q := \begin{bmatrix} \sigma 1 & 0 \\ 0 & \sigma 2 \end{bmatrix}$$

(with positive real numbers $\sigma 1$, $\sigma 2$) which is supposed to measure the output y defined in Section 7.4 by means of a term $y^T\, Q\, y$. The output variables

$y1$, $y2$ are related to the state variables $x1$, $x2$ and the input variables $u1$, $u2$ via $y = Y\,(x^T \quad u^T)^T$, so that the definition of Q implies the summand $(x^T \quad u^T)\,C1\,(x^T \quad u^T)^T$ for the cost functional, where

```
> C1 := evalm(linalg[transpose](Y) &* Q &* Y);
```

$$C1 := \begin{bmatrix} \frac{\sigma 1}{4\,\theta^2} & 0 & 0 & 0 \\ 0 & \sigma 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Following [KS72] we define a summand in (7.6) which measures the size of the input as follows (ρ, $\rho 1$, $\rho 2$ are positive real numbers):

```
> C2 := linalg[diag](0,0,rho*rho1,rho*rho2);
```

$$C2 := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \rho\,\rho 1 & 0 \\ 0 & 0 & 0 & \rho\,\rho 2 \end{bmatrix}$$

```
> C := evalm(C1 + C2);
```

$$C := \begin{bmatrix} \frac{\sigma 1}{4\,\theta^2} & 0 & 0 & 0 \\ 0 & \sigma 2 & 0 & 0 \\ 0 & 0 & \rho\,\rho 1 & 0 \\ 0 & 0 & 0 & \rho\,\rho 2 \end{bmatrix}$$

Now we consider the problem of minimizing the cost functional

$$\frac{1}{2}\int_0^T (x1 \quad x2 \quad u1 \quad u2)\,C \begin{pmatrix} x1 \\ x2 \\ u1 \\ u2 \end{pmatrix} dt$$

subject to the system equations $R\,(x1 \quad x2 \quad u1 \quad u2)^T = 0$ with given initial conditions $x1(0) = x10$, $x2(0) = x20$.

We shall see later that we can also add a term which measures the terminal state $x(T)$.

Using the fact that the system is parametrizable for generic configurations of the parameters, we can substitute the parametrization P defined in Section 7.2 into the cost functional (and possibly into the term measuring the terminal state) in order to obtain a variational problem without (differential) constraint on the arguments ξ_1, ξ_2 of the parametrization:

$$\min \frac{1}{2} \int_0^T (\xi_1 \quad \xi_2)\, P^T\, C\, P \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}^T dt.$$

Therefore, necessary conditions on optimality can be derived by computing the Euler-Lagrange equations for the new cost functional. For more details on the theory, we refer the reader to [Qua99] as well as [PQ04].

We add the constants $\sigma 1$, $\sigma 2$, ρ, $\rho 1$, and $\rho 2$ in the declaration of the Ore algebra:

```
> Alg := DefineOreAlgebra(diff=[D,t], polynom=[t],
> comm=[theta,V0,c0,c1,c2,sigma1,sigma2,rho,rho1,rho2]):
```

Then, we compute the Euler-Lagrange equations by using the `OreModules` command `LQEquations` (note that it is not necessary to specify the parametrization P because `LQEquations` applies `Parametrization` to R).

```
> LQP := LQEquations(R, C, Alg):
```

The result LQP is a list with three entries (the third of which is the parametrization of the system which we are going to use at the end of this discussion). The first entry of LQP is a matrix that contains the two Euler-Lagrange equations for ξ_1 and ξ_2, namely:

```
> LQP[1][1,1];
```

$$\begin{aligned}
&-2\,\xi_2(t)\; V0^2\,\rho\,\rho 2\; c2\; c1 + 2\,\xi_2(t)\; V0^2\,\rho\,\rho 2\; c2\; c0 + 4\,\xi_1(t)\,\theta^2\,\sigma 2\; c2^4 + 4\,\xi_2(t)\,\theta^2\,\sigma 2\; c2^3\; c0 \\
&- 4\,\xi_2(t)\,\theta^2\,\sigma 2\; c2^3\; c1 - 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 1\; c2\;\theta\; c1 \\
&- 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 1\; c0\;\theta\; c2 + 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 2\,\theta\; c1\; c0 \\
&+ 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 1\; c2^2\,\theta + 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 1\; c0\;\theta\; c1 \\
&+ 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 2\,\theta\; c2\; c1 - 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 2\,\theta\; c2\; c0 \\
&- 2\,(\tfrac{d}{dt}\,\xi_2(t))\; V0^2\,\rho\,\rho 2\,\theta\; c1^2 + 4\,\xi_1(t)\,\theta^2\,\sigma 2\; c1^2\; c0^2 + 4\,\xi_1(t)\,\theta^2\,\sigma 2\; c2^2\; c1^2 \\
&+ \xi_1(t)\; V0^2\,\rho\,\rho 2\; c0^2 + 4\,\xi_1(t)\,\theta^2\,\sigma 2\; c2^2\; c0^2 + 2\,\xi_2(t)\; V0^2\,\sigma 1\; c2\; c1 \\
&- 8\,\xi_1(t)\,\theta^2\,\sigma 2\; c2\; c1^2\; c0 - 8\,\xi_1(t)\,\theta^2\,\sigma 2\; c2\; c0^2\; c1 - 4\,\xi_1(t)\; V0^2\,\rho\,\rho 2\; c2\; c1 \\
&- 4\,\xi_1(t)\; V0^2\,\rho\,\rho 2\; c2\; c0 + 8\,(\tfrac{d^2}{dt^2}\,\xi_1(t))\; V0^2\,\rho\,\rho 2\,\theta^2\; c2\; c1 \\
&- 4\,(\tfrac{d^2}{dt^2}\,\xi_1(t))\; V0^2\,\rho\,\rho 2\,\theta^2\; c1^2 - 4\,(\tfrac{d^2}{dt^2}\,\xi_1(t))\; V0^2\,\rho\,\rho 2\,\theta^2\; c2^2 - 8\,\xi_1(t)\,\theta^2\,\sigma 2\; c2^3\; c0 \\
&- 8\,\xi_1(t)\,\theta^2\,\sigma 2\; c2^3\; c1 + \xi_1(t)\; V0^2\,\rho\,\rho 2\; c1^2 - 2\,\xi_1(t)\; V0^2\,\rho\,\rho 1\; c2\; c0
\end{aligned}$$

$$+\,4\,\xi_1(t)\,V0^2\,\rho\,\rho2\,c2^2 + \xi_1(t)\,V0^2\,\sigma1\,c2^2 + \xi_1(t)\,V0^2\,\sigma1\,c1^2 - \xi_2(t)\,V0^2\,\sigma1\,c2^2$$
$$-\,\xi_2(t)\,V0^2\,\sigma1\,c1^2 - 2\,\xi_1(t)\,V0^2\,\sigma1\,c2\,c1 + 8\,\xi_2(t)\,\theta^2\,\sigma2\,c2^2\,c1^2$$
$$-\,4\,\xi_2(t)\,\theta^2\,\sigma2\,c2\,c1^3 - 4\,\xi_2(t)\,\theta^2\,\sigma2\,c1^2\,c0^2 + 16\,\xi_1(t)\,\theta^2\,\sigma2\,c2^2\,c1\,c0$$
$$+\,\xi_1(t)\,V0^2\,\rho\,\rho1\,c2^2 + \xi_1(t)\,V0^2\,\rho\,\rho1\,c0^2 - 4\,\xi_2(t)\,\theta^2\,\sigma2\,c2^2\,c0^2$$
$$-\,4\,\xi_2(t)\,\theta^2\,\sigma2\,c2^2\,c1\,c0 + 8\,\xi_2(t)\,\theta^2\,\sigma2\,c2\,c0^2\,c1 - 4\,\xi_2(t)\,\theta^2\,\sigma2\,c2\,c1^2\,c0$$
$$+\,4\,\xi_2(t)\,\theta^2\,\sigma2\,c1^3\,c0 - 2\,\xi_2(t)\,V0^2\,\rho\,\rho1\,c2\,c1 + 2\,\xi_2(t)\,V0^2\,\rho\,\rho1\,c0\,c1$$
$$-\,\xi_2(t)\,V0^2\,\rho\,\rho1\,c0^2 + \xi_2(t)\,V0^2\,\rho\,\rho2\,c1^2 + \xi_2(t)\,V0^2\,\rho\,\rho1\,c2^2 - \xi_2(t)\,V0^2\,\rho\,\rho2\,c0^2$$
$$+\,2\,\xi_1(t)\,V0^2\,\rho\,\rho2\,c1\,c0$$

```
> LQP[1][2,1];
```

$$-4\,(\tfrac{d^2}{dt^2}\,\xi_2(t))\,V0^2\,\rho\,\rho1\,\theta^2\,c2^2 + 8\,(\tfrac{d^2}{dt^2}\,\xi_2(t))\,V0^2\,\rho\,\rho1\,\theta^2\,c2\,c1$$
$$-\,4\,(\tfrac{d^2}{dt^2}\,\xi_2(t))\,V0^2\,\rho\,\rho1\,\theta^2\,c1^2 - 4\,\xi_1(t)\,\theta^2\,\sigma2\,c1^2\,c0^2 + 8\,\xi_1(t)\,\theta^2\,\sigma2\,c2^2\,c1^2$$
$$-\,2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho2\,\theta\,c2\,c1 + 2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho2\,\theta\,c2\,c0$$
$$-\,2\,\xi_1(t)\,V0^2\,\rho\,\rho1\,c2\,c1 - \xi_1(t)\,V0^2\,\rho\,\rho2\,c0^2 - 4\,\xi_1(t)\,\theta^2\,\sigma2\,c2^2\,c0^2$$
$$-\,2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho2\,\theta\,c1\,c0 + 2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho1\,c2\,\theta\,c1$$
$$-\,2\,\xi_2(t)\,V0^2\,\sigma1\,c2\,c1 - 4\,\xi_1(t)\,\theta^2\,\sigma2\,c2\,c1^2\,c0 + 8\,\xi_1(t)\,\theta^2\,\sigma2\,c2\,c0^2\,c1$$
$$-\,2\,\xi_1(t)\,V0^2\,\rho\,\rho2\,c2\,c1 + 2\,\xi_1(t)\,V0^2\,\rho\,\rho2\,c2\,c0 + 4\,\xi_1(t)\,\theta^2\,\sigma2\,c2^3\,c0$$
$$-\,4\,\xi_1(t)\,\theta^2\,\sigma2\,c2^3\,c1 + \xi_1(t)\,V0^2\,\rho\,\rho2\,c1^2 - \xi_1(t)\,V0^2\,\sigma1\,c2^2 - \xi_1(t)\,V0^2\,\sigma1\,c1^2$$
$$+\,\xi_2(t)\,V0^2\,\sigma1\,c2^2 + \xi_2(t)\,V0^2\,\sigma1\,c1^2 + 4\,\xi_2(t)\,\theta^2\,\sigma2\,c1^4 + 2\,\xi_1(t)\,V0^2\,\sigma1\,c2\,c1$$
$$-\,4\,\xi_1(t)\,\theta^2\,\sigma2\,c2\,c1^3 - 2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho1\,c2^2\,\theta + 4\,\xi_2(t)\,\theta^2\,\sigma2\,c2^2\,c1^2$$
$$-\,8\,\xi_2(t)\,\theta^2\,\sigma2\,c2\,c1^3 + 4\,\xi_2(t)\,\theta^2\,\sigma2\,c1^2\,c0^2 + 2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho2\,\theta\,c1^2$$
$$+\,2\,\xi_1(t)\,V0^2\,\rho\,\rho1\,c0\,c1 - 2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho1\,c0\,\theta\,c1 - 4\,\xi_1(t)\,\theta^2\,\sigma2\,c2^2\,c1\,c0$$
$$+\,4\,\xi_1(t)\,\theta^2\,\sigma2\,c1^3\,c0 + \xi_1(t)\,V0^2\,\rho\,\rho1\,c2^2 - \xi_1(t)\,V0^2\,\rho\,\rho1\,c0^2$$
$$+\,2\,(\tfrac{d}{dt}\,\xi_1(t))\,V0^2\,\rho\,\rho1\,c0\,\theta\,c2 + 4\,\xi_2(t)\,\theta^2\,\sigma2\,c2^2\,c0^2 - 8\,\xi_2(t)\,\theta^2\,\sigma2\,c2^2\,c1\,c0$$
$$-\,8\,\xi_2(t)\,\theta^2\,\sigma2\,c2\,c0^2\,c1 + 16\,\xi_2(t)\,\theta^2\,\sigma2\,c2\,c1^2\,c0 - 8\,\xi_2(t)\,\theta^2\,\sigma2\,c1^3\,c0$$
$$-\,4\,\xi_2(t)\,V0^2\,\rho\,\rho1\,c2\,c1 - 4\,\xi_2(t)\,V0^2\,\rho\,\rho1\,c0\,c1 + 4\,\xi_2(t)\,V0^2\,\rho\,\rho1\,c1^2$$
$$+\,2\,\xi_2(t)\,V0^2\,\rho\,\rho1\,c2\,c0 + \xi_2(t)\,V0^2\,\rho\,\rho1\,c0^2 + \xi_2(t)\,V0^2\,\rho\,\rho2\,c1^2$$
$$-\,2\,\xi_2(t)\,V0^2\,\rho\,\rho2\,c1\,c0 + \xi_2(t)\,V0^2\,\rho\,\rho1\,c2^2 + \xi_2(t)\,V0^2\,\rho\,\rho2\,c0^2$$

Hence, a necessary condition for optimality is given by the equations

$$LQP[1][1,1] = 0, \quad LQP[1][2,1] = 0. \tag{7.7}$$

Moreover, the boundary terms arising from the integrations by parts are defined by the second entry in LQP:

```
> Boundary := collect(LQP[2], {delta[xi[1]](t),delta[xi[2]](t)});
```

$$\begin{aligned}&\mathit{Boundary} := 2\,\theta\,\mathit{V0}^2\,\rho(\mathit{c2}\,\rho 2\,\xi_2(t)\,\mathit{c0} + 2\,\mathit{c2}^2\,\rho 2\,\xi_1(t) + \mathit{c1}^2\,\rho 2\,\xi_2(t) - \mathit{c1}\,\rho 2\,\xi_2(t)\,\mathit{c0}\\&+ \mathit{c1}\,\rho 2\,\xi_1(t)\,\mathit{c0} + 2\,\mathit{c1}^2\,\rho 2\,(\tfrac{d}{dt}\,\xi_1(t))\,\theta + \mathit{c1}^2\,\rho 2\,\xi_1(t) - \mathit{c2}\,\rho 2\,\xi_1(t)\,\mathit{c0}\\&- 3\,\mathit{c2}\,\rho 2\,\xi_1(t)\,\mathit{c1} + 2\,\mathit{c2}^2\,\rho 2\,(\tfrac{d}{dt}\,\xi_1(t))\,\theta - 4\,\mathit{c2}\,\rho 2\,(\tfrac{d}{dt}\,\xi_1(t))\,\theta\,\mathit{c1} - \mathit{c2}\,\rho 2\,\xi_2(t)\,\mathit{c1})\\&\delta_{\xi_1}(t) + 2\,\theta\,\mathit{V0}^2\,\rho(\mathit{c2}^2\,\rho 1\,\xi_2(t) + 2\,\mathit{c1}^2\,\rho 1\,(\tfrac{d}{dt}\,\xi_2(t))\,\theta + 2\,\mathit{c2}^2\,\rho 1\,(\tfrac{d}{dt}\,\xi_2(t))\,\theta\\&+ \mathit{c2}^2\,\rho 1\,\xi_1(t) - \mathit{c1}\,\rho 1\,\xi_1(t)\,\mathit{c2} + 2\,\mathit{c1}^2\,\rho 1\,\xi_2(t) - \mathit{c1}\,\rho 1\,\xi_2(t)\,\mathit{c0}\\&- 4\,\mathit{c2}\,\rho 1\,(\tfrac{d}{dt}\,\xi_2(t))\,\theta\,\mathit{c1} - \mathit{c2}\,\rho 1\,\xi_1(t)\,\mathit{c0} + \mathit{c1}\,\rho 1\,\xi_1(t)\,\mathit{c0} - 3\,\mathit{c2}\,\rho 1\,\xi_2(t)\,\mathit{c1}\\&+ \mathit{c2}\,\rho 1\,\xi_2(t)\,\mathit{c0})\delta_{\xi_2}(t)\end{aligned}$$

Here δ_{ξ_1} and δ_{ξ_2} are the variations of ξ_1 resp. ξ_2. In order to obtain the initial and final conditions for the equations (7.7), we need to distinguish the case where the terminal value $x(T)$ is fixed and the case where there exists a summand in the cost functional of the form $x(T)^T\,\mathit{C3}\,x(T)$.

In the first case we do not need the boundary terms *Boundary*, but we have to translate the final conditions on $x(T)$ into final conditions on ξ_1, ξ_2 and their derivatives by using the parametrization P of the system.

In the second case the parametrization P of the system is substituted into the term $x(T)^T\,\mathit{C3}\,x(T)$. By computing the variation of this term, we obtain some linear expressions in the variations δ_{ξ_1}, δ_{ξ_2} which we add to the boundary terms that we have already computed.

Let us consider the case where the terminal state $x(T)$ is not measured by an extra term in the cost functional, i.e. $x(T)$ is free.

Comparison of the coefficients of δ_{ξ_1} and δ_{ξ_2} yields that the following two expressions in $\xi_1(T)$, $\xi_2(T)$, $D(\xi_1)(T)$, $D(\xi_2)(T)$ are zero:

```
> B1 := subs(t=T, convert(subs([delta[xi[1]](t)=1,
> delta[xi[2]](t)=0], Boundary), D));
```

$$\begin{aligned}&\mathit{B1} := 2\,\theta\,\mathit{V0}^2\,\rho(\mathit{c2}\,\rho 2\,\xi_2(T)\,\mathit{c0} + 2\,\mathit{c2}^2\,\rho 2\,\xi_1(T) + \mathit{c1}^2\,\rho 2\,\xi_2(T) - \mathit{c1}\,\rho 2\,\xi_2(T)\,\mathit{c0}\\&+ \mathit{c1}\,\rho 2\,\xi_1(T)\,\mathit{c0} + 2\,\mathit{c1}^2\,\rho 2\,\mathrm{D}(\xi_1)(T)\,\theta + \mathit{c1}^2\,\rho 2\,\xi_1(T) - \mathit{c2}\,\rho 2\,\xi_1(T)\,\mathit{c0}\\&- 3\,\mathit{c2}\,\rho 2\,\xi_1(T)\,\mathit{c1} + 2\,\mathit{c2}^2\,\rho 2\,\mathrm{D}(\xi_1)(T)\,\theta - 4\,\mathit{c2}\,\rho 2\,\mathrm{D}(\xi_1)(T)\,\theta\,\mathit{c1} - \mathit{c2}\,\rho 2\,\xi_2(T)\,\mathit{c1})\end{aligned}$$

```
> B2 := subs(t=T, convert(subs([delta[xi[1]](t)=0,
> delta[xi[2]](t)=1], Boundary), D));
```

$$\begin{aligned}&\mathit{B2} := 2\,\theta\,\mathit{V0}^2\,\rho(\mathit{c2}^2\,\rho 1\,\xi_2(T) + 2\,\mathit{c1}^2\,\rho 1\,\mathrm{D}(\xi_2)(T)\,\theta + 2\,\mathit{c2}^2\,\rho 1\,\mathrm{D}(\xi_2)(T)\,\theta + \mathit{c2}^2\,\rho 1\,\xi_1(T)\\&- \mathit{c1}\,\rho 1\,\xi_1(T)\,\mathit{c2} + 2\,\mathit{c1}^2\,\rho 1\,\xi_2(T) - \mathit{c1}\,\rho 1\,\xi_2(T)\,\mathit{c0} - 4\,\mathit{c2}\,\rho 1\,\mathrm{D}(\xi_2)(T)\,\theta\,\mathit{c1}\\&- \mathit{c2}\,\rho 1\,\xi_1(T)\,\mathit{c0} + \mathit{c1}\,\rho 1\,\xi_1(T)\,\mathit{c0} - 3\,\mathit{c2}\,\rho 1\,\xi_2(T)\,\mathit{c1} + \mathit{c2}\,\rho 1\,\xi_2(T)\,\mathit{c0})\end{aligned}$$

The initial conditions are obtained by translating the initial conditions $\mathit{x1}(0) = \mathit{x10}$, $\mathit{x2}(0) = \mathit{x20}$ into initial conditions on ξ_1 and ξ_2:

```
> IC := {subs(t=0, Parametrization(R, Alg)[1,1]=x10),
> subs(t=0, Parametrization(R, Alg)[2,1]=x20)};
```

$$IC := \{2\,\theta\,\xi_1(0)\,c2^2 - 2\,\theta\,\xi_1(0)\,c2\,c1 - 2\,\theta\,\xi_1(0)\,c2\,c0 + 2\,\theta\,\xi_1(0)\,c1\,c0$$
$$+ 2\,\theta\,\xi_2(0)\,c2\,c0 - 2\,\theta\,\xi_2(0)\,c2\,c1 - 2\,\theta\,\xi_2(0)\,c1\,c0 + 2\,\theta\,\xi_2(0)\,c1^2 = x20,$$
$$2\,\theta\,V0\,\xi_1(0)\,c2 - 2\,\theta\,V0\,\xi_1(0)\,c1 - 2\,\theta\,V0\,\xi_2(0)\,c2 + 2\,\theta\,V0\,\xi_2(0)\,c1 = x10\}$$

In order to obtain the optimal trajectories in terms of ξ_1, ξ_2, we need to solve the following system of ordinary differential equations

```
> ODEs := {LQP[1][1,1],LQP[1][2,1]}:
```

with the initial and final conditions:

```
> BoundaryConditions := {B1,B2,op(IC)}:
```

In general it is too difficult to solve this system using symbolic integrations (however, see also the Library of Examples on the `OreModules` web page [CQR06a]). In the present situation we need to know the particular values of the constants θ, $V0$, $c0$, $c1$, $c2$, $\sigma 1$, $\sigma 2$, ρ, $\rho 1$, $\rho 2$, and the final time instant T in order to solve the system of ordinary differential equations numerically.

The values of the parameters are chosen as previously (see p. 10 of [KS72]).

```
> conf := [c0=1.25,c1=1,c2=2,V0=1,theta=50]:
> BVP := subs([op(conf),sigma1=50,sigma2=0.02,rho=0.1,
> rho1=1/3,rho2=3,x10=0.1,x20=0,T=50],
> ODEs union map(convert, BoundaryConditions, D));
```

$$BVP := \{75.00\,\xi_1(0) + 25.00\,\xi_2(0) = 0, -5.000000004\,(\tfrac{d}{dt}\,\xi_2(t)) - 3000.0\,(\tfrac{d^2}{dt^2}\,\xi_1(t))$$
$$+ 163.437499\,\xi_1(t) - 12.33750000\,\xi_2(t),\ 100\,\xi_1(0) - 100\,\xi_2(0) = 0.1,$$
$$5.00000001\,(\tfrac{d}{dt}\,\xi_1(t)) - 12.337500\,\xi_1(t) + 62.57083332\,\xi_2(t)$$
$$- 333.3333333\,(\tfrac{d^2}{dt^2}\,\xi_2(t)),$$
$$4.166666660\,\xi_2(50) + 333.3333334\,\mathrm{D}(\xi_2)(50) + 2.500000000\,\xi_1(50),$$
$$7.500\,\xi_2(50) + 52.500\,\xi_1(50) + 3000.0\,\mathrm{D}(\xi_1)(50)\}$$

This defines the boundary value problem which we solve numerically:

```
> dsol5 := dsolve(BVP, {xi[1](t),xi[2](t)}, type=numeric);
```

$$dsol5 := \mathbf{proc}(x\_bvp)\ \ldots\ \mathbf{end\ proc}$$

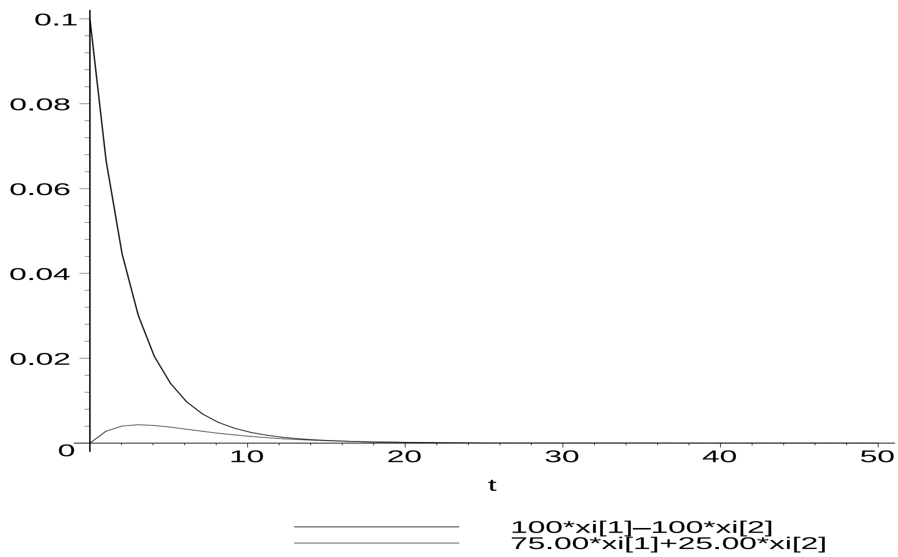
In order to find the trajectories for $x1$, $x2$, $u1$, $u2$ we have to adjust the parametrization $LQP[3]$ of the system (which coincides with the result of `Parametrization`) to the chosen values for the constants. We also rename the arbitrary functions ξ_1, ξ_2 in $LQP[3]$ because we are going to substitute the solutions of the boundary value problem for these functions.

```
> P_ex := subs([op(conf),xi=eta], LQP[3]);
```

$$P\_ex := \begin{bmatrix} 100\,\eta_1(t) - 100\,\eta_2(t) \\ 75.00\,\eta_1(t) + 25.00\,\eta_2(t) \\ -0.75\,\eta_1(t) - 1.25\,\eta_2(t) - 100\,(\frac{d}{dt}\,\eta_2(t)) \\ 1.75\,\eta_1(t) + 100\,(\frac{d}{dt}\,\eta_1(t)) + 0.25\,\eta_2(t) \end{bmatrix}$$

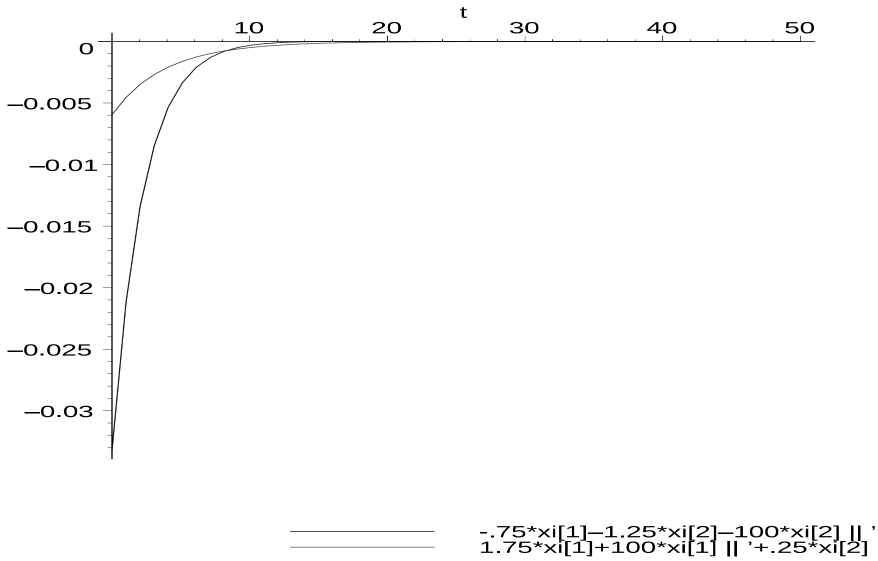
The first two components of the parametrization yield the trajectories for *x1*, *x2*:

```
> plots[odeplot](dsol5,
> [[t,subs([eta[1](t)=xi[1](t), eta[2](t)=xi[2](t)], P_ex[1,1]),
> color=blue],
> [t,subs([eta[1](t)=xi[1](t), eta[2](t)=xi[2](t)], P_ex[2,1]),
> color=red]], 0..50);
```



The third and the fourth component of the parametrization give the trajectories for *u1*, *u2*:

```
> plots[odeplot](dsol5, [[t,subs([eta[1](t)=xi[1](t),
> eta[2](t)=xi[2](t)], P_ex[3,1]), color=blue],
> [t,subs([eta[1](t)=xi[1](t), eta[2](t)=xi[2](t)], P_ex[4,1]),
> color=red]], 0..50);
```



Here we have chosen $\rho = 0.1$. The plots for *x1*, *x2* resp. *u1*, *u2* displayed above can be compared to the ones given in [KS72], p. 228 and 229 (first column).

7.7 A discrete-time model

In this section we assume that the stirred tank is commanded by a computer, which means that the values of the input variables change at discrete time instants only (see [KS72], p. 449). The discrete-time description

$$x(i+1) = \widetilde{A}(i)\, x(i) + \widetilde{B}(i)\, u(i)$$

is derived from the continuous-time description

$$\dot{x}(t) = A(t)\, x(t) + B(t)\, u(t)$$

as follows. Using the transition matrix $\Phi(t', t)$ of the system we have

$$x(t') = \Phi(t', t)\, x(t) + \int_t^{t'} \Phi(t', \tau)\, B(\tau)\, d\tau\, u(t).$$

By substituting t_i for t and t_{i+1} for t' we find

$$\widetilde{A}(i) = \Phi(t_{i+1}, t_i), \qquad \widetilde{B}(i) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)\, B(\tau)\, d\tau.$$

The continuous-time description of the stirred tank is time-invariant. Hence, we have $\Phi(t', t) = \exp(A\,(t' - t))$.

When each two consecutive time instants t_i, t_{i+1} differ by the same period of time $\Delta > 0$, then the discrete-time description is also time-invariant. For the stirred tank we obtain

$$\widetilde{A} = \begin{pmatrix} e^{-\Delta/(2\,\theta)} & 0 \\ 0 & e^{-\Delta/\theta} \end{pmatrix}, \quad \widetilde{B} = \begin{pmatrix} 2\,\theta(1 - e^{-\Delta/(2\,\theta)}) & 2\,\theta(1 - e^{-\Delta/(2\,\theta)}) \\ \frac{\theta\,(c1 - c0)}{V0}\,(1 - e^{-\Delta/\theta}) & \frac{\theta\,(c2 - c0)}{V0}\,(1 - e^{-\Delta/\theta}) \end{pmatrix}.$$

In order to deal with this time-invariant discrete-time system using `OreModules`, we define the Ore algebra $\mathcal{O} = \mathbb{Q}(\theta, V0, c0, c1, c2, \kappa)[\sigma]$ of polynomials in σ with coefficients that are rational functions in the parameters θ, $V0$, $c0$, $c1$, $c2$, and $\kappa := \exp(-\Delta/(2\,\theta))$, where σ represents the shift operator $(\sigma\, x)(i) = x(i+1)$.

```
> Alg2 := DefineOreAlgebra(shift=[sigma,i], polynom=[i],
> comm=[theta,V0,c0,c1,c2,kappa]):
```

The system matrix is:

```
> R2 := matrix([[sigma-kappa,0,-2*theta*(1-kappa),
> -2*theta*(1-kappa)], [0,sigma-kappa^2,-theta*(c1-c0)*
> (1-kappa^2)/V0,-theta*(c2-c0)*(1-kappa^2)/V0]]);
```

$$R2 := \begin{bmatrix} \sigma - \kappa & 0 & -2\,\theta\,(1-\kappa) & -2\,\theta\,(1-\kappa) \\ 0 & \sigma - \kappa^2 & -\dfrac{\theta\,(c1 - c0)\,(1-\kappa^2)}{V0} & -\dfrac{\theta\,(c2 - c0)\,(1-\kappa^2)}{V0} \end{bmatrix}$$

Let $M2$ be the $\mathcal{O}$-module which is associated with this linear system. In order to check parametrizability, we are going to compute the extension group $\mathrm{ext}^1_{\mathcal{O}}(M2^{\top}, \mathcal{O})$ of the transposed module of $M2$. Since we deal with a linear system with constant coefficients, the involution is just transposition of matrices.

```
> R2_adj := linalg[transpose](R2);
```

$$R2\_adj := \begin{bmatrix} \sigma - \kappa & 0 \\ 0 & \sigma - \kappa^2 \\ -2\,\theta\,(1-\kappa) & -\dfrac{\theta\,(c1 - c0)\,(1-\kappa^2)}{V0} \\ -2\,\theta\,(1-\kappa) & -\dfrac{\theta\,(c2 - c0)\,(1-\kappa^2)}{V0} \end{bmatrix}$$

We compute $\mathrm{ext}^1_{\mathcal{O}}(M2^{\top}, \mathcal{O})$.

```
> Ext2 := Exti(R2_adj, Alg2, 1);
```

$$Ext2 := [\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} \sigma - \kappa & 0 & 2\,\theta\,(-1+\kappa) & 2\,\theta\,(-1+\kappa) \\ 0 & \sigma\,V0 - V0\,\kappa^2 & -(c1 - c1\,\kappa^2 - c0 + c0\,\kappa^2)\,\theta & -(c2 - c2\,\kappa^2 - c0 + c0\,\kappa^2)\,\theta \end{bmatrix},$$

$$[2\,V0\,(-1+\kappa)\,\theta\,(-c2+c1),\, 2\,V0\,(-1+\kappa)\,\theta\,(-c2+c1)]$$

$$\Big[-\theta\,(c2^2\,\kappa^2 - c2\,\kappa^2\,c1 - c2^2 + c2\,c1 - c2\,c0\,\kappa^2 + c1\,c0\,\kappa^2 + c2\,c0 - c1\,c0),$$

$$-\theta\,(c2\,\kappa^2\,c1 - c2\,c1 - c2\,c0\,\kappa^2 + c1\,c0\,\kappa^2 + c2\,c0 - c1\,c0 - c1^2\,\kappa^2 + c1^2)\Big]$$

$$[-V0\,\kappa\,(c2 - c0 - \kappa\,c2 + \kappa\,c0),\, -V0\,\kappa\,(c2 - c0 + \kappa\,c0 - \kappa\,c1) - V0\,(-c2+c1)\,\sigma]$$

$$[V0\,\kappa\,(-\kappa\,c2 + \kappa\,c0 + c1 - c0) - V0\,(-c2+c1)\,\sigma,\; V0\,\kappa\,(c1 - c0 - \kappa\,c1 + \kappa\,c0)]]$$

The residue classes in $M2$ of the rows of $Ext2[2]$ generate the torsion submodule $t(M2)$, and the elements in the i-th column of $Ext2[1]$ annihilate the residue class represented by the i-th row of $Ext2[2]$. Therefore, $t(M2) = 0$ and the system is controllable for generic configurations of the parameters. $Ext2[3]$ provides a parametrization of the behavior. In terms of arbitrary functions $\xi_1(i)$, $\xi_2(i)$ this parametrization is also obtained as follows:

```
> Parametrization(R2, Alg2);
```

$$\Big[-2\,V0\,\theta\,\xi_1(i)\,\kappa\,c2 + 2\,V0\,\theta\,\xi_1(i)\,\kappa\,c1 + 2\,V0\,\theta\,\xi_1(i)\,c2 - 2\,V0\,\theta\,\xi_1(i)\,c1$$
$$-2\,V0\,\theta\,\xi_2(i)\,\kappa\,c2 + 2\,V0\,\theta\,\xi_2(i)\,\kappa\,c1 + 2\,V0\,\theta\,\xi_2(i)\,c2 - 2\,V0\,\theta\,\xi_2(i)\,c1\Big]$$

$$\Big[-\theta\,\xi_1(i)\,c2^2\,\kappa^2 + \theta\,\xi_1(i)\,c2\,\kappa^2\,c1 + \theta\,\xi_1(i)\,c2^2 - \theta\,\xi_1(i)\,c2\,c1 + \theta\,\xi_1(i)\,c2\,c0\,\kappa^2$$
$$-\theta\,\xi_1(i)\,c1\,c0\,\kappa^2 - \theta\,\xi_1(i)\,c2\,c0 + \theta\,\xi_1(i)\,c1\,c0 - \theta\,\xi_2(i)\,c2\,\kappa^2\,c1 + \theta\,\xi_2(i)\,c2\,c1$$
$$+\theta\,\xi_2(i)\,c2\,c0\,\kappa^2 - \theta\,\xi_2(i)\,c1\,c0\,\kappa^2 - \theta\,\xi_2(i)\,c2\,c0 + \theta\,\xi_2(i)\,c1\,c0 + \theta\,\xi_2(i)\,c1^2\,\kappa^2$$
$$-\theta\,\xi_2(i)\,c1^2\Big]$$

$$\Big[-V0\,\kappa\,\xi_1(i)\,c2 + V0\,\kappa\,\xi_1(i)\,c0 + V0\,\kappa^2\,\xi_1(i)\,c2 - V0\,\kappa^2\,\xi_1(i)\,c0 - V0\,\kappa\,\xi_2(i)\,c2$$
$$+V0\,\kappa\,\xi_2(i)\,c0 - V0\,\kappa^2\,\xi_2(i)\,c0 + V0\,\kappa^2\,\xi_2(i)\,c1 + V0\,\xi_2(i+1)\,c2$$
$$-V0\,\xi_2(i+1)\,c1\Big]$$

$$\Big[-V0\,\kappa^2\,\xi_1(i)\,c2 + V0\,\kappa^2\,\xi_1(i)\,c0 + V0\,\kappa\,\xi_1(i)\,c1 - V0\,\kappa\,\xi_1(i)\,c0 + V0\,\xi_1(i+1)\,c2$$
$$-V0\,\xi_1(i+1)\,c1 + V0\,\kappa\,\xi_2(i)\,c1 - V0\,\kappa\,\xi_2(i)\,c0 - V0\,\kappa^2\,\xi_2(i)\,c1 + V0\,\kappa^2\,\xi_2(i)\,c0\Big]$$

Since $\mathcal{O}$ is a principal ideal ring, $M2$ is also projective and free. The fact that $M2$ is projective can also be verified as in Section 7.2.

```
> SyzygyModule(R2, Alg2);
```

INJ(2)

```
> T := map(factor, RightInverse(R2, Alg2));
```

$$T := \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{c0 - c2}{2\,\theta\,(\kappa - 1)\,(c1 - c2)} & \dfrac{V0}{(\kappa - 1)\,(\kappa + 1)\,(c1 - c2)\,\theta} \\ -\dfrac{c0 - c1}{2\,(\kappa - 1)\,\theta\,(c1 - c2)} & -\dfrac{V0}{(\kappa - 1)\,(\kappa + 1)\,(c1 - c2)\,\theta} \end{bmatrix}$$

Hence, we have just proved that the short exact sequence

$$0 \longrightarrow \mathcal{O}^{1\times 2} \xrightarrow{.R2} \mathcal{O}^{1\times 4} \longrightarrow M2 \longrightarrow 0$$

splits (see Def. 6.3.1), which is equivalent to the fact that $M2$ is projective. The module $M2$ is free which means that the system is flat for generic configurations of the parameters. Let us compute a flat output of its behavior.

```
> S := map(factor, LeftInverse(Ext2[3], Alg2));
```

$$S := \begin{bmatrix} -\dfrac{c0 - c1}{2\,(c1 - c2)^2\,V0\,(\kappa - 1)\,\theta} & -\dfrac{1}{(c1 - c2)^2\,(\kappa - 1)\,(\kappa + 1)\,\theta} & 0 & 0 \\ \dfrac{c0 - c2}{2\,(c1 - c2)^2\,V0\,(\kappa - 1)\,\theta} & \dfrac{1}{(c1 - c2)^2\,(\kappa - 1)\,(\kappa + 1)\,\theta} & 0 & 0 \end{bmatrix}$$

A flat output is therefore defined by $(\xi 1 \quad \xi 2)^T = S\,(x1 \quad x2 \quad u1 \quad u2)^T$ and $(\xi 1 \quad \xi 2)^T$ satisfies $(x1 \quad x2 \quad u1 \quad u2)^T = Ext2[3]\,(\xi 1 \quad \xi 2)^T$. More explicitly, we have:

```
> matrix([[xi1(i)],[xi2(i)]])=
> ApplyMatrix(S, [x1(i),x2(i),u1(i),u2(i)], Alg2);
```

$$\begin{bmatrix} \xi 1(i) \\ \xi 2(i) \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{2}\,\dfrac{\mathrm{x1}(i)\,\kappa\,c0 + \mathrm{x1}(i)\,c0 - \mathrm{x1}(i)\,\kappa\,c1 - \mathrm{x1}(i)\,c1 + 2\,\mathrm{x2}(i)\,V0}{(c1 - c2)^2\,V0\,\theta\,(\kappa^2 - 1)} \\ \dfrac{1}{2}\,\dfrac{-\mathrm{x1}(i)\,\kappa\,c2 + \mathrm{x1}(i)\,\kappa\,c0 - \mathrm{x1}(i)\,c2 + \mathrm{x1}(i)\,c0 + 2\,\mathrm{x2}(i)\,V0}{(c1 - c2)^2\,V0\,\theta\,(\kappa^2 - 1)} \end{bmatrix}$$

We substitute $(\xi 1 \quad \xi 2)^T = S\,(x1 \quad x2 \quad u1 \quad u2)^T$ into $(x1 \quad x2 \quad u1 \quad u2)^T = Ext2[3]\,(\xi 1 \quad \xi 2)^T$.

```
> P2 := Mult(Ext2[3], S2, Alg2);
```

$P2 :=$
$[-V0\,(c2 - c1 - \kappa\,c2 + \kappa\,c1)\,,\,0\,,$
$2\;V0\,\theta\,c2 - 2\;V0\,\theta\,c1 - 2\;V0\,\theta\,\kappa\,c2 + 2\;V0\,\theta\,\kappa\,c1\;,\,0]$
$\left[\frac{c2^2\,\kappa^2}{2} - \frac{c2\,\kappa^2\,c1}{2} - \frac{c2^2}{2} + \frac{c2\;c1}{2} - \frac{c2\;c0\,\kappa^2}{2} + \frac{c1\;c0\,\kappa^2}{2} + \frac{c2\;c0}{2} - \frac{c1\;c0}{2}\,,\,0\,,\right.$
$-\theta\,c2\,\kappa^2\,c1 + \theta\,c2\,c1 + \theta\,c2\,c0\,\kappa^2 - \theta\,c1\;c0\,\kappa^2 - \theta\,c2\;c0 + \theta\,c1\;c0 + \theta\,c1^2\,\kappa^2$
$\left.-\,\theta\,c1^2,\,0\right]$
$\left[\frac{V0\,\kappa\,(c2 - c0 - \kappa\,c2 + \kappa\,c0)}{2\,\theta}\,,\,0\,,\right.$
$\left.-\,V0\,\kappa\,c2 +\,V0\,\kappa\,c0 -\,V0\,\kappa^2\,c0 +\,V0\,\kappa^2\,c1 +\,V0\,\sigma\,c2 -\,V0\,\sigma\,c1\;,\,0\right]$
$\left[-\,\frac{V0\,(-c2\,\kappa^2 + c0\,\kappa^2 + \kappa\,c1 - \kappa\,c0 + \sigma\,c2 - \sigma\,c1)}{2\,\theta}\,,\,0\,,\right.$
$\left.-\,V0\,\kappa^2\,c1 +\,V0\,\kappa\,c1 +\,V0\,\kappa^2\,c0 -\,V0\,\kappa\,c0\;,\,0\right]$

Obviously, $(x1 \quad x2)^T$ is another flat output of the behavior.

```
>  matrix([[x1(i)],[x2(i)],[u1(i)],[u2(i)]])=
>  ApplyMatrix(P2, [x1(i),x2(i),u1(i),u2(i)], Alg2);
```

$\begin{bmatrix} \mathrm{x1}(i) \\ \mathrm{x2}(i) \\ \mathrm{u1}(i) \\ \mathrm{u2}(i) \end{bmatrix} =$
$[-\,V0\,\mathrm{x1}(i)\;c2 +\,V0\,\mathrm{x1}(i)\;c1 +\,V0\,\mathrm{x1}(i)\,\kappa\,c2 -\,V0\,\mathrm{x1}(i)\,\kappa\,c1 + 2\,\mathrm{u1}(i)\;V0\,\theta\,c2$
$-\,2\,\mathrm{u1}(i)\;V0\,\theta\,c1 - 2\,\mathrm{u1}(i)\;V0\,\theta\,\kappa\,c2 + 2\,\mathrm{u1}(i)\;V0\,\theta\,\kappa\,c1]$
$\left[\frac{1}{2}\,\mathrm{x1}(i)\;c2^2\,\kappa^2 - \frac{1}{2}\,\mathrm{x1}(i)\;c2\,\kappa^2\;c1 - \frac{1}{2}\,\mathrm{x1}(i)\;c2^2 + \frac{1}{2}\,\mathrm{x1}(i)\;c2\;c1 - \frac{1}{2}\,\mathrm{x1}(i)\;c2\;c0\,\kappa^2\right.$
$+\,\frac{1}{2}\,\mathrm{x1}(i)\;c1\;c0\,\kappa^2 + \frac{1}{2}\,\mathrm{x1}(i)\;c2\;c0 - \frac{1}{2}\,\mathrm{x1}(i)\;c1\;c0 - \mathrm{u1}(i)\,\theta\,c2\,\kappa^2\;c1 + \mathrm{u1}(i)\,\theta\,c2\;c1$
$+\,\mathrm{u1}(i)\,\theta\,c2\;c0\,\kappa^2 - \mathrm{u1}(i)\,\theta\,c1\;c0\,\kappa^2 - \mathrm{u1}(i)\,\theta\,c2\;c0 + \mathrm{u1}(i)\,\theta\,c1\;c0$
$\left.+\,\mathrm{u1}(i)\,\theta\,c1^2\,\kappa^2 - \mathrm{u1}(i)\,\theta\,c1^2\right]$
$\left[-\,\frac{1}{2}\,V0(-\mathrm{x1}(i)\,\kappa\,c2 + \mathrm{x1}(i)\,\kappa\,c0 + \kappa^2\,\mathrm{x1}(i)\;c2 - \kappa^2\,\mathrm{x1}(i)\;c0 + 2\,\kappa\,\mathrm{u1}(i)\,\theta\,c2\right.$
$\left.-\,2\,\kappa\,\mathrm{u1}(i)\,\theta\,c0 + 2\,\kappa^2\,\mathrm{u1}(i)\,\theta\,c0 - 2\,\kappa^2\,\mathrm{u1}(i)\,\theta\,c1 - 2\,\mathrm{u1}(i+1)\,\theta\,c2 + 2\,\mathrm{u1}(i+1)\,\theta\,c1)/\theta\right]$
$\left[\frac{1}{2}\,V0(\kappa^2\,\mathrm{x1}(i)\;c2 - \kappa^2\,\mathrm{x1}(i)\;c0 - \mathrm{x1}(i)\,\kappa\,c1 + \mathrm{x1}(i)\,\kappa\,c0 - \mathrm{x1}(i+1)\;c2\right.$
$\left.+\,\mathrm{x1}(i+1)\;c1 + 2\,\kappa\,\mathrm{u1}(i)\,\theta\,c1 - 2\,\kappa\,\mathrm{u1}(i)\,\theta\,c0 - 2\,\kappa^2\,\mathrm{u1}(i)\,\theta\,c1 + 2\,\kappa^2\,\mathrm{u1}(i)\,\theta\,c0)/\theta\right]$

We conclude that the discrete-time model of the stirred tank is flat if *c1* $\neq$ *c2* and $\kappa^2 \neq 1$. By definition we have $\kappa = \exp(-\Delta/2\theta)$, so that $\kappa^2 \neq 1$ always holds because $\Delta > 0$. As in the continuous-time case, the flatness of the system can be exploited for motion planning (see Section 7.5).

Let us consider the system for the case *c0* = *c1* = *c2*.

```
> R2mod := subs([c2=c0,c1=c0], evalm(R2));
```

$$R2mod := \begin{bmatrix} \sigma - \kappa & 0 & -2\,\theta\,(1-\kappa) & -2\,\theta\,(1-\kappa) \\ 0 & \sigma - \kappa^2 & 0 & 0 \end{bmatrix}$$

We are going to check controllability, parametrizability and flatness for this particular setting.

```
> Ext2mod := Exti(linalg[transpose](R2mod), Alg2, 1);
```

$$Ext2mod := \left[\begin{bmatrix} 1 & 0 \\ 0 & \sigma - \kappa^2 \end{bmatrix}, \begin{bmatrix} \sigma - \kappa & 0 & -2\,\theta + 2\,\theta\,\kappa & -2\,\theta + 2\,\theta\,\kappa \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} -2\,\theta\,(\kappa - 1) & 0 \\ 0 & 0 \\ 0 & 1 \\ \sigma - \kappa & -1 \end{bmatrix} \right]$$

As in the continuous time case we find a non-trivial torsion element m in the $\mathcal{O}$-module which is associated with the system, namely the residue class which is represented by *x2*. The torsion element satisfies $(\sigma - \kappa^2)\, m = 0$. Therefore, the system is not controllable and not parametrizable in this case. Since free $\mathcal{O}$-modules are torsion-free, it follows by contraposition that the system is not flat. A generating set of torsion elements together with their annihilators is given in terms of functions by the command `TorsionElements`:

```
> TorsionElements(R2mod, [x1(i),x2(i),u1(i),u2(i)], Alg2);
```

$$[[\ -\kappa^2\,\theta_2(i) + \theta_2(i+1) = 0\ ],\ [\ \theta_2(i) = \mathrm{x2}(i)\ ]]$$

In a similar way as in Section 7.3 the controllable part of the system can be studied by considering the presentation matrix *Ext2mod*[2] and the parametrization *Ext2mod*[3].

7.8 A differential time-delay model

In this section we study a different model of the stirred tank which is described in [KS72], pages 449–452. The two feeds with concentrations *c1* resp. *c2* are mixed before they flow into the tank and the result is fed through a pipe as the only incoming flow. The length of the pipe causes a transport delay τ. In [KS72] the mass balances equations for this setting are linearized again and lead to the following model of linear ordinary differential equations with a shift (compare to (7.1)):

$$\dot{x}(t) = \begin{pmatrix} -\frac{1}{2\,\theta} & 0 \\ 0 & -\frac{1}{\theta} \end{pmatrix} x(t) + \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} u(t) + \begin{pmatrix} 0 & 0 \\ \frac{c1-c0}{V0} & \frac{c2-c0}{V0} \end{pmatrix} u(t-\tau).$$

For the application of `OreModules` to this differential time-delay model we define the Ore algebra $\mathcal{O} := \mathbb{Q}(\theta, V0, c0, c1, c2)[t, s][\mathrm{D}, \delta]$ of polynomials in D and δ with coefficients that are polynomials in t and s obeying the following commutation rules:

$$\begin{array}{rclcrcl} \mathrm{D}\, t &=& t\,\mathrm{D} + 1, && \mathrm{D}\, s &=& s\,\mathrm{D}, \\ \delta\, s &=& (s-1)\,\sigma, && \delta\, t &=& t\,\delta, \\ \mathrm{D}\,\delta &=& \delta\,\mathrm{D}. &&&& \end{array}$$

Hence, D represents differentiation with respect to time t and δ is a shift operator. Since the equations under consideration have constant coefficients, it is not a problem that we have $\delta\, t = t\,\delta$, but the option `shift_action=[delta,t,h]` in the following declaration specifies that, whenever a matrix with entries in *Alg3* is applied to a vector of functions, then δ acts by a shift of "length" h in t.

```
> Alg3 := DefineOreAlgebra(diff=[D,t], dual_shift=[delta,s],
> polynom=[t,s], comm=[theta,V0,c0,c1,c2],
> shift_action=[delta,t,h]):
```

The new system is described by the following matrix $R3 \in \mathcal{O}^{2\times 4}$.

```
> R3 := matrix([[D+1/(2*theta),0,-1,-1],
> [0,D+1/theta,-(c1-c0)*delta/V0,-(c2-c1)*delta/V0]]);
```

$$R3 := \begin{bmatrix} \mathrm{D} + \dfrac{1}{2\,\theta} & 0 & -1 & -1 \\ 0 & \mathrm{D} + \dfrac{1}{\theta} & -\dfrac{(c1-c0)\,\delta}{V0} & -\dfrac{(c2-c1)\,\delta}{V0} \end{bmatrix}$$

Let us denote the $\mathcal{O}$-module which is associated with this linear system by *M3*.

In order to check structural properties of the system, we compute the extension group $\mathrm{ext}^1_{\mathcal{O}}(M3^{\top}, \mathcal{O})$. The involution is just transposition of matrices because the equations have constant coefficients.

```
> Ext1 := Exti(linalg[transpose](R3), Alg3, 1);
```

$$\mathit{Ext1} := \left[\left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right], \left[\begin{array}{cccc} -2\,\mathrm{D}\,\theta - 1 & 0 & 2\,\theta & 2\,\theta \\ 0 & \theta\, \mathit{V0}\,\mathrm{D} + \mathit{V0} & \theta\,(-\mathit{c1} + \mathit{c0})\,\delta & (-\mathit{c2} + \mathit{c1})\,\theta\,\delta \end{array} \right], \right.$$

$$\left. \left[\begin{array}{c} -2\,\mathit{V0}\,\theta\,(\mathit{c2} + \mathit{c0} - 2\,\mathit{c1})\,,\; -2\,\mathit{V0}\,\theta\,(\mathit{c2} + \mathit{c0} - 2\,\mathit{c1}) \\ 2\,\delta\,(\mathit{c2} + \mathit{c0} - 2\,\mathit{c1})\,\theta\,(-\mathit{c2} + \mathit{c1})\,,\; 2\,(\mathit{c2} + \mathit{c0} - 2\,\mathit{c1})\,\delta\,\theta\,(-\mathit{c1} + \mathit{c0}) \\ -\mathit{V0}\,(-\mathit{c2} + \mathit{c1})\,,\; -(2\,\mathit{c0} - 3\,\mathit{c1} + \mathit{c2})\,\mathit{V0} - 2\,(\mathit{c2} + \mathit{c0} - 2\,\mathit{c1})\,\mathit{V0}\,\theta\,\mathrm{D} \\ -(-3\,\mathit{c1} + \mathit{c0} + 2\,\mathit{c2})\,\mathit{V0} - 2\,(\mathit{c2} + \mathit{c0} - 2\,\mathit{c1})\,\mathit{V0}\,\theta\,\mathrm{D}\,,\; \mathit{V0}\,(-\mathit{c1} + \mathit{c0}) \end{array} \right] \right]$$

The residue classes in $\mathit{M3}$ of the rows of $\mathit{Ext1}[2]$ generate the torsion submodule $t(\mathit{M3})$ of $\mathit{M3}$. The i-th generator is annihilated by the entries of the i-th column of $\mathit{Ext1}[1]$, so that we conclude that $t(\mathit{M3}) = 0$. Hence, the present stirred tank model is controllable and parametrizable for generic configurations of the parameters. The third matrix in $\mathit{Ext1}$ provides a parametrization.

The same parametrization is given in terms of arbitrary (smooth) functions ξ_1, ξ_2 by the command `Parametrization` (note that the action of δ on functions is by shift on t):

```
> Parametrization(R3, Alg3);
```

$$\Big[-2\,\theta\,\mathit{V0}\,\xi_1(t)\,\mathit{c2} - 2\,\mathit{V0}\,\theta\,\xi_1(t)\,\mathit{c0} + 4\,\theta\,\mathit{V0}\,\xi_1(t)\,\mathit{c1} - 2\,\theta\,\mathit{V0}\,\xi_2(t)\,\mathit{c2}$$
$$-\,2\,\mathit{V0}\,\theta\,\xi_2(t)\,\mathit{c0} + 4\,\theta\,\mathit{V0}\,\xi_2(t)\,\mathit{c1}\Big]$$
$$\Big[-2\,\theta\,\xi_1(t-h)\,\mathit{c2}^2 + 6\,\theta\,\xi_1(t-h)\,\mathit{c2}\,\mathit{c1} - 2\,\theta\,\xi_1(t-h)\,\mathit{c2}\,\mathit{c0} + 2\,\theta\,\xi_1(t-h)\,\mathit{c1}\,\mathit{c0}$$
$$-\,4\,\theta\,\xi_1(t-h)\,\mathit{c1}^2 - 2\,\theta\,\xi_2(t-h)\,\mathit{c2}\,\mathit{c1} - 6\,\theta\,\xi_2(t-h)\,\mathit{c1}\,\mathit{c0} + 4\,\theta\,\xi_2(t-h)\,\mathit{c1}^2$$
$$+\,2\,\theta\,\xi_2(t-h)\,\mathit{c2}\,\mathit{c0} + 2\,\theta\,\xi_2(t-h)\,\mathit{c0}^2\Big]$$
$$\Big[\mathit{V0}\,\xi_1(t)\,\mathit{c2} - \mathit{V0}\,\xi_1(t)\,\mathit{c1} - 2\,\mathit{V0}\,\xi_2(t)\,\mathit{c0} + 3\,\mathit{V0}\,\xi_2(t)\,\mathit{c1} - \mathit{V0}\,\xi_2(t)\,\mathit{c2}$$
$$-\,2\,\mathit{V0}\,\theta\,\mathrm{D}(\xi_2)(t)\,\mathit{c2} - 2\,\mathit{V0}\,\theta\,\mathrm{D}(\xi_2)(t)\,\mathit{c0} + 4\,\mathit{V0}\,\theta\,\mathrm{D}(\xi_2)(t)\,\mathit{c1}\Big]$$
$$\Big[3\,\mathit{V0}\,\xi_1(t)\,\mathit{c1} - \mathit{V0}\,\xi_1(t)\,\mathit{c0} - 2\,\mathit{V0}\,\xi_1(t)\,\mathit{c2} - 2\,\mathit{V0}\,\theta\,\mathrm{D}(\xi_1)(t)\,\mathit{c2}$$
$$-\,2\,\mathit{V0}\,\theta\,\mathrm{D}(\xi_1)(t)\,\mathit{c0} + 4\,\mathit{V0}\,\theta\,\mathrm{D}(\xi_1)(t)\,\mathit{c1} - \mathit{V0}\,\xi_2(t)\,\mathit{c1} + \mathit{V0}\,\xi_2(t)\,\mathit{c0}\Big]$$

In order to check whether the system is flat, we have to investigate whether $\mathit{M3}$ is projective. Therefore, we need to compute the second extension group $\mathrm{ext}^2_{\mathcal{O}}(\mathit{M3}^\top, \mathcal{O})$ of $\mathit{M3}^\top$ (see Theorem 5.1.5 and Figure 5.1, p. 99).

```
> Ext2 := Exti(linalg[transpose](R3), Alg3, 2);
```

$$Ext2 := \left[\left[\begin{array}{cc} \delta & 0 \\ 1+\mathrm{D}\,\theta & 0 \\ 0 & \delta \\ 0 & -\mathrm{D}\,\theta-1 \end{array}\right], \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right], \mathrm{SURJ}(2)\right]$$

The representatives of generators for $\mathrm{ext}^2_{\mathcal{O}}(M3^\top, \mathcal{O})$ given by the rows of $Ext2[2]$ are annihilated by the entries in $Ext2[1]$ as explained before. We conclude $\mathrm{ext}^2_{\mathcal{O}}(M3^\top, \mathcal{O}) \neq 0$, which means that $M3$ is not projective. In particular, $M3$ is not free. Therefore, the system is not flat. This is coherent with the fact that the parametrization $Ext1[3]$ does not induce an injective homomorphism:

```
> LeftInverse(Ext1[3], Alg3);
```

$$[\,]$$

However, we can compute a so-called "π-polynomial" (see [Mou95], [CQR05]) $\pi \in \mathbb{Q}(\theta, V0, c0, c1, c2)[\delta]$ such that

$$S^{-1}\,\mathcal{O} \otimes M3 = \{\, m/a \mid m \in M3,\ a = \pi^j,\ j \in \mathbb{Z}_{\geq 0} \,\}$$

is a free $S^{-1}\,\mathcal{O}$-module, where $S := \{\, \pi^j \mid j \in \mathbb{Z}_{\geq 0} \,\}$ is the multiplicatively closed subset of the commutative ring $\mathcal{O}$ generated by π. This means, if we allow to invert the π-polynomial, then the system, considered over this new algebra, becomes flat. Let us compute a polynomial $\pi \in \mathbb{Q}(\theta, V0, c0, c1, c2)[\delta]$ with these properties:

```
> PiPolynomial(R3, Alg3, [delta]);
```

$$[\delta]$$

Hence, if we introduce the advance operator δ^{-1}, then the system is flat. Let us compute a flat output of its behavior:

```
> S := map(factor, LocalLeftInverse(Ext1[3], [delta], Alg3));
```

$$S := \left[\begin{array}{cccc} -\dfrac{-c1+c0}{2\,(c2+c0-2\,c1)^2\,V0\,\theta} & -\dfrac{1}{2\,\delta\,\theta\,(c2+c0-2\,c1)^2} & 0 & 0 \\ \dfrac{-c2+c1}{2\,(c2+c0-2\,c1)^2\,V0\,\theta} & \dfrac{1}{2\,\delta\,\theta\,(c2+c0-2\,c1)^2} & 0 & 0 \end{array}\right]$$

A flat output is defined by $(\xi 1 \quad \xi 2)^T = S\,(x1 \quad x2 \quad u1 \quad u2)^T$.

```
> T := simplify(evalm(Ext1[3] &* S));
```

$$T := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\dfrac{(2\,\mathrm{D}\,\theta+1)\,(-c2+c1)}{2\,\theta\,(c2+c0-2\,c1)} & -\dfrac{(1+\mathrm{D}\,\theta)\,V0}{(c2+c0-2\,c1)\,\theta\,\delta} & 0 & 0 \\ \dfrac{(2\,\mathrm{D}\,\theta+1)\,(-c1+c0)}{2\,\theta\,(c2+c0-2\,c1)} & \dfrac{(1+\mathrm{D}\,\theta)\,V0}{(c2+c0-2\,c1)\,\theta\,\delta} & 0 & 0 \end{bmatrix}$$

The previous computation shows that we can also choose $(x1 \quad x2)^T$ as a flat output. The third and the fourth row of T express the input variables $u1$, $u2$ in terms of $x1$, $x2$. Note that they involve the advance operator.

```
> P := ApplyMatrix(T, [x1(t),x2(t),u1(t),u2(t)], Alg3):
> matrix([[x1(t)],[x2(t)],[u1(t)],[u2(t)]])=evalm(P);
```

$$\begin{bmatrix} \mathrm{x1}(t) \\ \mathrm{x2}(t) \\ \mathrm{u1}(t) \\ \mathrm{u2}(t) \end{bmatrix} =$$

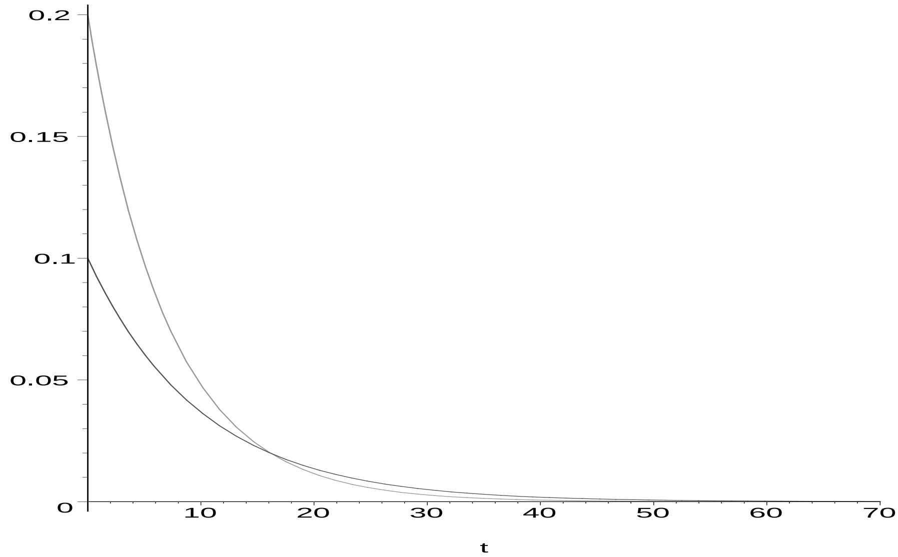
$$[\mathrm{x1}(t)]$$
$$[\mathrm{x2}(t)]$$
$$\Big[-\frac{1}{2}(-\mathrm{x1}(t)\,c2+\mathrm{x1}(t)\,c1-2\,\mathrm{D}(x1)(t)\,\theta\,c2+2\,\mathrm{D}(x1)(t)\,\theta\,c1+2\,V0\,\mathrm{x2}(t+h)$$
$$+2\,V0\,\mathrm{D}(x2)(t+h)\,\theta)/((c2+c0-2\,c1)\,\theta)\Big]$$
$$\Big[\frac{1}{2}(-\mathrm{x1}(t)\,c1+\mathrm{x1}(t)\,c0-2\,\mathrm{D}(x1)(t)\,\theta\,c1+2\,\mathrm{D}(x1)(t)\,\theta\,c0+2\,V0\,\mathrm{x2}(t+h)$$
$$+2\,V0\,\mathrm{D}(x2)(t+h)\,\theta)/((c2+c0-2\,c1)\,\theta)\Big]$$

Note that the flat output $(x1 \quad x2)^T$ is well-defined if and only if $c1 \neq \frac{c0+c2}{2}$.

```
> u1ref3 := t -> evalm(P[3,1]):
> u2ref3 := t -> evalm(P[4,1]):
```

Using the flat output, the problem of motion planning is easily solved for the stirred tank. If we specify the desired trajectories $x1_{\mathrm{ref}}$, $x2_{\mathrm{ref}}$ for $x1$, $x2$, then substitution into P results in the corresponding open-loop inputs $u1_{\mathrm{ref}}$, $u2_{\mathrm{ref}}$ which realize the given trajectories. Let us give an example.

```
> x1ref3 := 0.1*exp(-alpha*t):  x2ref3 := 0.2*exp(-beta*t):
> alpha := 1/10:  beta := 1/7:
> plot([x1ref3,x2ref3], t=0..70, color=[red,green]);
```



Let us choose a time-delay of "length" $h = 5$. Then the corresponding open-loop inputs are:

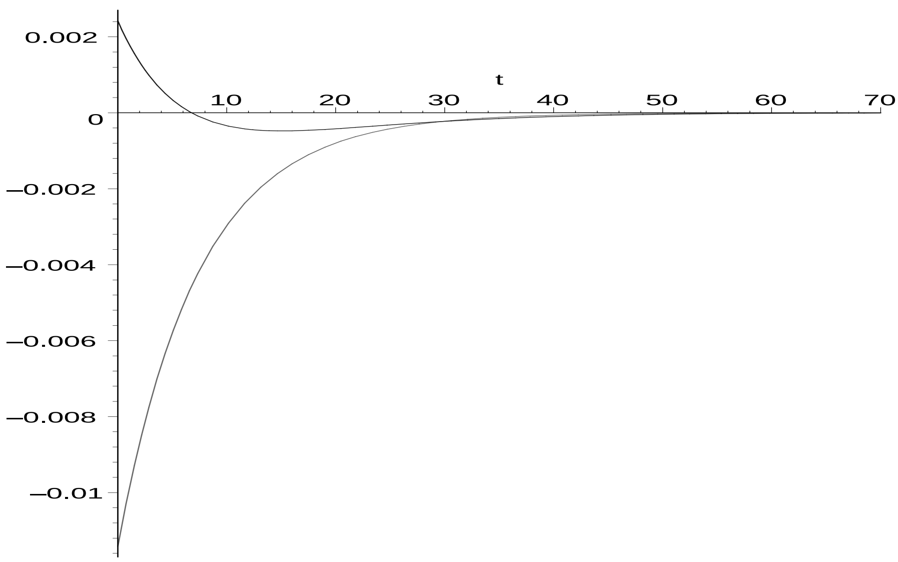
```
> conf := [c1=1,c2=2,c0=1.25,V0=1,theta=50]:
> u1ref5 := subs([h=5,op(conf)],
> ApplyMatrix(linalg[submatrix](T, 3..4, 1..2),
> [x1ref3,x2ref3], Alg3)[1,1]);
```

$$u1ref5 := -0.007200000000\, e^{(-0.1000000000\, t)} + 0.01965714285\, e^{(-0.1428571429\, t - 0.7142857145)}$$

```
> u2ref5 := subs([h=5,op(conf)],
> ApplyMatrix(linalg[submatrix](T, 3..4, 1..2),
> [x1ref3,x2ref3], Alg3)[2,1]);
```

$$u2ref5 := -0.001800000000\, e^{(-0.1000000000\, t)} - 0.01965714285\, e^{(-0.1428571429\, t - 0.7142857145)}$$

```
> plot([u1ref5,u2ref5], t=0..70, color=[blue,magenta]);
```



Let us choose a time-delay of "length" $h = 10$. Then the corresponding open-loop inputs are:

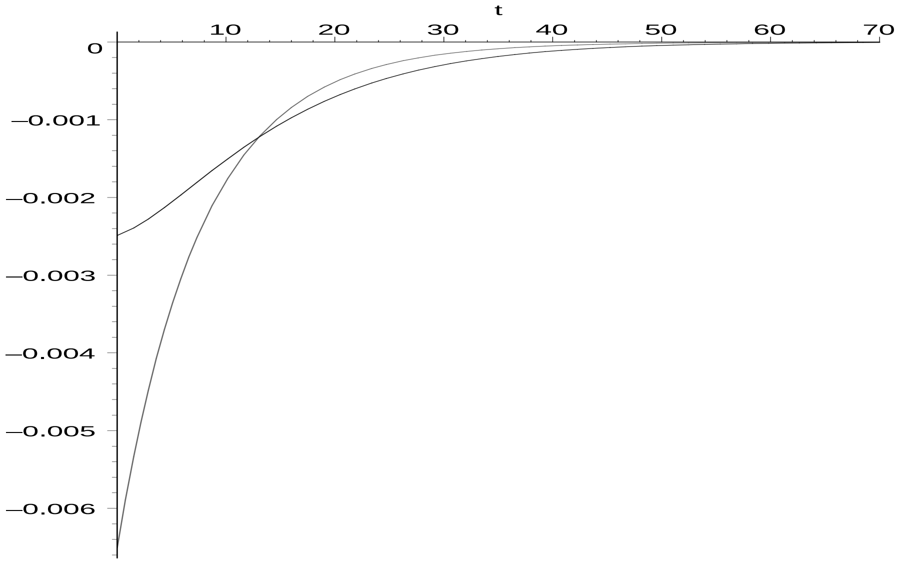
```
> u1ref10 := subs([h=10,op(conf)],
> ApplyMatrix(linalg[submatrix](T, 3..4, 1..2),
> [x1ref3,x2ref3], Alg3)[1,1]);
```

$$u1ref10 := -0.007200000000\, e^{(-0.1000000000\, t)} + 0.01965714285\, e^{(-0.1428571429\, t - 1.428571429)}$$

```
> u2ref10 := subs([h=10,op(conf)],
> ApplyMatrix(linalg[submatrix](T, 3..4, 1..2),
> [x1ref3,x2ref3], Alg3)[2,1]);
```

$$u2ref10 := -0.001800000000\, e^{(-0.1000000000\, t)} - 0.01965714285\, e^{(-0.1428571429\, t - 1.428571429)}$$

```
> plot([u1ref10,u2ref10], t=0..70, color=[blue,magenta]);
```



From the above computations we derived that the system was flat if $c1 \neq \frac{c0+c2}{2}$. Let us study the case $c1 = \frac{c0+c2}{2}$.

```
> R3mod := simplify(subs(c1=(c0+c2)/2, evalm(R3)));
```

$$R3mod := \begin{bmatrix} \dfrac{2\,\mathrm{D}\,\theta + 1}{2\,\theta} & 0 & -1 & -1 \\ 0 & \dfrac{1 + \mathrm{D}\,\theta}{\theta} & \dfrac{(-c2 + c0)\,\delta}{2\,V0} & \dfrac{(-c2 + c0)\,\delta}{2\,V0} \end{bmatrix}$$

We check whether the system is controllable for this particular configuration of the parameters. The $\mathcal{O}$-module which is associated with this linear system is denoted by $M3'$. First we compute the extension group $\mathrm{ext}^1_{\mathcal{O}}(M3'^{\top}, \mathcal{O})$ of the transposed module of $M3'$.

```
> Ext1mod := Exti(linalg[transpose](R3mod), Alg3, 1);
```

$$Ext1mod := \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2\,\mathrm{D}\,\theta + 1 & 0 & -2\,\theta & -2\,\theta \\ 0 & -2\,\theta\,V0\,\mathrm{D} - 2\,V0 & -(c0 - c2)\,\theta\,\delta & -(c0 - c2)\,\theta\,\delta \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} 4\,V0\,\theta^2\,\mathrm{D} + 4\,V0\,\theta & 0 \\ -(c0 - c2)\,\theta\,\delta - 2\,\theta^2\,(c0 - c2)\,\mathrm{D}\,\delta & 0 \\ 0 & 1 \\ 4\,V0\,\theta^2\,\mathrm{D}^2 + 6\,\theta\,V0\,\mathrm{D} + 2\,V0 & -1 \end{bmatrix} \right]$$

In a similar way as before, we conclude from the annihilating elements in the first matrix of *Ext1mod* that the torsion submodule $t(M3')$ of $M3'$ is trivial. Hence, the system is controllable and parametrizable. A parametrization of the behavior is given by *Ext1mod*[3] or by the following command:

```
> matrix([[x1(t)],[x2(t)],[u1(t)],[u2(t)]])=
> Parametrization(R3mod, Alg3);
```

$$\begin{bmatrix} \mathrm{x1}(t) \\ \mathrm{x2}(t) \\ \mathrm{u1}(t) \\ \mathrm{u2}(t) \end{bmatrix} = \begin{bmatrix} 4\,V0\,\theta^2\,\mathrm{D}(\xi_1)(t) + 4\,V0\,\theta\,\xi_1(t) \\ \theta\,\xi_1(t-h)\,c2 - \theta\,\xi_1(t-h)\,c0 + 2\,\theta^2\,\mathrm{D}(\xi_1)(t-h)\,c2 - 2\,\theta^2\,\mathrm{D}(\xi_1)(t-h)\,c0 \\ \xi_2(t) \\ 4\,V0\,\theta^2\,(\mathrm{D}^{(2)})(\xi_1)(t) + 6\,V0\,\theta\,\mathrm{D}(\xi_1)(t) + 2\,V0\,\xi_1(t) - \xi_2(t) \end{bmatrix}$$

Flatness of the system is equivalent to freeness of $M3'$. Let us first check whether $M3'$ is projective or not by computing the second extension group $\mathrm{ext}^2_{\mathcal{O}}(M3'^{\top}, \mathcal{O})$:

```
> Ext2mod := Exti(linalg[transpose](R3mod), Alg3, 2);
```

$$Ext2mod := \left[\begin{bmatrix} \delta & 0 \\ 1 + \mathrm{D}\,\theta & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathrm{SURJ}(2) \right]$$

From the annihilating elements in the first column of *Ext2mod*[1] we see that the first row of *Ext2mod*[2] defines a non-trivial generator of $\mathrm{ext}^2_{\mathcal{O}}(M3'^{\top}, \mathcal{O})$. Hence, $M3'$ is not projective and not free.

The system is not flat. This is coherent with the fact that the parametrization *Ext1mod*[3] does not induce an injective homomorphism:

```
> LeftInverse(Ext1mod[3], Alg3);
```

$$[\,]$$

However, we can find a "π-polynomial" π in $\mathbb{Q}(\theta, V0, c0, c1, c2)[\delta]$ such that

$$S^{-1}\,D \otimes M3' = \{\, m/a \mid m \in M3',\ a = \pi^j,\ j \in \mathbb{Z}_{\geq 0} \,\}$$

is a free $S^{-1}\,D$-module, where $S := \{\, \pi^j \mid j \in \mathbb{Z}_{\geq 0} \,\}$ is the multiplicatively closed subset of D generated by π. Hence, π encodes the obstructions towards the flatness of the system.

```
> PiPolynomial(R3mod, Alg3, [delta]);
```

$$[\delta]$$

Therefore, if we allow to invert the time-delay operator δ, then there exists an injective parametrization of the system. Let us show that *Ext1mod*[3] induces an injective homomorphism of $S^{-1}\,D$-modules:

```
> Smod := LocalLeftInverse(Ext1mod[3], [delta], Alg3);
```

$$Smod := \begin{bmatrix} \dfrac{1}{2\ V0\,\theta} & \dfrac{1}{\delta\,\theta\,(c0 - c2)} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Using the advance operator δ^{-1}, a flat output $(\xi 1 \quad \xi 2)^T$ of the behavior is defined by

```
> matrix([[xi[1](t)],[xi[2](t)]])=
> ApplyMatrix(Smod, [x1(t),x2(t),u1(t),u2(t)], Alg3);
```

$$\begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{2}\,\dfrac{\mathrm{x1}(t)\,c0 - \mathrm{x1}(t)\,c2 + 2\ V0\,\mathrm{x2}(t+h)}{V0\,\theta\,(c0 - c2)} \\ \mathrm{u1}(t) \end{bmatrix}$$

The flat output satisfies $(x1 \quad x2 \quad u1 \quad u2)^T = Ext1mod[3]\,(\xi 1 \quad \xi 2)^T$. It is well-defined if and only if $c0 \neq c2$.

Finally, if $c1 = \frac{c0+c2}{2}$ and $c0 = c2$, i.e. $c0 = c1 = c2$, then the system matrix is:

```
> subs([c2=c0,c1=c0], evalm(R3));
```

$$\begin{bmatrix} \mathrm{D} + \dfrac{1}{2\,\theta} & 0 & -1 & -1 \\ 0 & \mathrm{D} + \dfrac{1}{\theta} & 0 & 0 \end{bmatrix}$$

For this configuration of the parameters the system matrix involves no time-delay operator. This is exactly the case studied in Section 7.3, where it was shown that the system was not (completely) controllable; in particular, the system is not flat in this case.

Bibliography

[ABMP95] E. Aranda-Bricaire, C. H. Moog, and J.-B. Pomet. Infinitesimal Brunovský form for nonlinear systems, with applications to dynamic linearization. In *Geometry in Nonlinear Control and Differential Inclusions (Warsaw, 1993)*, volume 32 of *Banach Center Publications*, pages 19–33. Polish Academy of Sciences, Warsaw, 1995. 110

[AH05] J. Apel and R. Hemmecke. Detecting unnecessary reductions in an involutive basis computation. *J. Symbolic Computation*, 40(4–5):1131–1149, 2005. 36

[AL94] W. W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases*. American Mathematical Society, 1994. 25, 37

[ALLR82] P. Agrawal, C. Lee, H. C. Lim, and D. Ramkrishna. Theoretical Investigations of Dynamic Behavior of Isothermal Continuous Stirred Tank Biological Reactors. *Chemical Engineering Science*, 37(3):453–462, 1982. 106, 107

[AMR88] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*. Springer, second edition, 1988. 49, 52

[And] I. M. Anderson. The Variational Bicomplex. Preprint, `http://www.math.usu.edu/~fg_mp/Publications/VB/vb.pdf`. 113

[Aub77] J.-P. Aubin. *Applied Abstract Analysis*. John Wiley & Sons, 1977. 48, 49

[Bar01a] M. Barakat. *Functional Spaces. A Direct Approach*. PhD thesis, RWTH Aachen, 2001. 43, 113

[Bar01b] M. Barakat. Jets. A MAPLE-Package for Formal Differential Geometry. In V. G. Ganzha, E. W. Mayr, and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing CASC 2001*, pages 1–12. Springer, 2001. 43, 47, 60

[BCG$^+$03a] Y. A. Blinkov, C. F. Cid, V. P. Gerdt, W. Plesken, and D. Robertz. The MAPLE Package "Janet": I. Polynomial Systems. In *Proceedings of the 6th International Workshop on Computer Algebra in Scientific Computing, Passau (Germany)*, pages 31–40, 2003. `http://wwwb.math.rwth-aachen.de/Janet`. 36

[BCG$^+$03b] Y. A. Blinkov, C. F. Cid, V. P. Gerdt, W. Plesken, and D. Robertz. The MAPLE Package "Janet": II. Linear Partial Differential Equations. In *Proceedings of the 6th International Workshop on Computer Algebra in Scientific Computing, Passau (Germany)*, pages 41–54, 2003. `http://wwwb.math.rwth-aachen.de/Janet`. 36

[BLOP95] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Representation for the radical of a finitely generated differential ideal. In *Proceedings of ISSAC95*, pages 158–166, 1995. 59, 63

[Bou80] N. Bourbaki. *Elements de Mathematiques, Algèbre Ch. 10*. Masson, 1980. 76, 86

[BR] M. Barakat and D. Robertz. homalg: An abstract package for homological algebra. In preparation. 72

[BR06] M. Barakat and D. Robertz. Computing Invariants of Multidimensional Linear Systems on an Abstract Homological Level. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2006), Kyoto (Japan)*, pages 542–559, 2006. 72

[BY83] H. Blomberg and R. Ylinen. *Algebraic Theory for Multivariable Linear Systems*. Academic Press, 1983. 96

[Car14] E. Cartan. Sur l'équivalence absolue de certains systèmes d'équations différentielles et sur certaines familles de courbes. *Bulletin de la Société Mathématique de France*, 42:12–48, 1914. 138

[CE56] H. Cartan and S. Eilenberg. *Homological Algebra*. Princeton University Press, 1956. 72

[Chy98] F. Chyzak. *Fonctions holonomes en calcul formel*. PhD thesis, Ecole Polytechnique, France, 1998. 6, 28

[Coh85] P. M. Cohn. *Free Rings and their Relations*. Academic Press, second edition, 1985. 31

[Cou95] S. C. Coutinho. *A Primer of Algebraic D-modules*. London Mathematical Society Student Texts 33. Cambridge University Press, 1995. 91, 93

[CQR05] F. Chyzak, A. Quadrat, and D. Robertz. Effective algorithms for parametrizing linear control systems over Ore algebras. *Applicable Algebra in Engineering, Communication and Computing*, 16:319–376, 2005. 75, 97, 98, 116, 117, 140, 185

[CQR06a] F. Chyzak, A. Quadrat, and D. Robertz. OreModules *project*, 2003-2006. `http://wwwb.math.rwth-aachen.de/OreModules`. 116, 175

[CQR06b] F. Chyzak, A. Quadrat, and D. Robertz. OreModules: A symbolic package for the study of multidimensional linear systems. In J. Chiasson and J.-J. Loiseau, editors, *Applications of Time-Delay Systems*, 2006. To appear. 116

[CS98] F. Chyzak and B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *J. Symbolic Computation*, 26:187–227, 1998. 6, 28

[Die69] J. Dieudonné. *Foundations of Modern Analysis*. Academic Press, second edition, 1969. 48, 50

[DPR99] F. Dubois, N. Petit, and P. Rouchon. Motion Planning and Nonlinear Simulations for a Tank Containing a Fluid. In *Proceedings of the European Control Conference, Karlsruhe (Germany)*, 1999. 119

[Ehr70] L. Ehrenpreis. *Fourier Analysis in Several Complex Variables*. Wiley Interscience Publishers, 1970. 85

[Eva06] G. A. Evans. *Noncommutative Involutive Bases*. PhD thesis, University of Wales, Bangor, 2006. 6

[FG93] M. Fliess and S. T. Glad. An Algebraic Approach to Linear and Nonlinear Control. In H. L. Trentelman and J. C. Willems, editors, *Essays on Control: Perspectives in the Theory and its Applications*, pages 223–267. Birkhäuser, 1993. 97, 138

[FLMR95] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: introductory theory and examples. *Int. J. Control*, 61:1327–1361, 1995. 117, 138

[FM98] M. Fliess and H. Mounier. Controllability and observability of linear delay systems: an algebraic approach. *ESAIM COCV*, 3:301–314, 1998. 97, 116, 139

[FO98] S. Fröhler and U. Oberst. Continuous time-varying linear systems. *Systems & Control Letters*, 35:97–110, 1998. 86

[FPA03] C. Fleck, T. Paulus, and D. Abel. Eine flachheitsbasierte Randsteuerung für das Stefan-Problem. In *GMA Kongress 2003, VDI-Berichte 1756, ISBN 3-18-091756-3*, pages 311–319, 2003. 139

[FQ06] A. Fabiańska and A. Quadrat. Flat shift-invariant multidimensional linear systems are algebraically equivalent to controllable 1-D linear systems. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2006), Kyoto (Japan)*, pages 560–582, 2006. 140

[GB98a] V. P. Gerdt and Y. A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45:519–541, 1998. 5, 10, 36

[GB98b] V. P. Gerdt and Y. A. Blinkov. Minimal involutive bases. *Mathematics and Computers in Simulation*, 45:543–560, 1998. 5, 36

[GBM06] V. P. Gerdt, Y. A. Blinkov, and V. V. Mozzhilkin. Gröbner Bases and Generation of Difference Schemes for Partial Differential Equations. *Symmetry, Integrability and Geometry: Methods and Applications*, 2, 2006. 36

[Ger05] V. P. Gerdt. Involutive Algorithms for Computing Gröbner Bases. In S. Cojocaru, G. Pfister, and V. Ufnarovski, editors, *Computational Commutative and Non-Commutative Algebraic Geometry*, NATO Science Series, pages 199–225. IOS Press, 2005. 6

[Ger06] V. P. Gerdt. On Computation of Gröbner Bases for Linear Difference Systems. *Nuclear Instruments and Methods in Physics Research, A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559(1):211–214, 2006. 36

[GL01] H. Gluesing-Luerssen. *Linear Delay-Differential Systems with Commensurate Delays: An Algebraic Approach*. Springer, 2001. 96

[GR06] V. P. Gerdt and D. Robertz. A Maple Package for Computing Gröbner Bases for Linear Recurrence Relations. *Nuclear Instruments and Methods in Physics Research, A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559(1):215–219, 2006. 36

[GY05] V. P. Gerdt and D. A. Yanovich. Experimental Analysis of Involutive Criteria. In A. Dolzmann, A. Seidl, and T. Sturm, editors, *Algorithmic Algebra and Logic*, pages 105–109. BOD Norderstedt, 2005. 36

[Hil12] D. Hilbert. Über den Begriff der Klasse von Differentialgleichungen. *Math. Ann.*, 73:95–108, 1912. 138

[HS97] P. J. Hilton and U. Stammbach. *A Course in Homological Algebra*. Springer, 1997. 72

[HS01] A. Hillebrand and W. Schmale. Towards an Effective Version of a Theorem of Stafford. *J. Symbolic Computation*, 32:699–716, 2001. 116, 141, 146

[HSS02] M. Hausdorf, W. M. Seiler, and R. Steinwandt. Involutive Bases in the Weyl Algebra. *J. Symbolic Computation*, 34:181–198, 2002. 6

[Hub99] E. Hubert. Essential Components of an Algebraic Differential Equation. *J. Symbolic Computation*, 29(4-5):657–680, 1999. 63

[Hub00] E. Hubert. Factorisation free decomposition algorithms in differential algebra. *J. Symbolic Computation*, 29(4-5):641–662, 2000. 59, 63

[Jac85] N. Jacobson. *Basis Algebra I*. Freeman, second edition, 1985. 101

[Jan29] M. Janet. *Leçons sur les systèmes des équationes aux dérivées partielles*. Cahiers Scientifique IV. Gauthiers-Villars, Paris, 1929. 5, 8

[Jan71] M. Janet. P. Zervos et le problème de Monge. *Bull. Sci. Math.*, 95(2):15–26, 1971. 117, 138

[Kol73] E. Kolchin. *Differential Algebra and Algebraic Groups*. Academic Press, 1973. 58, 63

[KRW90] A. Kandri-Rody and V. Weispfenning. Non-commutative Gröbner Bases in Algebras of Solvable Type. *J. Symbolic Computation*, 9:1–26, 1990. 6

[KS72] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. John Wiley & Sons, 1972. 149, 150, 157, 161, 164, 170, 171, 175, 177, 182

[Lan93] S. Lang. *Algebra*. Addison-Wesley, third edition, 1993. 78

[Lev05] V. Levandovskyy. *Non-commutative Computer Algebra for polynomial algebras: Gröbner bases, applications and implementation*. PhD thesis, Universität Kaiserslautern, Germany, 2005. 6

[Ley04] A. Leykin. Algorithmic proofs of two theorems of Stafford. *J. Symbolic Computation*, 38:1535–1550, 2004. 116, 141, 146

[Mal63] B. Malgrange. Systèmes différentiels à coefficients constants. *Séminaire Bourbaki*, 246:1–11, 1963. 71, 85

[May97] E. W. Mayr. Some Complexity Results for Polynomial Ideals. *J. Complexity*, 13:303–325, 1997. 37

[Mér80] C. Méray. Démonstration générale de l'existence des intégrales des équations aux dérivées partielles. *Journal de mathématiques pures et appliquées, 3e série*, tome VI:235–265, 1880. 5

[Mou95] H. Mounier. *Propriétés des systèmes linéaires à retards: aspects théoriques et pratiques*. PhD thesis, University of Orsay, France, 1995. 97, 116, 139, 185

[MR00] J. C. McConnell and J. C. Robson. *Noncommutative Noetherian Rings*. American Mathematical Society, 2000. 28, 31, 74, 75, 98, 142

[MRFR98] H. Mounier, J. Rudolph, M. Fliess, and P. Rouchon. Tracking control of a vibrating string with an interior mass viewed as delay system. *ESAIM COCV*, 3:315–321, 1998. 139

[Obe90] U. Oberst. Multidimensional constant linear systems. *Acta Appl. Math.*, 20:1–175, 1990. 83, 85, 86

[Olv93] P. J. Olver. *Applications of Lie Groups to Differential Equations*. Graduate Texts in Mathematics 107. Springer, second edition, 1993. 44, 48, 52, 114

[OM02] J. Oldenburg and W. Marquardt. Flatness and higher order differential model representations in dynamic optimization. *Computers and Chemical Engineering*, 26:385–400, 2002. 139

[Ore33] O. Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34:480–508, 1933. 31

[Pal70] V. P. Palamodov. *Linear Differential Operators with Constant Coefficients*. Springer, 1970. 85

[Pom94] J.-F. Pommaret. *Partial Differential Equations and Group Theory*. Kluwer Academic Publishers, 1994. 5, 43, 44, 53, 110

[Pom95] J.-F. Pommaret. Dualité différentielle et applications. *C.R. Acad. Sci. Paris*, Série I 320:1225–1230, 1995. 97, 117

[Pom01] J.-F. Pommaret. *Partial Differential Control Theory*. Kluwer Academic Publishers, 2001. 96, 97, 104, 117, 123, 132

[PQ98] J.-F. Pommaret and A. Quadrat. Generalized Bezout Identity. *Appl. Algebra Engrg. Comm. Comput.*, 9:91–116, 1998. 96

[PQ99a] J.-F. Pommaret and A. Quadrat. Algebraic analysis of linear multidimensional control systems. *IMA J. Control and Optimization*, 16:275–297, 1999. 96, 117

[PQ99b] J.-F. Pommaret and A. Quadrat. Localization and parametrization of linear multidimensional control systems. *Systems & Control Letters*, 37:247–260, 1999. 96, 97, 117, 139, 158

[PQ03] J.-F. Pommaret and A. Quadrat. A functorial approach to the behaviour of multidimensional control systems. *Int. J. Appl. Math. Comput. Sci.*, 13:7–13, 2003. 98

[PQ04] J.-F. Pommaret and A. Quadrat. A differential operator approach to multidimensional optimal control. *Int. J. Control*, 77:821–836, 2004. 121, 131, 149, 172

[PR05] W. Plesken and D. Robertz. Janet's approach to presentations and resolutions for polynomials and linear pdes. *Archiv der Mathematik*, 84(1):22–37, 2005. 6, 7

[PR06] W. Plesken and D. Robertz. Representations, commutative algebra, and Hurwitz groups. *Journal of Algebra*, 300:223–247, 2006. 38, 39

[PS98] H. K. Pillai and S. Shankar. A behavioral approach to control of distributed systems. *SIAM J. Control Optim.*, 37:388–408, 1998. 97

[PW98] J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory. A Behavioral Approach*. TAM 26. Springer, 1998. 96, 117

[QR05a] A. Quadrat and D. Robertz. On the blowing-up of stably free behaviours. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference, Seville*, pages 1541–1546, 2005. 141

[QR05b] A. Quadrat and D. Robertz. Parametrizing all solutions of uncontrollable multidimensional linear systems. In *Proceedings of the 16th IFAC World Congress, Prague*, 2005. 117, 125, 134

[QR06a] A. Quadrat and D. Robertz. Constructive computation of flat outputs of a class of multidimensional linear systems with variable coefficients. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2006), Kyoto (Japan)*, pages 583–595, 2006. 140, 141, 142, 143, 146

[QR06b] A. Quadrat and D. Robertz. On the Monge problem and multidimensional optimal control. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2006), Kyoto (Japan)*, pages 596–605, 2006. 121, 131

[Qua99] A. Quadrat. *Analyse algébrique des systèmes de contrôle linéaires multidimensionnels*. PhD thesis, Ecole Nationale des Ponts et Chaussées, France, 1999. 96, 97, 172

[Reh02] H. P. Rehm. Differential-Algebra. Lecture Notes, Univ. of Karlsruhe, Germany, WS 2001/2002. 31

[Riq10] C. Riquier. *Les systèmes d'équations aux dérivées partielles*. Gauthiers-Villars, Paris, 1910. 5

[Rit66] J. F. Ritt. *Differential Algebra*. Dover, 1966. 59, 63

[Rot79] J. J. Rotman. *An Introduction to Homological Algebra*. Academic Press, 1979. 72, 74, 76, 77, 85, 86, 126

[Rot97] R. Rothfuß. *Anwendung der flachheitsbasierten Analyse und Regelung nichtlinearer Mehrgrößensysteme*. Meß-, Steuerungs- und Regelungstechnik, Nr. 664. Fortschrittberichte VDI, 1997. 139

[Son98] E. D. Sontag. *Mathematical Control Theory. Deterministic Finite Dimensional Systems*. TAM 6. Springer, second edition, 1998. 96, 155, 170

[Sta78] J. T. Stafford. Module structure of Weyl algebras. *J. London Math. Soc.*, 18:429–442, 1978. 116, 142, 145

[Tho37] J. Thomas. *Differential Systems*. American Mathematical Society, 1937. 10

[vzGG03] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, second edition, 2003. 37

[Wil91] J. C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, 36:259–294, 1991. 96

[Woo00] J. Wood. Modules and behaviours in nD systems theory. *Multidimensional Systems and Signal Processing*, 11:11–48, 2000. 96, 97

[Zer32] P. Zervos. Le problème de Monge. *Mémorial des Sciences Mathématiques, fasc. LIII*, 1932. 117, 138

[Zer00] E. Zerz. *Topics in Multidimensional Linear Systems Theory*. Lecture Notes in Control and Information Sciences 256. Springer, 2000. 96, 117

[Zer06] E. Zerz. An algebraic analysis approach to linear time-varying systems. *IMA J. Mathematical Control & Information*, 23:113–126, 2006. 86

[ZL01] E. Zerz and V. Lomadze. A Constructive Solution to Interconnection and Decomposition Problems with Multidimensional Behaviors. *SIAM J. Control Optim.*, 40:1072–1086, 2001. 125

Symbol Table

Index

Curriculum Vitae

Daniel Robertz

25.11.1977	Geburt in Aachen
1984–1988	Gemeinschaftsgrundschule Driescher Hof, Aachen
1988–1997	Inda-Gymnasium, Aachen; Abschluss: Abitur
1997–1998	Zivildienst in Aachen
Okt. 1998	Beginn des Mathematik-Studiums mit Nebenfach Informatik an der RWTH Aachen
Okt. 2000 – Jan. 2003	Studentische Hilfskraft am Lehrstuhl B für Mathematik, RWTH Aachen
Dez. 2000	Teilnahme am Sun-Softwarepreis (organisiert durch die Fachgruppe Informatik, RWTH Aachen): Programmierung eines autonomen Agenten für die Robocup-Simulationsliga; Erreichen des 1. Platzes (zusammen mit zwei Kommilitonen)
Jan. 2003	Diplom in Mathematik mit dem Gesamturteil “mit Auszeichnung”
Feb. 2003 – Sep. 2003	Stipendiat des Graduiertenkollegs “Hierarchie und Symmetrie in mathematischen Modellen” an der RWTH Aachen
Feb. 2003 – Apr. 2005	6- bis 10-wöchige Besuche am Forschungsinstitut INRIA, Sophia Antipolis, Frankreich
Jun. 2003 – Sep. 2003	Wissenschaftliche Hilfskraft am Lehrstuhl B für Mathematik, RWTH Aachen
seit Okt. 2003	Wissenschaftlicher Mitarbeiter am Lehrstuhl B für Mathematik, RWTH Aachen, und Kollegiat des Graduiertenkollegs “Hierarchie und Symmetrie in mathematischen Modellen” an der RWTH Aachen