Computing Invariants of Multidimensional Linear Systems on an Abstract Homological Level

Mohamed Barakat and Daniel Robertz

***Abstract*— Methods from homological algebra [16] play a more and more important role in the study of multidimensional linear systems [15], [14], [6]. The use of modules allows an algebraic treatment of linear systems which is independent of their presentations by systems of equations. The type of linear system (ordinary/partial differential equations, time-delay systems, discrete systems...) is encoded in the (non-commutative) ring of (differential, shift, ...) operators over which the modules are defined. In this framework, homological algebra gives very general information about the structural properties of linear systems.**

Homological algebra is a natural extension of the theory of modules over rings. The category of modules and their homomorphisms is replaced by the category of chain complexes and their chain maps. A module is represented by any of its resolutions. The module is then recovered as the only non-trivial homology of the resolution. The notions of derived functors and their homologies, connecting homomorphism and the resulting long exact homology sequences play a central role in homological algebra.

The MAPLE-package homalg [1], [2] provides a way to deal with these powerful notions. The package is abstract in the sense that it is independent of any specific ring arithmetic. If one specifies a ring in which one can solve the ideal membership problem and compute syzygies, the above homological algebra constructions over that ring become accessible using homalg.

In this paper we introduce the package homalg and present several applications of homalg to the study of multi-dimensional linear systems using available MAPLE-packages which provide the ring arithmetics, e.g. OREMODULES [4], [5] and JANET [3], [13].

***Keywords*— Homological algebra, multidimensional linear systems, SMITH normal form, JACOBSON normal form, extension modules, computer algebra.**

I. INTRODUCTION

In linear control theory it became more and more evident that properties of the system are encoded by its intrinsic nature as a module over a certain ring of operators, rather than its specific realization as system of equations. The theory that deals with these intrinsic properties is the general theory of modules over rings and the homological algebra of the category of such modules. As the name of this package suggests, our intention has been to make as much as possible of the basic homological machinery available in a computer algebra system without the need to specify the ring of operators from the beginning.

M. Barakat and D. Robertz, RWTH – Aachen, Templergraben 64, 52056 Aachen, Germany, mohamed.barakat@rwth-aachen.de, daniel@momo.math.rwth-aachen.de.

II. THE PHILOSOPHY OF THE PACKAGE

The basic objects of homalg are finitely presented left modules over rings in which the ideal membership problem is algorithmically solvable and syzygies are effectively computable. We call such rings *computable*. homalg implements the homological constructions for modules over such rings, whereas the ring arithmetic has to be provided by a ring-specific package. The following ring-specific packages have successfully been used with homalg: INVOLUTIVE and JANET [3], OREMODULES [4]. PIR is one more tiny package, or rather a pseudo-package, that makes MAPLE's builtin facilities for dealing with integers and some other principal ideal rings available to homalg. The package PIR uses the SMITH normal form to provide a standard form for the presentation of a module.

The central objects in homalg are functors. Functors map on the one hand objects of a source category to objects of a target category, and on the other hand morphisms between two objects in the source category to morphisms between their images in the target category in a compatible way. The two most important functors are the Hom-functor and the tensor product functor $\otimes$ and their derived functors, the definition of which will be reproduced below.

A major effort in the implementation was to find the suitable scheme for realizing the functor part on objects in order to have a unified way in extracting the part of the functor on morphisms. Composition and derivation of functors in homalg rely exclusively on this and define again functors. I.e. extracting the morphism part of composed or derived functors is done in the same unified way as for all functors. Hence, using the two basic operations of composing and deriving functors, the user can without effort add new functors to those already existing in homalg.

Given a (covariant) functor F the i-th left derivation of F is as usual denoted by L_iF. A short exact sequence $0 \to M' \to M \to M'' \to 0$ of modules then gives rise to a long exact sequence connecting $L_iF(M') \to L_iF(M) \to L_iF(M'')$ and $L_{i+1}F(M') \to L_{i+1}F(M) \to L_{i+1}F(M'')$ for all $i \geq 0$. These so-called connecting homomorphisms are implemented in homalg.

Some natural transformations between functors are also implemented in homalg. The most prominent are the embedding of a kernel in the source of a map and the natural epimorphism from the target of a map onto its cokernel.

In homalg one finds procedures to compute homologies of complexes (especially to test exactness of complexes), to check commutativity of diagrams, to check surjectivity or injectivity of maps, etc.

One major restriction in `homalg` is that one cannot change the base ring. All functors are hence functors where the source and target category are defined over the same ring.

III. Finitely Presented Modules

`homalg` can only deal with finitely presented modules. A finitely presented module M over a ring D is a quotient of a free module of finite rank $D^{1\times l_0}$ by a finitely generated submodule $D^{1\times l_1}A = \mathrm{im}(.A)$, where $A \in D^{l_1\times l_0}$:

$$M = D^{1\times l_0}/D^{1\times l_1}A = \mathrm{coker}(.A).$$

As usual a presentation is given by generators and relations. A presentation of a module in `homalg` is a list containing as first entry the list of generators and as second entry the list of relations. The third entry is a string delimiter to optically indicate the end of the presentation. This string, unless changed by the user, defaults to `"Presentation"`. The remaining entries provide extra information about the presented module, e.g. its HILBERT series. This extra information can only be provided by the ring-specific package.

In the list of generators the concrete generators are numbered by abstract generators being the l_0 standard basis vectors of the underlying free module $D^{1\times l_0}$. The list of relations simply contains the rows of the matrix A. An example is given in Fig. III.

IV. Functors

Here we define the basic functors implemented in `homalg`. We restrict ourselves to describe only those functors with the source category also being the category of left D-modules. Nevertheless functors like the kernel functor $\ker$, the cokernel functor coker, the pullback functor and the defect of homomorphisms functor are implemented. The objects of their source categories are not merely modules but themselves morphisms between modules. In `homalg` the object and morphism part of a functor are two different procedures. If the object part has the name `F` then the morphism part is `FMap`.

Let D be a computable ring, as defined above.

A. The functor T

Over an ORE-domain D the set of all torsion elements of a left D-module M forms a submodule TM called the torsion submodule of M. Taking the torsion submodule is functorial, i.e. every D-module homomorphism $M \xrightarrow{\alpha} N$ induces by restriction again a homomorphism $TM \xrightarrow{T\alpha := \alpha|_{TM}} TN$. More precisely, T is a covariant functor from the category of left D-modules to itself.

B. The Hom-functor

For two left D-modules M and L denote by $\mathrm{Hom}(M, L)$ the abelian group of all D-module homomorphisms from M to N. For a D-module homomorphism $M \xrightarrow{\alpha} N$ let $\mathrm{Hom}(\alpha, L) : \mathrm{Hom}(N, L) \to \mathrm{Hom}(M, L) : \psi \mapsto \psi \circ \alpha$. Thus, $\mathrm{Hom}(-, L)$ is a contravariant functor from the category of D-modules to the category of abelian groups.

For this functor to comply with the above mentioned restriction certain properties of the ring D are required. Either D is commutative, or $L = D$, in which case the ring D should come with a fixed involution, i.e. self-inverse anti-automorphism $\theta : D \to D$. θ allows one to transform a right module structure to a left module structure again. If this is provided, then $\mathrm{Hom}(-, L)$ is a contravariant functor from the category of left D-modules to itself.

C. The tensor product functor $\otimes$

For a left resp. right D-module M resp. L denote by $M \otimes L$ the tensor product over D of M and L, which is an abelian group. For a D-module homomorphism $M \xrightarrow{\alpha} N$ let $\alpha \otimes L : M \otimes L \to N \otimes L : \ \varphi \mapsto \varphi \otimes \mathrm{Id}_L$. Thus $- \otimes L$ is a covariant functor from the category of D-modules to the category of abelian groups.

Again, for this functor to comply with the above mentioned restriction we always assume D to be commutative (note that $- \otimes D$ is equivalent to the identity functor). If this is provided, then $- \otimes L$ is a covariant functor from the category of D-modules to itself.

D. Derivations

We define the left (resp. right) derived functor of a covariant (resp. contravariant) functor F using projective resolutions: For a D-module M compute a projective resolution P

$$\underbrace{\cdots \to P_{i+1} \to P_i \to P_{i-1} \to \cdots \to P_1 \to P_0}_{=:P} \to M \to 0$$

of M. Define for $i \geq 0$ the left (resp. right) derived functor L_iF (resp. R_iF) of the covariant (resp. contravariant) functor F by taking the homology (resp. cohomology) of the complex (resp. cocomplex) $F(P)$ at the i-th position [10].

The most prominent left derived functor of a covariant functor is $\mathrm{Tor}_i(-, L)$ $(i \geq 0)$. Since $- \otimes L$ is right exact, the two functors $- \otimes L$ and $\mathrm{Tor}_0(-, L)$ are equivalent.

The most prominent right derived functor of a contravariant functor is $\mathrm{Ext}^i(-, L)$ $(i \geq 0)$. Since $\mathrm{Hom}(-, L)$ is left exact, the two functors $\mathrm{Hom}(-, L)$ and $\mathrm{Ext}^0(-, L)$ are equivalent. The Ext-functor has up to our knowledge simply more applications in systems theory than the Tor-functor.

Since we cannot compute injective resolutions, we are not able to implement right (resp. left) derived functors of a covariant (resp. contravariant) functor.

From the point of view of derived categories, a module M should be replaced by any of its resolutions, which is a complex, say P. All the resolutions of the module

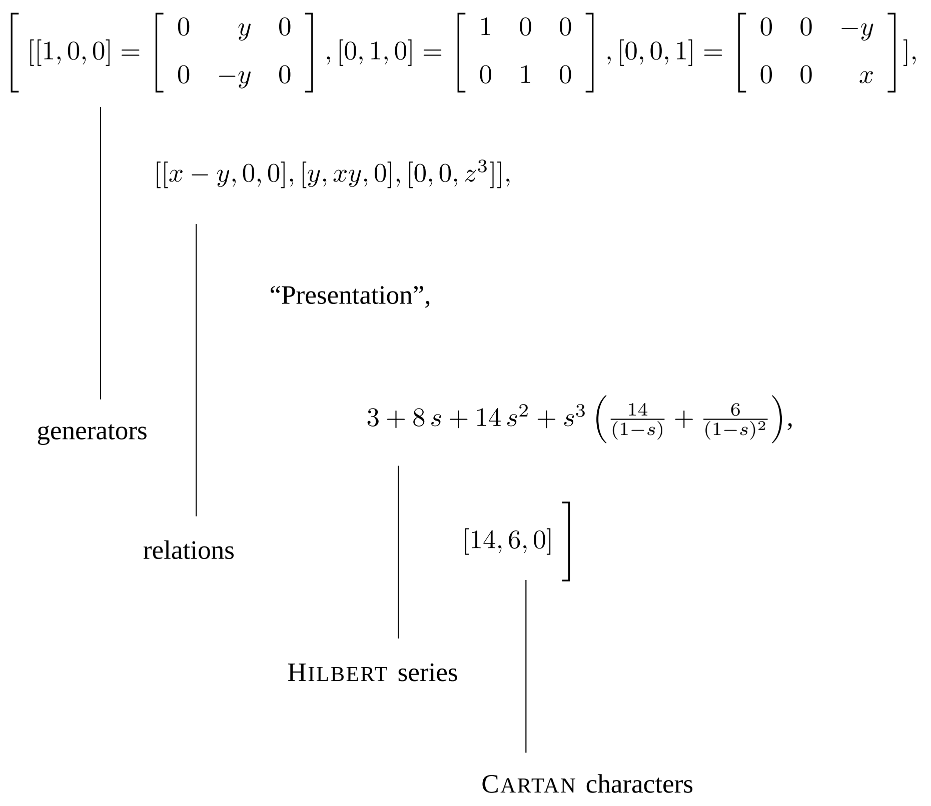
$[[1,0,0] = \begin{bmatrix} 0 & y & 0 \\ 0 & -y & 0 \end{bmatrix}, [0,1,0] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, [0,0,1] = \begin{bmatrix} 0 & 0 & -y \\ 0 & 0 & x \end{bmatrix}],$
$[[x-y,0,0],[y,xy,0],[0,0,z^3]],$
“Presentation”,
generators
$3 + 8\,s + 14\,s^2 + s^3 \left(\frac{14}{(1-s)} + \frac{6}{(1-s)^2}\right),$
relations
$[14,6,0]$
HILBERT series
CARTAN characters

Fig. 1. A module of homomorphisms between two modules over $D = \mathbb{Q}[x,y,z]$ with INVOLUTIVE

are homotopy equivalent. The complex P is exact in all degrees except for degree 0 and the module M is recovered as the only non-trivial homology of P at degree 0. In general two complexes are identified in the derived category if there exists a chain of quasi-isomorphisms, i.e. chain maps inducing isomorphism on homology, connecting the one complex with the other. Homotopy equivalences are special cases of quasi-isomorphisms. So we obtain the derived category by inverting quasi-isomorphisms. The connecting homomorphisms lead to the so-called exact triangles in the derived category of the category of finitely presented D-modules, which is simply the way to look at long exact sequences in the realm of triangulated categories.

V. AN EXAMPLE OVER THE GAUSSIAN INTEGERS

Here we take $D = \mathbb{Z}[\sqrt{-1}]$, which is a EUCLIDEAN domain, which is not a field, and hence has global dimension 1. In the following example M and N will be finitely generated D-modules. For M we consider

$$M \xrightarrow{\varepsilon} M^{**},$$

where $M^{**} := \operatorname{Hom}(\operatorname{Hom}(M,D),D)$ and ε is the evaluation map. We also consider the short exact sequence

$$0 \to TM \xrightarrow{\iota} M \xrightarrow{\nu} M/TM \to 0,$$

where ι is the embedding and ν is the natural epimorphism. The last sequence induces via the contravariant functor $\operatorname{Hom}(-,N)$ the sequence

$$0 \to \operatorname{Hom}(M/TM,N) \xrightarrow{\operatorname{Hom}(\nu,N)} \operatorname{Hom}(M,N)$$

$$\xrightarrow{\eta\ :=\ \operatorname{Hom}(\iota,N)} \operatorname{Hom}(TM,N) \to 0.$$

This sequence is again exact, since M/TM over the principal ideal domain D is free and $\operatorname{Ext}^1(M/TM,N)$ vanishes. We start with the diagram in Fig. V, where we only indicate the arrows we need. The middle square in the bottom row is specified in Fig. V. We assume that $a,b,c,d \in D$ satisfy $ab = c$ for the square to be commutative. $A' \cong \operatorname{Hom}(M/TM,N) \oplus TM$ resp. B' is defined as the kernel of α_2 resp. β_2, and τ is the map induced by ψ between the kernels. The two middle columns $A' \xrightarrow{\alpha_1} A \xrightarrow{\alpha_2} A''$ and $B' \xrightarrow{\beta_1} B \xrightarrow{\beta_2} B''$ regarded as chain complexes and (τ,ψ,ϕ) as a chain map induce a kernel sequence $K' \xrightarrow{\kappa_1} K \xrightarrow{\kappa_2} K''$ and a cokernel sequence $C' \xrightarrow{\omega_1} C \xrightarrow{\omega_2} C''$. Since, as seen above, α_2 is surjective and β_1 is injective by definition, there exists a connecting homomorphism δ connecting the kernel and the cokernel sequence to a long exact sequence:

$$K' \xrightarrow{\kappa_1} K \xrightarrow{\kappa_2} K'' \xrightarrow{\delta} C' \xrightarrow{\omega_1} C \xrightarrow{\omega_2} C''.$$

$$\begin{tikzcd}
 & & 0 \arrow[d] & & & \\
0 & C' \arrow[l] \arrow[d, "\omega_1"] & B' \arrow[l] \arrow[d, "\beta_1"] & A' \arrow[l, "\tau"'] \arrow[d, "\alpha_1"] & K' \arrow[l] \arrow[d, "\kappa_1"] & 0 \arrow[l] \\
0 & C \arrow[l] \arrow[d, "\omega_2"] & B \arrow[l] \arrow[d, "\beta_2"] & A \arrow[l, "\psi"'] \arrow[d, "\alpha_2"] & K \arrow[l] \arrow[d, "\kappa_2"] & 0 \arrow[l] \\
0 & C'' \arrow[l] & B'' \arrow[l] & A'' \arrow[l, "\phi"'] \arrow[d] & K'' \arrow[l] & 0 \arrow[l] \\
 & & & 0 & &
\end{tikzcd}$$

Fig. 2. Diagram for the example in Section V

$$\begin{tikzcd}
B \arrow[d, "\beta_2"] & A \arrow[l, "\psi"'] \arrow[d, "\alpha_2"] & & \mathrm{Hom}(TM,N) \oplus M^{**} \arrow[d, "b\,Id \oplus 0"] & \mathrm{Hom}(M,N) \oplus M \arrow[l, "a\,\eta \oplus d\,\varepsilon"'] \arrow[d, "\eta \oplus \nu"] \\
B'' & A'' \arrow[l, "\phi"'] & = & \mathrm{Hom}(TM,N) \oplus 0 & \mathrm{Hom}(TM,N) \oplus M/TM \arrow[l, "c\,Id \oplus 0"']
\end{tikzcd}$$

Fig. 3. Middle square in the bottom row of the diagram in Fig. V

```
> restart;
```

The package PIR enables one to work over several MAPLE-builtin principal ideal rings:

```
> with(homalg): with(PIR):
```

```
> RPP:='PIR/homalg';
```

$$RPP := PIR/homalg$$

Since we won't change the base ring during the computation we fix it once and for all:

```
> 'homalg/default':=RPP;
```

$$homalg/default := PIR/homalg$$

Specify $D = \mathbb{Z}[\sqrt{-1}]$, the ring of GAUSSIAN integers:

```
> var:=[I];
```

$$var := [I]$$

```
> Pvar(var);
```

$$[\text{“Z[I]”}]$$

Define the four variables with $c = a\,b$:

```
> a:=1+I; b:=5; c:=a*b; d:=2*(1+I);
```

$$a := 1 + I$$
$$b := 5$$
$$c := 5 + 5\,I$$
$$d := 2 + 2\,I$$

Define the D-module M:

```
> M:=Cokernel([[1,2,4,6],[6*(1+I)*1,6*(1+I)*3,6*(1+I)*4,6*(1+I)*5]],var);
```

$$M := [[[1, 0, 0] = [0, 1, 0, -1], [0, 1, 0] = [0, 0, 1, 0], [0, 0, 1] = [0, 0, 0, 1]], [[6 + 6\,I, 0, 0]], \text{“Presentation”}, [6 + 6\,I, 0, 0], 2]$$

The torsion submodule TM:

```
> TM:=TorsionSubmodule(M,var);
```

$$TM := [[1 = [0, 1, 0, -1]], [6 + 6\,I], \text{“Presentation”}, [6 + 6\,I], 0]$$

The embedding map ι:

```
> iota:=TorsionSubmoduleEmb(M,var);
```

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

The torsion free part $FM := M/TM$:

```
> FM:=Cokernel(iota,M,var);
```

$$FM := [[[1, 0] = [0, 0, 1, 0], [0, 1] = [0, 0, 0, 1]], [[0, 0]], \text{“Presentation”}, [0, 0], 2]$$

The natural epimorphism $M \to M/TM$:

```
>   nu:=CokernelEpi(iota,M,var);
```

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

The double dual M^{**}:

```
>   HHM:=HomHom_R(M,var);
```

$$HHM := [[[1,\ 0] = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ [0,\ 1] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}],\ [[0,\ 0]],\ \text{“Presentation”},\ [0,\ 0],\ 2]$$

The evaluation map $M \to M^{**}$:

```
>   epsilon:=NatTrIdToHomHom_R(M,var);
```

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

The D-module N:

```
>   N:=Cokernel([[1,2,4,0],[2*(1-I)*1,2*(1-I)*3,2*(1-I)*4,0],[0,0,0,2]],var);
```

$$N := [[[1,\ 0,\ 0] = [0,\ 0,\ 0,\ 1],\ [0,\ 1,\ 0] = [0,\ -1,\ 0,\ 1],\ [0,\ 0,\ 1] = [0,\ 0,\ 1,\ 0]],$$
$$[[2,\ 0,\ 0],\ [0,\ 2+2\,I,\ 0]],\ \text{“Presentation”}\ ,\ [2,\ 2+2\,I,\ 0],\ 1]$$

The module of homomorphisms $\mathrm{Hom}(M, N)$:

```
>   HMN:=Hom(M,N,var);
```

$$HMN := \Bigg[\Bigg[[1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},\ [0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0] = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$
$$[0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0] = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},\ [0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0] = \begin{bmatrix} -1 & 0 & 0 \\ 3 & 0 & 0 \\ -3 & 1 & 0 \end{bmatrix},$$
$$[0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0] = \begin{bmatrix} -1 & 0 & 0 \\ 3 & 1 & 0 \\ -4 & 1 & 0 \end{bmatrix},\ [0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 3 & 0 \\ -5 & 1 & 0 \end{bmatrix},$$
$$[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},\ [0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\Bigg],$$
$$[[2,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0],\ [0,\ 2,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0],\ [0,\ 0,\ 2,\ 0,\ 0,\ 0,\ 0,\ 0],$$
$$[0,\ 0,\ 0,\ 2+2\,I,\ 0,\ 0,\ 0,\ 0],\ [0,\ 0,\ 0,\ 0,\ 2+2\,I,\ 0,\ 0,\ 0],\ [0,\ 0,\ 0,\ 0,\ 0,\ 2+2\,I,\ 0,\ 0]],$$
$$\text{“Presentation”},\ [2,\ 2,\ 2,\ 2+2\,I,\ 2+2\,I,\ 2+2\,I,\ 0,\ 0],\ 2\Bigg]$$

The module of homomorphisms $\mathrm{Hom}(TM, N)$:

```
>   HTMN:=Hom(TM,N,var);
```

$$HTMN := [[[1,\ 0] = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},\ [0,\ 1] = \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}],\ [[2,\ 0],\ [0,\ 2+2\,I]],\ \text{“Presentation”},\ [2,\ 2+2\,I],\ 0]$$

The identity map of $\mathrm{Hom}(TM, N)$:

```
>   Id:=IdentityMap(HTMN,var);
```

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

The induced map $\eta := \mathrm{Hom}(\iota, N)$:

```
>  eta:=HomMap(TM,iota,M,N,var);
```

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

The zero module:

```
>  Z:=ZeroModule(var);
```

$$Z := [[1 = 0],\ [1],\ \text{“Presentation”},\ [1],\ 0]$$

The zero map from M/TM to the zero module:

```
>  zeta:=ZeroMap(FM,Z,var);
```

$$\zeta := \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The zero map from M^{**} to the zero module:

```
>  chi:=ZeroMap(HHM,Z,var);
```

$$\chi := \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

A is the direct sum of $\mathrm{Hom}(M, N)$ and M:

```
>  A:=DirectSum(HMN,M,var);
```

$$\begin{aligned}
&A := [[[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],\\
&[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0] = [0, -1, 1, 0, 0, 0, 0, 0, 0, 0, 0],\\
&[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0] = [1, -2, 1, 0, 0, 0, 0, 0, 0, 0, 0],\\
&[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0] = [-1, 3, -3, 0, 0, 1, 0, 0, 0, 0, 0],\\
&[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] = [0, 0, 0, 0, -1, 1, 0, 0, 0, 0, 0],\\
&[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0] = [0, 0, 0, 1, -2, 1, 0, 0, 0, 0, 0],\\
&[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0] = [0, 0, 0, -1, 3, -3, 0, 0, 1, 0, 0],\\
&[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0] = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],\\
&[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0] = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],\\
&[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],\\
&[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]], [\\
&[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],\ [0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0],\\
&[0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0],\ [0, 0, 0, 2 + 2\,I, 0, 0, 0, 0, 0, 0, 0],\\
&[0, 0, 0, 0, 2 + 2\,I, 0, 0, 0, 0, 0, 0],\ [0, 0, 0, 0, 0, 2 + 2\,I, 0, 0, 0, 0, 0],\\
&[0, 0, 0, 0, 0, 0, 6 + 6\,I, 0, 0, 0, 0]],\ \text{“Presentation”},\\
&[2, 2, 2, 2 + 2\,I, 2 + 2\,I, 2 + 2\,I, 6 + 6\,I, 0, 0, 0, 0],\ 4]
\end{aligned}$$

A'' is the direct sum of $\mathrm{Hom}(TM, N)$ and M/TM:

```
>  _A:=DirectSum(HTMN,FM,var);
```

$$\begin{aligned}
&\_A := [[[1, 0, 0, 0] = [1, 0, 0, 0],\ [0, 1, 0, 0] = [-1, 1, 0, 0],\ [0, 0, 1, 0] = [0, 0, 1, 0],\\
&[0, 0, 0, 1] = [0, 0, 0, 1]],\ [[2, 0, 0, 0],\ [0, 2 + 2\,I, 0, 0]],\ \text{“Presentation”},\ [2, 2 + 2\,I, 0, 0],\ 2]
\end{aligned}$$

α_2 is the direct sum of the maps η and ν:

```
>   alpha2:=DirectSumMap(HMN,M,eta,nu,HTMN,FM,var);
```

$$\alpha 2 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A' is the kernel of α_2:

```
>   A_:=Kernel(A,alpha2,_A,var);
```

$$\begin{aligned}
&A\_ := [[[1,\,0,\,0,\,0,\,0,\,0,\,0] = [-1,\,2,\,-2-2\,I,\,2+2\,I,\,-4-4\,I,\,2+2\,I,\,0,\,0,\,0,\,0,\,0],\\
&[0,\,1,\,0,\,0,\,0,\,0,\,0] = [-3,\,5,\,-2-2\,I,\,2+2\,I,\,-4-4\,I,\,2+2\,I,\,0,\,0,\,0,\,0,\,0],\\
&[0,\,0,\,1,\,0,\,0,\,0,\,0] = [5,\,-10,\,8+4\,I,\,-9-4\,I,\,21+8\,I,\,-16-4\,I,\,0,\,0,\,3,\,0,\,0],\\
&[0,\,0,\,0,\,1,\,0,\,0,\,0] = [1,\,-3,\,4,\,-4,\,12,\,-12,\,0,\,0,\,3,\,0,\,0],\\
&[0,\,0,\,0,\,0,\,1,\,0,\,0] = [-3,\,9,\,-11,\,12,\,-33,\,32,\,0,\,0,\,-8,\,0,\,0],\\
&[0,\,0,\,0,\,0,\,0,\,1,\,0] = [0,\,0,\,0,\,0,\,0,\,0,\,0,\,1,\,0,\,0,\,0],\\
&[0,\,0,\,0,\,0,\,0,\,0,\,1] = [0,\,0,\,0,\,0,\,0,\,0,\,1,\,0,\,0,\,0,\,0]],[[2,\,0,\,0,\,0,\,0,\,0,\,0],\\
&[0,\,2,\,0,\,0,\,0,\,0,\,0],\,[0,\,0,\,2+2\,I,\,0,\,0,\,0,\,0],\,[0,\,0,\,0,\,2+2\,I,\,0,\,0,\,0],\\
&[0,\,0,\,0,\,0,\,6+6\,I,\,0,\,0]],\ \text{“Presentation”},\,[2,\,2,\,2+2\,I,\,2+2\,I,\,6+6\,I,\,0,\,0],\,2]
\end{aligned}$$

α_1 is the embedding map:

```
>   alpha1:=KernelEmb(A,alpha2,_A,var);
```

$$\alpha 1 := \begin{bmatrix} -1-2\,I & 0 & -1 & 0 & 0 & 2+2\,I & 0 & 0 & 0 & 0 & 0 \\ -2\,I & 1 & -3 & 0 & 0 & 2+2\,I & 0 & 0 & 0 & 0 & 0 \\ 2+4\,I & -1 & 4 & -1 & 0 & -6-4\,I & 3 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 3 & 1 & 4 & -8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The A-sequence is exact:

```
>   IsShortExactSeq(A_,alpha1,A,alpha2,_A,var,"VERBOSE");
```

$$\mathit{true}$$

B is the direct sum of $\operatorname{Hom}(TM,N)$ and M^{**}:

```
>   B:=DirectSum(HTMN,HHM,var);
```

$$\begin{aligned}
&B := [[[1,\,0,\,0,\,0] = [1,\,0,\,0,\,0],\,[0,\,1,\,0,\,0] = [-1,\,1,\,0,\,0],\,[0,\,0,\,1,\,0] = [0,\,0,\,1,\,0],\\
&[0,\,0,\,0,\,1] = [0,\,0,\,0,\,1]],\,[[2,\,0,\,0,\,0],\,[0,\,2+2\,I,\,0,\,0]],\,\text{“Presentation”},\,[2,\,2+2\,I,\,0,\,0],\,2]
\end{aligned}$$

B'' is the direct sum of $\operatorname{Hom}(TM,N)$ and the zero module:

```
>   _B:=DirectSum(HTMN,Z,var);
```

$$\_B := [[[1,\,0] = [-1,\,0,\,1],\,[0,\,1] = [0,\,-1,\,1]],\,[[2,\,0],\,[0,\,2+2\,I]],\,\text{“Presentation”},\,[2,\,2+2\,I],\,0]$$

β_2 is the direct sum of the map $b\,Id$ and the zero map χ:

```
>   beta2:=DirectSumMap(HTMN,HHM,MulMat(b,Id,var),chi,HTMN,Z,var);
```

$$\beta2 := \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

B' is the kernel of β_2:

```
>  B_:=Kernel(B,beta2,_B,var);
```

$$B\_ := [[[1, 0] = [0, 0, 1, 0], [0, 1] = [0, 0, 0, 1]], [[0, 0]], \text{“Presentation”}, [0, 0], 2]$$

β_1 is the embedding map:

```
>  beta1:=KernelEmb(B,beta2,_B,var);
```

$$\beta1 := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The B-sequence is in this example (depending on the choice of the number b) exact:

```
>  IsShortExactSeq(B_,beta1,B,beta2,_B,var,"VERBOSE");
```

$$\mathit{true}$$

ψ is the direct sum of the maps $a\,\eta$ and $d\,\varepsilon$:

```
>  psi:=DirectSumMap(HMN,M,MulMat(a,eta,var),MulMat(d,epsilon,var),HTMN,
>  HHM,var);
```

$$\psi := \begin{bmatrix} 1+I & 0 & 0 & 0 \\ 1+I & 0 & 0 & 0 \\ 1+I & 0 & 0 & 0 \\ -1-I & 1+I & 0 & 0 \\ 1+I & 1+I & 0 & 0 \\ 1+I & 1+I & 0 & 0 \\ 0 & -1-I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2+2\,I & 0 \\ 0 & 0 & 0 & 2+2\,I \end{bmatrix}$$

Some infos about ψ:

```
>  IsHom(A,psi,B,var);
```

$$\mathit{true}$$

```
>  IsSurjective(psi,B,var);
```

$$\mathit{false}$$

```
>  IsInjective(A,psi,B,var);
```

$$\mathit{false}$$

A necessary condition to be able to complete the square:

```
>  CheckKerSq(A,alpha2,_A,psi,B,beta2,_B,var);
```

$$[\%1, \%1, \%1, \%1, \%1, \%1, \%1]$$
$$\%1 := [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

Completing the square by ϕ, which is the direct sum of the map $c\,Id$ and the zero map ζ:

```
>  phi:=DirectSumMap(HTMN,FM,MulMat(c,Id,var),zeta,HTMN,Z,var);
```

$$\phi := \begin{bmatrix} -1-I & 0 \\ 1+I & -1-I \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Some infos about ϕ:

```
>  IsHom(_A,phi,_B,var);
```
$$\mathit{true}$$
```
>  IsSurjective(phi,_B,var);
```
$$\mathit{false}$$
```
>  IsInjective(_A,phi,_B,var);
```
$$\mathit{false}$$

Check the commutativity of the square:

```
>  IsCommutativeSq(alpha2,phi,psi,beta2,_B,var);
```
$$\mathit{true}$$

The induced kernel map τ:

```
>  tau:=KernelMap(A,alpha2,_A,psi,B,beta2,_B,var);
```
$$\tau := \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Some infos about τ:

```
>  IsHom(A_,tau,B_,var);
```
$$\mathit{true}$$
```
>  IsSurjective(tau,B_,var);
```
$$\mathit{false}$$
```
>  IsInjective(A_,tau,B_,var);
```
$$\mathit{false}$$

Check the commutativity of the square:

```
>  IsCommutativeSq(alpha1,psi,tau,beta1,B,var);
```
$$\mathit{true}$$

Compute the kernel sequence:

```
>  K:=Kernel(A,psi,B,var);
```
$$\begin{aligned}
&K := [[[1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0] = [0,\ 0,\ 2,\ -2,\ 4,\ -2,\ 0,\ 0,\ 0,\ 0,\ 0],\\
&[0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0] = [-1-I,\ 2+2\,I,\ 1-I,\ -2,\ 4,\ -2,\ 0,\ 0,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0] = [-1,\ 2,\ 2,\ -2,\ 4,\ -2,\ 0,\ 0,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0] = [-3,\ 5,\ 2,\ -2,\ 4,\ -2,\ 0,\ 0,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0] = [0,\ 0,\ 2,\ -1,\ 3,\ -2,\ 0,\ 0,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0] = [-1,\ 3,\ -1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0] = [6+I,\ -12-2\,I,\ -7+I,\ 8,\ -20,\ 12,\ 0,\ 0,\ -1,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0] = [0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1] = [0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0]],[\\
&[1+I,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0],\ [0,\ 1+I,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0],\ [0,\ 0,\ 2,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 2,\ 0,\ 0,\ 0,\ 0,\ 0],\ [0,\ 0,\ 0,\ 0,\ 2+2\,I,\ 0,\ 0,\ 0,\ 0],\ [0,\ 0,\ 0,\ 0,\ 0,\ 2+2\,I,\ 0,\ 0,\ 0],\\
&[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 6+6\,I,\ 0,\ 0]],\ \text{“Presentation”},\\
&[1+I,\ 1+I,\ 2,\ 2,\ 2+2\,I,\ 2+2\,I,\ 6+6\,I,\ 0,\ 0],\ 2]
\end{aligned}$$

```
>  K_:=Kernel(A_,tau,B_,var);
```

$K\_ := [[[1, 0, 0, 0, 0, 0, 0] = [-3, 5, -2-2I, 2+2I, -4-4I, 2+2I, 0, 0, 0, 0, 0],$
$[0, 1, 0, 0, 0, 0, 0] = [-2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],$
$[0, 0, 1, 0, 0, 0, 0] = [6, -11, 6+2I, -6-2I, 16+4I, -14-2I, 0, 0, 3, 0, 0],$
$[0, 0, 0, 1, 0, 0, 0] = [-4, 7, -4-4I, 5+4I, -9-8I, 4+4I, 0, 0, 0, 0, 0],$
$[0, 0, 0, 0, 1, 0, 0] = [0, 5, -11+4I, 11-4I, -36+8I, 40-4I, 0, 0, -11, 0, 0],$
$[0, 0, 0, 0, 0, 1, 0] = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],$
$[0, 0, 0, 0, 0, 0, 1] = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]], [[2, 0, 0, 0, 0, 0, 0],$
$[0, 2, 0, 0, 0, 0, 0], [0, 0, 2+2I, 0, 0, 0, 0], [0, 0, 0, 2+2I, 0, 0, 0],$
$[0, 0, 0, 0, 6+6I, 0, 0]],$ "Presentation", $[2, 2, 2+2I, 2+2I, 6+6I, 0, 0], 2]$

```
>  kappa1:=KernelMap(A_,tau,B_,alpha1,A,psi,B,var);
```

$$\kappa1 := \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 1 & 0 & 0 & -3 & 0 & 0 \\ -1 & 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
>  _K:=Kernel(_A,phi,_B,var,"var_to_assign_embedding_info"='_KK');
```

$\_K := [[[1, 0, 0, 0] = [-2, 2, 0, 0], [0, 1, 0, 0] = [-3-I, 2, 0, 0], [0, 0, 1, 0] = [0, 0, 1, 0],$
$[0, 0, 0, 1] = [0, 0, 0, 1]], [[1+I, 0, 0, 0], [0, 1+I, 0, 0]],$ "Presentation", $[1+I, 1+I, 0, 0], 2]$

```
>  copy(_KK);
```

$$\begin{bmatrix} 0 & 2 & 0 & 0 \\ -1-I & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
>  kappa2:=KernelMap(A,psi,B,alpha2,_A,phi,_B,var);
```

$$\kappa2 := \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The kernel sequence has a non-zero cokernel at K'':

```
>  IsShortExactSeq(K_,kappa1,K,kappa2,_K,var,"VERBOSE");
```

"homs" $= true$, "cmps" $= true$, "defs" $= [true, true,$
$[[[1, 0] = [0, 0, 1, 0], [0, 1] = [0, 0, 0, 1]], [[0, 0]],$ "Presentation", $[0, 0], 2]]$

Define the cokernel sequence:

```
>  C:=Cokernel(psi,B,var);
```

$C := [[[1, 0, 0, 0] = [-1, 1, 0, 0], [0, 1, 0, 0] = [-2, 1, 0, 0], [0, 0, 1, 0] = [3, -2, 0, 1],$
$[0, 0, 0, 1] = [0, 0, -1, 1]], [[1+I, 0, 0, 0], [0, 1+I, 0, 0], [0, 0, 2+2I, 0], [0, 0, 0, 2+2I]],$
"Presentation", $[1+I, 1+I, 2+2I, 2+2I], 0]$

```
>  C_:=Cokernel(tau,B_,var,"var_to_assign_embedding_info"='CC_');
```

$C\_ := [[[1, 0] = [0, 0, 1, 0], [0, 1] = [0, 0, 0, 1]], [[0, 0]],$ "Presentation", $[0, 0], 2]$

```
> copy(CC_);
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
> omega1:=CokernelMap(tau,B_,beta1,psi,B,var);
```

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

```
> _C:=Cokernel(phi,_B,var);
```

$$\_C := [[[1, 0] = [0, -1, 1], [0, 1] = [1, -1, 0]], [[1 + I, 0], [0, 1 + I]], \text{“Presentation”} , [1 + I, 1 + I], 0]$$

```
> omega2:=CokernelMap(psi,B,beta2,phi,_B,var);
```

$$\omega 2 := \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ -1 & 1 \\ 0 & 0 \end{bmatrix}$$

The cokernel sequence has a non-zero kernel at C':

```
> IsShortExactSeq(C_,omega1,C,omega2,_C,var,"VERBOSE");
```

$$\text{“homs”} = \mathit{true}, \text{“cmps”} = \mathit{true}, \text{“defs”} = [[[[1, 0] = [0, 0, 2 + 2\,I, 0], [0, 1] = [0, 0, 0, 2 + 2\,I]], [[0, 0]], \text{“Presentation”}, [0, 0], 2], \mathit{true}, \mathit{true}]$$

Compute the connecting homomorphism between the kernel and the cokernel sequence:

```
> delta:=ConnectingHom(_K,alpha2,psi,tau,beta1,C_,var,
> "Hqn_embedding_info"=_KK,"Hsn_1_embedding_info"=CC_,
> "Cqn_Bqn"=_A,"Zn_1"=B,"Zsn_1"=B_);
```

$$\delta := \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2 + 2\,I & 0 \\ 0 & 2 + 2\,I \end{bmatrix}$$

The resulting sequence is a long exact sequence:

```
> IsExactCoseq([K_,kappa1,K,kappa2,_K,delta,C_,omega1,C,omega2,_C],var,
> "VERBOSE");
```

$$\mathit{true}$$

VI. SYSTEM-THEORETIC INTERPRETATION OF HOMOLOGICAL CONSTRUCTIONS

For a finitely presented module M with relation matrix A denote by $M^\top$ the module with relation matrix $\theta(A)$ defined by $(\theta(A))_{ij} = \theta(A_{ji})$, where θ is the fixed involution coming with the ring. Of course $M^\top$ and $\mathrm{Ext}^0(M^\top, D)$ depend on the presentation of M. Nevertheless $\mathrm{Ext}^i(M^\top, D)$ for $i > 0$ only depends on the isomorphism type of M. For instance

$$\mathrm{Ext}^1(M^\top, D) \cong TM, \tag{1}$$

a fact that is often demonstrated in the following examples. Since one has the exact sequence

$$0 \to \mathrm{Ext}^1(M^\top, D) \to M \xrightarrow{\varepsilon} M^{**} \to \mathrm{Ext}^2(M^\top, D) \to 0,$$

the module M is reflexive, iff $\mathrm{Ext}^i(M^\top, D) = 0$ for $i = 1, 2$. Furthermore M is projective, iff $\mathrm{Ext}^i(M, D) = 0$ for all $0 < i \leq n$, where n is the global dimension (possibly infinite) of D. This is summarized in Table I (for more details see [15], [6]).

An immediate application of the homological machinery, is that all the homological constructions depend only on the isomorphism type of the module, i.e. on the intrinsic structural properties of the system (independent of the specific realization). By this, one can interpret these constructions as invariants, and one can distinguish between intrinsically different systems by finding a differing homological property.

VII. APPLICATIONS

A. A bipendulum

In this subsection we demonstrate methods for the study of structural properties of linear systems of ordinary differential equations with rational coefficients, i.e. systems defined over the Weyl algebra of differential operators with respect to time t with rational functions in t as coefficients. We consider the example of a mechanical system called bipendulum which consists of two pendula of length $l1$ respectively $l2$, fixed at the two ends of a bar [14]. The bar can be moved horizontally.

We load the package `homalg` and the ring-specific package JANET providing procedures for the algebraic analysis of linear systems of partial differential equations.

```
>  with(Janet):
>  with(homalg):
```

First we define the list of independent variables defining the noncommutative ring $\mathbb{Q}(t)[\frac{d}{dt}]$ and then the list of dependent variables which are the generators of the $\mathbb{Q}(t)[\frac{d}{dt}]$-module corresponding to the linear system:

```
>  ivar:=[t];dvar:=[x1,x2,x1t,x2t,u];
```

$$ivar := [t]$$

$$dvar := [x1,\ x2,\ x1t,\ x2t,\ u]$$

'Janet1' indicates that the package JANET will be used with one independent variable only. `homalg` will then use the JACOBSON normal form [7] for ordinary differential operators with rational coefficients to generate the best basis for a module. This demonstrates how the flexibility of `homalg` can be exploited by using different ring-specific features.

```
>  RPJ:='Janet/homalg';
```

$$RPJ := Janet/homalg$$

```
>  RPJ1:='Janet1/homalg';
```

$$RPJ1 := Janet1/homalg$$

The system of the bipendulum is described by equating the following system of ordinary differential expressions to 0:

```
>  R:=[diff(x1(t),t)-x1t(t),
>  diff(x2(t),t)-x2t(t), g/l1*x1(t)+
>  diff(x1t(t),t)+g/l1*u(t), g/l2*
>  x2(t)+diff(x2t(t),t)+g/l2*u(t)];
```

$$R := [(\tfrac{d}{dt}\,\mathrm{x1}(t)) - \mathrm{x1t}(t),\ (\tfrac{d}{dt}\,\mathrm{x2}(t)) - \mathrm{x2t}(t),$$
$$\frac{g\,\mathrm{x1}(t)}{l1} + (\tfrac{d}{dt}\,\mathrm{x1t}(t)) + \frac{g\,\mathrm{u}(t)}{l1},$$
$$\frac{g\,\mathrm{x2}(t)}{l2} + (\tfrac{d}{dt}\,\mathrm{x2t}(t)) + \frac{g\,\mathrm{u}(t)}{l2}]$$

Here g is the gravitational constant, $\mathrm{x1}(t)$ and $\mathrm{x2}(t)$ are the positions of the end points of the two pendula at time t and $\mathrm{u}(t)$ is the position of the bar at time t.

These differential expressions are obtained from a second order ordinary differential system by substituting $x1t$ for the derivative of $x1$ with respect to time t and similarly for the derivative of $x2$ with respect to t. Hence, we consider a first order linear system. The corresponding differential operator, which is expected as input for the `homalg` procedures, can be written as follows:

```
>  A:=Diff2Op(R, ivar, dvar);
```

$$A := \begin{bmatrix} [[1,\ [t]]] & 0 & [[-1,\ []]] & 0 & 0 \\ 0 & [[1,\ [t]]] & 0 & [[-1,\ []]] & 0 \\ [[\frac{g}{l1},\ []]] & 0 & [[1,\ [t]]] & 0 & [[\frac{g}{l1},\ []]] \\ 0 & [[\frac{g}{l2},\ []]] & 0 & [[1,\ [t]]] & [[\frac{g}{l2},\ []]] \end{bmatrix}$$

Here each entry is to be interpreted as a linear combination of the differential operators $\frac{d^i}{dt^i}$ represented by $[\underbrace{t, \ldots, t}_{i\ \text{times}}]$, $i \in \mathbb{Z}_{\geq 0}$. For example, $[[C1, [t,t,t]], [C2, [t]], [C3, []]]$ represents the differential operator $C1\,\frac{d^3}{dt^3} + C2\,\frac{d}{dt} + C3$.

We find a presentation of the module associated with the linear system over the Weyl algebra with rational coefficients, i.e. of the cokernel of $(.A)$:

```
>  M:=Cokernel(A, ivar, RPJ);
```

TABLE I

CHARACTERIZING SYSTEM/MODULE PROPERTIES

system	**module**	**homological algebra**
autonomous elements	$TM \neq 0$	$\mathrm{Ext}^1(M^\top, D) \neq 0$
controllability, parametrizability	$TM = 0$	$\mathrm{Ext}^1(M^\top, D) = 0$
parametrizability of the parametrization	reflexive	$\mathrm{Ext}^i(M^\top, D) = 0,$ $i = 1, 2$
...	...	...
int. stabilizability, BÉZOUT-identity, chain of n parametrizations	projective	$\mathrm{Ext}^i(M^\top, D) = 0,$ $1 \leq i \leq n$
flatness	free	in general no criteria but for a PID: torsion-free = free

$$
\begin{aligned}
&M := [[[[[1,\ []]],\ 0,\ 0] = [[[1,\ []]],\ 0,\ 0,\ 0,\ 0],\\
&[0,\ [[1,\ []]],\ 0] = [0,\ [[1,\ []]],\ 0,\ 0,\ 0],\\
&[0,\ 0,\ [[1,\ []]]] = [0,\ 0,\ 0,\ 0,\ [[1,\ []]]]],\\
&[[0,\ [[1,\ [t,\ t]],\ [\frac{g}{\mathit{l2}},\ []]],\\
&[[\frac{g}{\mathit{l2}},\ []]]],\ [[[1,\ [t,\ t]],\ [\frac{g}{\mathit{l1}},\ []]],\ 0,\ [[\frac{g}{\mathit{l1}},\ []]]]],\\
&\text{“Presentation”},\ 3 + 3\,s + \frac{s^2}{1-s},\ [1]]
\end{aligned}
$$

This presentation uses the above notation for differential operators. A more readable representation of M can be obtained by using the procedure `Pres2Diff` from the package JANET:

```
>  Pres2Diff(M, ivar, dvar);
```

$$
\left[[\_\mathrm{T1}(t) = \mathrm{x1}(t),\ \_\mathrm{T2}(t) = \mathrm{x2}(t),\ \_\mathrm{T3}(t) = \mathrm{u}(t)], \right.
$$
$$
\left[\frac{(\frac{d^2}{dt^2}\,\_\mathrm{T2}(t))\,\mathit{l2} + g\,\_\mathrm{T2}(t)}{\mathit{l2}} + \frac{g\,\_\mathrm{T3}(t)}{\mathit{l2}}, \right.
$$
$$
\left. \frac{(\frac{d^2}{dt^2}\,\_\mathrm{T1}(t))\,\mathit{l1} + g\,\_\mathrm{T1}(t)}{\mathit{l1}} + \frac{g\,\_\mathrm{T3}(t)}{\mathit{l1}} \right],
$$
$$
\left. \text{“Presentation”},\ 3 + 3\,s + \frac{s^2}{1-s},\ [1] \right]
$$

Using ’Janet1’ (and hence the JACOBSON normal form) it turns out that the cokernel is cyclic and even free (of rank 1). This only holds in the generic case $\mathit{l1} \neq \mathit{l2}$ because in the following computation $\mathit{l1} - \mathit{l2} \neq 0$ is assumed:

```
>  Pres2Diff(Cokernel(A, ivar, RPJ1),
>  ivar, dvar);
```

$$
\begin{aligned}
&[[\_\mathrm{T1}(t) = -\frac{\mathit{l1}\,\mathrm{x1}(t)}{\mathit{l2}} + \mathrm{x2}(t)],\ [0],\\
&\text{“Presentation”},\ \frac{1}{1-s},\ [1]]
\end{aligned}
$$

Now we study the case that the lengths $\mathit{l1}$, $\mathit{l2}$ of the pendula are equal:

```
>  R2:=subs(l2=l1, R);
```

$$
\begin{aligned}
&\mathit{R2} := [(\tfrac{d}{dt}\,\mathrm{x1}(t)) - \mathrm{x1t}(t),\ (\tfrac{d}{dt}\,\mathrm{x2}(t)) - \mathrm{x2t}(t),\\
&\frac{g\,\mathrm{x1}(t)}{\mathit{l1}} + (\tfrac{d}{dt}\,\mathrm{x1t}(t)) + \frac{g\,\mathrm{u}(t)}{\mathit{l1}},\\
&\frac{g\,\mathrm{x2}(t)}{\mathit{l1}} + (\tfrac{d}{dt}\,\mathrm{x2t}(t)) + \frac{g\,\mathrm{u}(t)}{\mathit{l1}}]
\end{aligned}
$$

The system needs to be converted to the differential operator form for the use of `homalg`:

```
>  A2:=Diff2Op(R2, ivar, dvar);
```

$$
\mathit{A2} := \begin{bmatrix}
[[1,\ [t]]] & 0 & [[-1,\ []]] & 0 & 0\\
0 & [[1,\ [t]]] & 0 & [[-1,\ []]] & 0\\
[[\frac{g}{\mathit{l1}},\ []]] & 0 & [[1,\ [t]]] & 0 & [[\frac{g}{\mathit{l1}},\ []]]\\
0 & [[\frac{g}{\mathit{l1}},\ []]] & 0 & [[1,\ [t]]] & [[\frac{g}{\mathit{l1}},\ []]]
\end{bmatrix}
$$

Again we find a presentation of the module associated with the linear system over the Weyl algebra with rational coefficients, i.e. of the cokernel of $(.A2)$:

```
>  M2:=Cokernel(A2, ivar, RPJ):
>  Pres2Diff(M2, ivar, dvar);
```

$$\left[[\_\mathrm{T1}(t) = \mathrm{x1}(t),\ \_\mathrm{T2}(t) = \mathrm{x2}(t),\ \_\mathrm{T3}(t) = \mathrm{u}(t)],\right.$$
$$\left[\frac{(\frac{d^2}{dt^2}\ \_\mathrm{T2}(t))\ \mathit{l1} + g\ \_\mathrm{T2}(t)}{\mathit{l1}} + \frac{g\ \_\mathrm{T3}(t)}{\mathit{l1}},\right.$$
$$\left.\frac{(\frac{d^2}{dt^2}\ \_\mathrm{T1}(t))\ \mathit{l1} + g\ \_\mathrm{T1}(t)}{\mathit{l1}} + \frac{g\ \_\mathrm{T3}(t)}{\mathit{l1}}\right],$$
$$\left.\text{“Presentation”},\ 3 + 3\,s + \frac{s^2}{1-s},\ [1]\right]$$

From this presentation the structural properties of the module are not evident at first sight. However, the JACOBSON normal form provides a different presentation with two generators, a torsion and a free one:

```
>  Pres2Diff(Cokernel(A2, ivar,
>  RPJ1), ivar, dvar);
```

$$\left[[\_\mathrm{T1}(t) = -\mathrm{x1}(t) + \mathrm{x2}(t),\ \_\mathrm{T2}(t) = \mathrm{x1}(t)],\right.$$
$$\left[\frac{(\frac{d^2}{dt^2}\ \_\mathrm{T1}(t))\ \mathit{l1} + g\ \_\mathrm{T1}(t)}{\mathit{l1}}\right],\ \text{“Presentation”},$$
$$\left.2 + 2\,s + \frac{s^2}{1-s},\ [1]\right]$$

In fact, using 'Janet' again, we find that the torsion submodule of $\mathrm{coker}(.A2)$ is generated by the difference of the positions $\mathrm{x1}(t)$, $\mathrm{x2}(t)$ of the end points of the two pendula, which is an autonomous element of the system. This autonomous element satisfies the second order ordinary differential equation given in the second entry of the result:

```
>  Pres2Diff(TorsionSubmodule(M2,
>  ivar, RPJ), ivar, dvar);
```

$$\left[[\_\mathrm{T1}(t) = \mathrm{x1}(t) - \mathrm{x2}(t)],\right.$$
$$\left.\left[\frac{(\frac{d^2}{dt^2}\ \_\mathrm{T1}(t))\ \mathit{l1} + g\ \_\mathrm{T1}(t)}{\mathit{l1}}\right],\ \text{“Presentation”},\ 1 + s,\ [0]\right]$$

B. *A satellite in a circular equatorial orbit*

In this subsection we apply homalg and OREMODULES to a linear system describing a satellite in a circular equatorial orbit. For more details see [11], p. 60 and p. 145, and [12], p. 6, p. 11 and p. 17, and the *Library of Examples* [4].

We load the package homalg and the ring-specific package OREMODULES providing procedures for the algebraic analysis of linear systems over Ore algebras.

```
>  with(OreModules):
```

```
>  with(homalg):

Warning, the name Involution
has been redefined
```

Since we only use the ring-specific package OREMODULES, we set the default package for homalg to 'OreModules':

```
>  `homalg/default`:=
>  `OreModules/homalg`;
```

$$\mathit{homalg/default} := \mathit{OreModules/homalg}$$

First, we define the Weyl algebra $\mathit{Alg} = A_1(\mathbb{Q}(\omega, m, r, a, b)) = \mathbb{Q}(\omega, m, r, a, b)[t][Dt]$, where Dt acts as differentiation w.r.t. time t. Note that we have to declare the parameters ω (angular velocity), m (mass of the satellite), r (radius component in the polar coordinates), a and b (parameters specifying the thrust) of the system in the definition of the Ore algebra:

```
>  Alg:=DefineOreAlgebra(diff=[Dt,t],
>  polynom=[t],comm=[omega,m,r,a,b]):
```

The linearized ordinary differential equations for the satellite in a circular orbit are given by the following matrix R. These equations describe the motion of the satellite in the equatorial plane, where the fifth and the sixth column of R incorporate the controls $\mathit{u1}$, $\mathit{u2}$ which represent radial thrust resp. tangential thrust caused by rocket engines ([11], p. 60 and p. 145).

```
>  R:=matrix([[Dt,-1,0,0,0,0],
>  [-3*omega^2,Dt,0,-2*omega*r,-a/m,
>  0], [0,0,Dt,-1,0,0],
>  [0,2*omega/r,0,Dt,0,-b/(m*r)]]);
```

$$R := \begin{bmatrix} Dt & -1 & 0 & 0 & 0 & 0 \\ -3\,\omega^2 & Dt & 0 & -2\,\omega\,r & -\dfrac{a}{m} & 0 \\ 0 & 0 & Dt & -1 & 0 & 0 \\ 0 & \dfrac{2\,\omega}{r} & 0 & Dt & 0 & -\dfrac{b}{m\,r} \end{bmatrix}$$

We find a presentation of the module associated with the linear system over the Weyl algebra Alg, i.e. of the cokernel[1] of $(.R)$:

```
>  M:=Cokernel(R, Alg);
```

[1]Cokernel uses several methods to construct a presentation with a small number of generators, i.e. if one generator can be expressed in terms of the others, then it is eliminated from the presentation. This is for example the case, if one of the relations contains a unit. The ring-package is responsible for recognizing the units. In the latest version of homalg and OREMODULES the output M of Cokernel is a presentation with two generators without relations, hence the cokernel is free of rank 2, and hence torsion-free. In order to demonstrate the computation of extension modules, we work with the presentation matrix R instead of M.

$$M := [[[1, 0, 0, 0] = [1, 0, 0, 0, 0, 0],$$
$$[0, 1, 0, 0] = [0, 0, 1, 0, 0, 0],$$
$$[0, 0, 1, 0] = [0, 0, 0, 0, 1, 0],$$
$$[0, 0, 0, 1] = [0, 0, 0, 0, 0, 1]],$$
$$[[2\,\omega\,Dt\,m,\ Dt^2\,m\,r,\ 0,\ -b],$$
$$[-3\,m\,\omega^2 + m\,Dt^2,\ -2\,\omega\,r\,Dt\,m,\ -a,\ 0]],$$
$$\text{“Presentation”},\ -\frac{2\,(s+1)}{-1+s} + \frac{2}{(-1+s)^2}]$$

We compute the formal adjoint of the differential operator R:

```
>  R_adj:=Involution(R, Alg);
```

$$R\_adj := \begin{bmatrix} -Dt & -3\,\omega^2 & 0 & 0 \\ -1 & -Dt & 0 & \frac{2\,\omega}{r} \\ 0 & 0 & -Dt & 0 \\ 0 & -2\,\omega\,r & -1 & -Dt \\ 0 & -\frac{a}{m} & 0 & 0 \\ 0 & 0 & 0 & -\frac{b}{m\,r} \end{bmatrix}$$

Some structural properties of the linear system under consideration are determined by computing the extension modules with values in Alg of the cokernel of $(.R\_adj)$. We compute the first extension module:

```
>  Ext_R(1, R_adj, Alg);
```

$$\left[\left[1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right],\ [1],\ \text{“Presentation”},\ 0\right]$$

From this presentation we see that the first extension module is zero. By (1) we conclude that the torsion submodule of the cokernel of $(.R)$ is zero. Hence, the system of the satellite is controllable.

```
>  TorsionSubmodule(R, Alg);
```

$$[[1 = [0, 0, 0, 0, 0, 0]], [1],\ \text{“Presentation”}, 0]$$

The next three statements demonstrate that this torsion submodule was computed by `homalg` using the procedure `ParametrizeModule` which returns a differential operator P such that the composition of R and P is zero. P defines a parametrization of the linear system given by R if and only if the kernel of $(.P)$ equals the image of $(.R)$, which means that the complex defined by these morphisms is exact. If we consider functions in an injective cogenerator (e.g. smooth functions for the present case of a time-invariant linear system, [6], [17]), then we have R y = 0 if and only if y = $P\ \xi$ for some vector of functions ξ. In general, P defines an embedding of the biggest possible factor module of the cokernel of $(.R)$ into a free module.

```
>  P:=ParametrizeModule(R, Alg);
```

$$P := \begin{bmatrix} b\,a & 0 \\ b\,a\,Dt & 0 \\ 0 & b\,a \\ 0 & b\,a\,Dt \\ -3\,b\,m\,\omega^2 + Dt^2\,b\,m & -2\,Dt\,b\,\omega\,r\,m \\ 2\,a\,Dt\,m\,\omega & a\,Dt^2\,m\,r \end{bmatrix}$$

```
>  Compose(R, P, Alg);
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

```
>  DefectOfHoms(R, P, Alg);
```

$$[[1 = [0, 0, 0, 0, 0, 0]], [1],\ \text{“Presentation”}, 0]$$

Since the system is controllable, we now check whether the system is flat [9], [6]. Every left-inverse of the parametrization P gives a flat output of the system:

```
>  S:=Leftinverse(P, Alg);
```

$$S := \begin{bmatrix} \frac{1}{b\,a} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{b\,a} & 0 & 0 & 0 \end{bmatrix}$$

Therefore, $(\xi_1, \xi_2)^T = S\,(x_1, x_2, x_3, x_4, u_1, u_2)^T$ is a flat output of the system which satisfies $(x_1, x_2, x_3, x_4, u_1, u_2) = P\,(\xi_1, \xi_2)^T$.

We notice that this flat output exists only if $ab \neq 0$. Hence, in the generic case the system is flat. Equivalently, the cokernel of $(.R)$ is free and, in particular, projective.

Let us remember that the full row-rank matrix R admits a right-inverse if and only if the cokernel of $(.R)$ is projective. By the theorem of Quillen-Suslin [16], [8] for modules over commutative polynomial rings, projectiveness is the same as freeness. So, $\mathrm{coker}(.R)$ is projective which we could also have discovered by succeeding to compute a right-inverse of R:

```
>  Rightinverse(R, Alg);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -\frac{Dt\,m}{a} & -\frac{m}{a} & \frac{2\,\omega\,r\,m}{a} & 0 \\ -\frac{2\,m\,\omega}{b} & 0 & -\frac{Dt\,m\,r}{b} & -\frac{m\,r}{b} \end{bmatrix}$$

Following [12], we modify the description of the control of the satellite in the system. If the rocket engines are commanded from the earth, then, due to transmission time, a constant time-delay occurs in the system. Hence, we enlarge the above Ore algebra by a shift operator δ:

```
>  Alg2:=DefineOreAlgebra(
>  diff=[Dt,t],dual_shift=[delta,s],
>  polynom=[t,s],comm=[omega,m,r,a,
>  b], shift_action=[delta,t]):
```

The system matrix is given as follows:

```
> R2:=matrix([[Dt,-1,0,0,0,0],
> [-3*omega^2,Dt,0,-2*omega*r,
> -a*delta/m,0], [0,0,Dt,-1,0,0],
> [0,2*omega/r,0,Dt,0,
> -b*delta/(m*r)]]);
```

$$R2 := \begin{bmatrix} Dt & -1 & 0 & 0 & 0 & 0 \\ -3\,\omega^2 & Dt & 0 & -2\,\omega\,r & -\frac{a\,\delta}{m} & 0 \\ 0 & 0 & Dt & -1 & 0 & 0 \\ 0 & \frac{2\,\omega}{r} & 0 & Dt & 0 & -\frac{b\,\delta}{m\,r} \end{bmatrix}$$

We define a formal adjoint $R2\_adj$ of $R2$ using an involution of $Alg2$:

```
> R2_adj:=Involution(R2, Alg2);
```

$$R2\_adj := \begin{bmatrix} -Dt & -3\,\omega^2 & 0 & 0 \\ -1 & -Dt & 0 & \frac{2\,\omega}{r} \\ 0 & 0 & -Dt & 0 \\ 0 & -2\,\omega\,r & -1 & -Dt \\ 0 & \frac{a\,\delta}{m} & 0 & 0 \\ 0 & 0 & 0 & \frac{b\,\delta}{m\,r} \end{bmatrix}$$

We check controllability and parametrizability of the system:

```
> Ext_R(1, R2_adj, Alg2);
```

$$\left[\left[1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right], [1], \text{“Presentation”}, 0\right]$$

We find that the first extension module with values in $Alg2$ of the cokernel of $(.R2\_adj)$ is generically the zero module. Equivalently, the system is generically controllable, i.e. parametrizable.

We continue to study the structural properties of the system by examining the algebraic properties of the cokernel of $(.R2)$. The next step is to compute the second extension module with values in $Alg2$ of the cokernel of $(.R2\_adj)$:

```
> Ext_R(2, R2_adj, Alg2);
```

$$[[[1,\,0] = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ [0,\,1] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}],$$
$$[[0,\,\delta],\ [\delta,\,0],\ [2\,\omega\,Dt,\ Dt^2\,r],\ [Dt^2 - 3\,\omega^2,\ -2\,Dt\,\omega\,r]],$$
$$\text{“Presentation”},\ \frac{2\,(s+1)}{(-1+s)^2}]$$

The second extension module is not zero. Hence, the cokernel of $(.R2)$ is not projective. Since $R2$ has full row-rank, this is equivalent to the fact that $R2$ does not admit a right-inverse:

```
> Rightinverse(R2, Alg2);
```

$$FAIL$$

In the special case where $a = 1$ and $b = 0$, i.e. the case where there is only a radial thrust, we have the following system matrix:

```
> R20:=subs(a=1, b=0, copy(R2));
```

$$R20 := \begin{bmatrix} Dt & -1 & 0 & 0 & 0 & 0 \\ -3\,\omega^2 & Dt & 0 & -2\,\omega\,r & -\frac{\delta}{m} & 0 \\ 0 & 0 & Dt & -1 & 0 & 0 \\ 0 & \frac{2\,\omega}{r} & 0 & Dt & 0 & 0 \end{bmatrix}$$

A presentation of the first extension module with values in $Alg2$ of the cokernel of the formal adjoint of $(.R20)$ is given by:

```
> Ext_R(1, Involution(R20, Alg2),
> Alg2);
```

$$\left[\left[1 = \begin{bmatrix} 2\,\omega \\ 0 \\ 0 \\ r \\ 0 \\ 0 \end{bmatrix}\right], [Dt], \text{“Presentation”}, -\frac{1}{(-1+s)^3}\right]$$

Hence, we find a torsion element of the cokernel of $(.R20)$ which corresponds to an autonomous element of the satellite system. Using the procedure `TorsionSubmodule` of `homalg` this presentation can be obtained directly:

```
> TorsionSubmodule(R20, Alg2);
```

$$[[1 = [2\,\omega,\ 0,\ 0,\ r,\ 0,\ 0]],\ [Dt],\ \text{“Presentation”},$$
$$-\frac{1}{(-1+s)^3}]$$

REFERENCES

[1] M. Barakat, D. Robertz, *First steps to an abstract package for homological algebra*, to appear, Proceedings EACA 2006, Spain.

[2] M. Barakat, D. Robertz, `homalg`*: An abstract package for homological algebra*, in preparation.

[3] Y. A. Blinkov, C. F. Cid, V. P. Gerdt, W. Plesken, D. Robertz, *The MAPLE Package “Janet”: I. Polynomial Systems. II. Linear Partial Differential Equations.* Proc. 6th Int. Workshop on Computer Algebra in Scientific Computing, Passau, 2003. Cf. also `http://wwwb.math.rwth-aachen.de/Janet`.

[4] F. Chyzak, A. Quadrat, D. Robertz, OREMODULES project, `http://wwwb.math.rwth-aachen.de/OreModules`.

[5] F. Chyzak, A. Quadrat, D. Robertz, OREMODULES: *A symbolic package for the study of multidimensional linear systems.* Proceedings MTNS 2004, Belgium.

[6] F. Chyzak, A. Quadrat, D. Robertz, *Effective algorithms for parametrizing linear control systems over Ore algebras*, Appl. Algebra Engrg. Comm. Comput. **16** (2005), pp. 319–376.

[7] P. M. Cohn, *Free Rings and their Relations*, Academic Press, second edition, 1985.

[8] A. Fabiańska, A. Quadrat, *Flat shift-invariant multidimensional linear systems are algebraically equivalent to controllable 1-D linear systems*, to appear, Proceedings MTNS 2006, Japan.

[9] M. Fliess, J. Lévine, P. Martin, P. Rouchon, *Flatness and defect of nonlinear systems: introductory theory and examples*, Int. J. Control **61** (1995), pp. 1327–1361.

[10] P. J. Hilton, U. Stammbach, *A Course in Homological Algebra*, second edition, Springer, 1997.

[11] T. Kailath, *Linear Systems*, Prentice-Hall, 1980.

[12] H. Mounier, *Propriétés structurelles des systèmes linéaires à retards: aspects théoriques et pratiques*, PhD thesis, University of Orsay, France, 1995.

[13] W. Plesken, D. Robertz. *Janet's approach to presentations and resolutions for polynomials and linear pdes.* Arch. Math. **84**:1 (2005), pp. 22–37.

[14] J.-F. Pommaret, *Partial Differential Control Theory*, Kluwer, 2001.

[15] A. Quadrat, *Analyse algébrique des systèmes de contrôle linéaires multidimensionnels*, PhD thesis, Ecole Nationale des Ponts et Chaussées, France, 1999.

[16] J. J. Rotman, *An Introduction to Homological Algebra*, Academic Press, 1979.

[17] E. Zerz, *Topics in Multidimensional Linear Systems Theory*, Lecture Notes in Control and Information Sciences 256, Springer, 2000.