OreMorphisms: A Homological Algebraic Package for Factoring, Reducing and Decomposing Linear Functional Systems

Thomas Cluzeau[1] and Alban Quadrat[2]

[1] University of Limoges; CNRS; XLIM UMR 6172, DMI; ENSIL, 16 Rue d'Atlantis, Parc Ester Technopole, 87068 Limoges Cedex, France
cluzeau@ensil.unilim.fr

[2] INRIA Sophia Antipolis, APICS Project, 2004 Route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France
Alban.Quadrat@sophia.inria.fr

Summary. The purpose of this paper is to demonstrate the symbolic package OreMorphisms which is dedicated to the implementation of different algorithms and heuristic methods for the study of the factorization, reduction and decomposition problems of general linear functional systems (e.g., systems of partial differential or difference equations, differential time-delay systems). In particular, we explicitly show how to decompose a differential time-delay system (a string with an interior mass [15]) formed by 4 equations in 6 unknowns and prove that it is equivalent to a simple equation in 3 unknowns. We finally give a list of reductions of classical systems of differential time-delay equations and partial differential equations coming from control theory and mathematical physics.

Introduction

In [6], we have recalled the main theoretical results of [5] on the factorization, reduction and decomposition problems for general linear functional systems obtained within a constructive homological algebra approach. The purpose of this paper is to demonstrate the Maple package OreMorphisms which is dedicated to the implementation of those results.

The OreMorphisms package focuses on the following problems:

- Compute D-morphisms between two finitely presented left D-modules over certain classes of Ore algebras D, i.e., the ones implemented in the package *Ore_algebra* available in the current Maple releases.
- Compute idempotents of the endomorphism ring $\mathrm{end}_D(M)$ of a finitely presented left D-module M (i.e., $f \in \mathrm{end}_D(M)$, $f^2 = f$) and, among the latter, those further defined by idempotent matrices P and Q, i.e., $P^2 = P$ and $Q^2 = Q$.

- Compute presentations of the kernel, image, cokernel, and coimage of a given morphism. Test whether or not a given morphism is injective, surjective or defines a D-isomorphism.
- Compute factorizations, reductions and decompositions of linear functional systems.

The package OreMorphisms is based on the Maple library OreModules ([2, 4]) devoted to the symbolic study of multidimensional systems. The OreModules library and its subpackage OreMorphisms are both freely available. For more details, see [2] for OreModules and [7] for OreMorphisms.

1 Computational issues and OreMorphisms functions

Hereafter, we use the notations of [6]. In particular, D denotes a noncommutative polynomial ring and $R \in D^{q \times p}$.

To implement our algorithms for factoring, reducing and decomposing linear functional systems, we mainly need to be able to constructively perform the two following tasks:

1. Compute a finite number of pairs (P, Q) of matrices which define D-endomorphisms of the left D-module $M = D^{1 \times p}/(D^{1 \times q}\, R)$, i.e., satisfy the relation $R\,P = Q\,R$. From these pairs, compute those that define idempotent D-endomorphisms (namely, $f \in \mathrm{end}_D(M)$ satisfying $f^2 = f$) defined by idempotent matrices P and Q, i.e., $P^2 = P$, $Q^2 = Q$.
2. Decide whether a left D-module is free and, if so, compute a basis of it.

In the package OreMorphisms, we have implemented algorithms handling the first point. In certain cases, the computation of idempotent D-endomorphisms defined by idempotent matrices P and Q reduces to calculating the solutions of the algebraic Riccati equation (see [5, Lemma 4.4])

$$\Lambda\, R \Lambda + (P - I_p)\, \Lambda + \Lambda\, Q + Z = 0, \tag{1}$$

where $P^2 = P + Z\,R$, $Q^2 = Q + R\,Z$, $R\,P = Q\,R$, $Z \in D^{p \times q}$ and R has a *full row rank*, namely, $\ker_D(.R) \triangleq \{\lambda \in D^{1 \times q} \mid \lambda\, R = 0\} = 0$. Then, the matrices $\overline{P} = P + \Lambda\, R$ and $\overline{Q} = Q + R\, \Lambda$ satisfy $\overline{P}^2 = \overline{P}$, $\overline{Q}^2 = \overline{Q}$ and $R\,\overline{P} = \overline{Q}\,R$. Hence, we have also implemented an algorithm computing solutions of (1).

For the second point, testing whether or not a given finitely presented left D-module over an Ore algebra D is free has been constructively studied in [2, 4] and algorithmic and heuristic methods have been implemented in OreModules. Moreover, the (non-trivial) problem of computing bases of free left modules over certain classes of Ore algebras has been made algorithmic:

- When D is the ring of ordinary differential or shift operators with rational coefficients, this can be achieved by means of *Jacobson normal form* computations (see, e.g., [9]). Jacobson normal forms have been implemented by G. Culianez for certain Ore algebras in the package Jacobson ([9]) of the library OreModules.
- When D is a commutative Ore algebra (e.g., differential time-delay operators with constant coefficients), we can use constructive versions of the *Quillen-Suslin theorem* of Serre's conjecture (see, e.g., [11]). A constructive algorithm has recently been implemented by A. Fabiańska (Aachen University) in the Maple package QuillenSuslin ([10, 11]).
- When D is the *Weyl algebra* $A_n(\mathbb{Q})$ (resp., $B_n(\mathbb{Q})$) of differential operators with polynomial (resp., rational) coefficients, a recent algorithm for computing bases of free left D-modules has been developed in [18] based on *Stafford's theorems*. This algorithm is implemented in the package Stafford of the OreModules library ([2]).

A list of OreMorphisms functions is given in Table 1. We use the notation A for the ring of functional operators as D is protected in Maple. The suffix "ConstCoeff" (resp., "Rat") distinguishes the procedures which deal with constant (resp., rational) coefficients from those dealing with polynomial coefficients (no suffix means that the procedures handle the polynomial coefficients case). In this table, $R \in A^{q \times p}$ and $R' \in A^{q' \times p'}$ denote two matrices with coefficients in an Ore algebra A handled in the Maple package *Ore_algebra*. We denote $M = A^{1 \times p}/(A^{1 \times q}\, R)$ and $M' = A^{1 \times p'}/(A^{1 \times q'}\, R')$ the two associated finitely presented left A-modules.

2 A worked example using the package OreMorphisms

We consider the model of a string with an interior mass studied in [15]

$$\begin{cases} \phi_1(t) + \psi_1(t) - \phi_2(t) - \psi_2(t) = 0, \\ \dot{\phi}_1(t) + \dot{\psi}_1(t) + \eta_1\, \phi_1(t) - \eta_1\, \psi_1(t) - \eta_2\, \phi_2(t) + \eta_2\, \psi_2(t) = 0, \\ \phi_1(t - 2\, h_1) + \psi_1(t) - u(t - h_1) = 0, \\ \phi_2(t) + \psi_2(t - 2\, h_2) - v(t - h_2) = 0, \end{cases} \tag{2}$$

where η_1, η_2 are constant parameters and $h_1, h_2 \in \mathbb{R}_+$ are such that $\mathbb{Q}\, h_1 + \mathbb{Q}\, h_2$ is a 2-dimensional $\mathbb{Q}$-vector space. Let us denote by $A = \mathbb{Q}(\eta_1, \eta_2)\,[d, \sigma_1, \sigma_2]$ the commutative polynomial algebra of differential incommensurable time-delay operators in d, σ_1 and σ_2, where:

$$d\, f(t) = \dot{f}(t), \quad \sigma_1 f(t) = f(t - h_1), \quad \sigma_2 f(t) = f(t - h_2).$$

The system matrix $R \in A^{4 \times 6}$ of (2) is defined by:

```
> with(OreModules):
> with(OreMorphisms): with(linalg):
```

Table 1. List of the main functions of the package OreMorphisms

`Morphisms(ConstCoeff,Rat)`	Compute a finite family of matrices $P \in A^{p \times p'}$ which define elements of $\hom_A(M, M')$, i.e., such that there exist matrices $Q \in A^{q \times q'}$ satisfying the relation $R\,P = Q\,R'$
`Idempotents(ConstCoeff,Rat)`	Compute a finite family of matrices $P \in A^{p \times p}$ defining idempotent elements of $\mathrm{end}_A(M)$, i.e., such that there exist three matrices $Q \in A^{q \times q}$, $Z \in A^{p \times q}$ and $Z' \in A^{q \times r}$ satisfying the relations $R\,P = Q\,R$, $P^2 = P + Z\,R$, $Q^2 = Q + R\,Z + Z'\,R_2$, where $\ker_A(.R) = A^{1 \times r}\,R_2$
`IdempotentsMat(ConstCoeff,Rat)`	Compute a finite family of idempotent matrices $P \in A^{p \times p}$ defining idempotent elements of $\mathrm{end}_A(M)$, i.e., such that there exist matrices $Q \in A^{q \times q}$ satisfying the relations $R\,P = Q\,R$, $P^2 = P$, $Q^2 = Q$ (R has full row rank)
`Riccati(ConstCoeff,Rat)`	Find a finite family of solutions $\Lambda \in A^{p \times q}$ of the algebraic Riccati equation $$\Lambda\,R\,\Lambda + (P - I_p)\,\Lambda + \Lambda\,Q + Z = 0$$ where the pair (P, Q) defines an idempotent element of $\mathrm{end}_A(M)$, i.e., satisfies the relations $R\,P = Q\,R$, $P^2 = P + Z\,R$, $Q^2 = Q + R\,Z$, with $Z \in A^{p \times q}$ (R has full row rank)
`KerMorphism(Rat)`	Compute the kernel of $f \in \hom_A(M, M')$, i.e., compute $S \in A^{r \times p}$ and $X \in A^{s \times r}$ such that we have $\ker f = (A^{1 \times r}\,S)/(A^{1 \times q}\,R) \cong A^{1 \times r}/(A^{1 \times s}\,X)$
`ImMorphism(Rat)`	Compute the image of $f \in \hom_A(M, M')$ defined by a pair of matrices (P, Q), i.e., $\mathrm{im}\, f = (A^{1 \times (p+q')}\,(P^T \quad R'^T)^T)/(A^{1 \times q'}\,R')$, by reducing the rows of the matrix $(P^T \quad R'^T)^T$ modulo the left A-module $A^{1 \times q'}\,R'$
`CoimMorphism(Rat)`	Compute the coimage of $f \in \hom_A(M, M')$, i.e., compute a matrix $S \in A^{r \times p}$ such that $\mathrm{coim}\, f = A^{1 \times p}/(A^{1 \times r}\,S)$
`CokerMorphism(Rat)`	Compute the cokernel of $f \in \hom_A(M, M')$ defined by a pair of matrices (P, Q), i.e., $\mathrm{coker}\, f = A^{1 \times p'}/(A^{1 \times (p+q')}\,(P^T \quad R'^T)^T)$
`TestInj(Rat)`	Test whether or not a given element of $\hom_A(M, M')$ is injective
`TestSurj(Rat)`	Test whether or not a given element of $\hom_A(M, M')$ is surjective
`TestIso(Rat)`	Test whether or not a given element of $\hom_A(M, M')$ is an A-isomorphism

`HeuristicReduction(Rat)`	Compute a reduction of the matrix R, i.e., compute an equivalent matrix with a block-triangular form. The heuristic part corresponds to the computation of bases of the different free left A-modules
`HeuristicDecomposition(Rat)`	Compute a decomposition of the matrix R, i.e., compute an equivalent matrix with a block-diagonal form. The heuristic part corresponds to the computation of bases of the different free left A-modules

```
> A:=DefineOreAlgebra(diff=[d,t],dual_shift=[sigma[1],x[1]],
> dual_shift=[sigma[2],x[2]],polynom=[t,x[1],x[2]],
> comm=[eta[1],eta[2]]):
```

```
> R:=matrix(4,6,[1,1,-1,-1,0,0,d+eta[1],d-eta[1],-eta[2],
> eta[2],0,0, sigma[1]^2,1,0,0,-sigma[1],0,0,0,1,sigma[2]^2,
> 0,-sigma[2]]);
```

$$R := \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 \\ d+\eta_1 & d-\eta_1 & -\eta_2 & \eta_2 & 0 & 0 \\ {\sigma_1}^2 & 1 & 0 & 0 & -\sigma_1 & 0 \\ 0 & 0 & 1 & {\sigma_2}^2 & 0 & -\sigma_2 \end{bmatrix}$$

2.1 Factorization problem

We show how to use OREMORPHISMS for computing a factorization of R of the form $R = L\,S$. We first need to compute the endomorphism ring $\mathrm{end}_A(M)$ of the A-module $M = A^{1\times 6}/(A^{1\times 4}\,R)$ finitely presented by the matrix R.

```
> Endo:=MorphismsConstCoeff(R,R,A):
```

Then, we choose a particular morphism f by selecting the first element P_1 of $Endo[1]$ and compute a matrix Q_1 satisfying $R\,P_1 = Q_1\,R$. The latter operation can be performed by means of the *Factorize* procedure of OREMODULES.

```
> P[1]:=Endo[1,1]; Q[1]:=Factorize(Mult(R,P[1],A),R,A);
```

$$P_1 := \begin{bmatrix} 0 & 0 & \eta_2\,\sigma_2 & \eta_2\,\sigma_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2\,\eta_1 & \eta_2\,\sigma_2 & 0 & 0 & 0 \\ 0 & -\sigma_2\,\eta_1 & 0 & \eta_2\,\sigma_2 & 0 & 0 \\ 0 & 0 & \eta_2\,\sigma_2\,\sigma_1 & \eta_2\,\sigma_2\,\sigma_1 & 0 & 0 \\ 0 & \eta_1 - {\sigma_2}^2\eta_1 & 0 & 0 & 0 & \eta_2\,\sigma_2 \end{bmatrix}$$

$$Q_1 := \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\eta_2\,\sigma_2\,\eta_1 - \eta_2\,\sigma_2\,d & \eta_2\,\sigma_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \eta_2\,\sigma_2 \end{bmatrix}$$

By [6, Theorem 2], the matrix S that we are searching for is the one defining the coimage of the endomorphism f of M defined by the previous matrices P_1 and Q_1. So, we compute it using the *CoimMorphism* procedure.

```
> S:=CoimMorphism(R,R,P[1],Q[1],A)[1];
```

$$S := \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_1 & \sigma_1 & -1 & 0 \\ 0 & 0 & 1 & \sigma_2^2 & 0 & -\sigma_2 \\ 0 & 0 & -d + \eta_2 - \eta_1 & -d - \eta_2 - \eta_1 & 0 & 0 \end{bmatrix}$$

The matrix L such that $R = L\,S$ can be obtained by right factoring R by S.

```
> L:=Factorize(R,S,A);
```

$$L := \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ d + \eta_1 & d - \eta_1 & 0 & 1 & 0 & 0 \\ \sigma_1^2 & 1 & \sigma_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

We note that choosing another endomorphism of M, i.e., another element of *Endo*[1], would lead to another factorization of the matrix R.

2.2 Reduction problem

We use the package OreMorphisms to reduce the matrix R, i.e., to find an equivalent matrix with a block-triangular form. By [6, Theorem 3], this can be done using an endomorphism of M defined by a pair of matrices P and Q provided that the A-modules $\ker_A(.P)$, $\operatorname{coim}_A(.P)$, $\ker_A(.Q)$ and $\operatorname{coim}_A(.Q)$ are free. We use the library OreModules to check that these properties are fulfilled and use a heuristic method to compute bases of those free A-modules. We then form the matrices U and V as defined in [6, Theorem 3]. We note that we generally need to use the package QuillenSuslin to compute bases of free modules over a commutative polynomial ring.

```
> U1:=SyzygyModule(P[1],A): EU:=Exti(Involution(U1,A),A,1):
> U2:=LeftInverse(EU[3],A): U:=stackmatrix(U1,U2);
> V1:=SyzygyModule(Q[1],A): EV:=Exti(Involution(V1,A),A,1):
> V2:=LeftInverse(EV[3],A): V:=stackmatrix(V1,V2);
```

$$U := \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_1 & \sigma_1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad V := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, we can compute the reduction $V\,R\,U^{-1}$ of the matrix R:

```
> R_red:=Mult(V,R,LeftInverse(U,A),A);
```

$$R\_red := \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ {\sigma_1}^2 & 1 & \sigma_1 & 0 & 0 & 0 \\ d+\eta_1 & d-\eta_1 & 0 & -\eta_1-\eta_2-d & -2\,\eta_2 & 0 \\ 0 & 0 & 0 & -{\sigma_2}^2 & 1-{\sigma_2}^2 & -\sigma_2 \end{bmatrix}$$

This reduction can be obtained using the *HeuristicReduction* procedure.

```
> HeuristicReduction(R,P[1],A)[1];
```

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ {\sigma_1}^2 & 1 & \sigma_1 & 0 & 0 & 0 \\ d+\eta_1 & d-\eta_1 & 0 & -\eta_1-\eta_2-d & -2\,\eta_2 & 0 \\ 0 & 0 & 0 & -{\sigma_2}^2 & 1-{\sigma_2}^2 & -\sigma_2 \end{bmatrix}$$

2.3 Decomposition problem

We now show how to use the package OreMorphisms to decompose the differential time-delay linear system (2), i.e., to find an equivalent system defined by a block-diagonal matrix. To achieve this decomposition, we first need to compute idempotent endomorphisms of M that are defined by idempotent matrices P and Q i.e., $R\,P = Q\,R$, $P^2 = P$ and $Q^2 = Q$. A way to do that is to use the procedure *IdempotentsMatConstCoeff* of OreMorphisms. We need to specify the total order in d, σ_1 and σ_2 of the idempotent matrix P, a piece of information which is specified by the fourth entry of the procedure. We first start by searching for idempotents of M defined by constant matrices.

```
> Idem_order0:=IdempotentsMatConstCoeff(R,Endo[1],A,0)[1];
```

$$Idem\_order0 := \left[\begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&-1&1&1&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix} \right]$$

We choose the non-trivial idempotent, i.e., the second entry of $Idem\_order0$:

```
> P[2]:=Idem_order0[2]; Q[2]:=Factorize(Mult(R,P[2],A),R,A);
```

$$P_2 := \begin{bmatrix} 0&-1&1&1&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix} \quad Q_2 := \begin{bmatrix} 0&0&0&0 \\ -d-\eta_1&1&0&0 \\ -{\sigma_1}^2&0&1&0 \\ 0&0&0&1 \end{bmatrix}$$

The fact that $P_2^2 = P_2$ and $Q_2^2 = Q_2$ imply that the A-modules $\ker_A(.P_2)$, $\ker_A(.Q_2)$, $\mathrm{im}_A(.P_2) = \ker_A(.(I_6 - P_2))$ and $\mathrm{im}_A(.Q_2) = \ker_A(.(I_4 - Q_2))$ are projective, and thus, free by the Quillen-Suslin theorem. We need to compute bases of those free A-modules. We then form the matrices U and V as explained in [6, Theorem 4].

```
> U1:=SyzygyModule(P[2],A):
> U2:=SyzygyModule(evalm(1-P[2]),A):
> U:=stackmatrix(U1,U2);
> V1:=SyzygyModule(Q[2],A):
> V2:=SyzygyModule(evalm(1-Q[2]),A):
> V:=stackmatrix(V1,submatrix(V2,[1, 2, 4],1..4));
```

$$U := \begin{bmatrix} 1&1&-1&-1&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix} \quad V := \begin{bmatrix} 1&0&0&0 \\ d+\eta_1&-1&0&0 \\ {\sigma_1}^2&0&-1&0 \\ 0&0&0&1 \end{bmatrix}$$

Now, we can compute the corresponding decomposition $V\,R\,U^{-1}$ of R:

```
> R_dec:=Mult(V,R,LeftInverse(U,A),A);
```

$$R\_dec := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\,\eta_1 & -d+\eta_2-\eta_1 & -d-\eta_2-\eta_1 & 0 & 0 \\ 0 & {\sigma_1}^2-1 & -{\sigma_1}^2 & -{\sigma_1}^2 & \sigma_1 & 0 \\ 0 & 0 & 1 & {\sigma_2}^2 & 0 & -\sigma_2 \end{bmatrix}$$

We can now try to decompose the second diagonal block matrix S of *R_dec*:

```
>  S:=submatrix(R_dec,2..4,2..6):
```

We apply the same technique as above: compute the endomorphism ring of the A-module $N = A^{1\times 5}/(A^{1\times 3}\,S)$ finitely presented by S, find one idempotent defined by idempotent matrices, compute bases of the free A-modules defined by their kernels and images, form the corresponding unimodular matrices and deduce the decomposition of S.

```
>  Endo1:=MorphismsConstCoeff(S,S,A):
>  Idem1_order0:=IdempotentsMatConstCoeff(S,Endo1[1],A,0)[1];
```

$$Idem1\_order0 := \left[\begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{bmatrix}, \begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix} \right]$$

We do not obtain a non-trivial idempotent of order 0 by means of the *IdempotentsMatConstCoeff* procedure. Hence, we can try another technique which searches for idempotents which are obtained by homotopies from the trivial idempotent id_N defined by $P_3 = I_5$ and $Q_3 = I_3$, i.e., $S\,P_3 = Q_3\,S$.

```
>  P[3]:=diag(1$5): Q[3]:=diag(1$3): Z[3]:=matrix(5,3,[0$15]):
```

We then need to solve the algebraic Riccati equation $\Lambda\,S\,\Lambda + \Lambda = 0$:

```
>  Mu:=RiccatiConstCoeff(S,P[3],Q[3],Z[3],A,0,alpha):
```

We choose one solution Λ_1 of the previous algebraic Riccati equation:

```
>  Lambda[1]:=subs({b321=0,b521=0},Mu[1,2]);
```

$$\Lambda_1 := \begin{bmatrix} 0&1&0 \\ 0&1&0 \\ 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{bmatrix}$$

We get a non-trivial idempotent defined by the following matrices P_4 and Q_4:

```
> P[4]:=simplify(evalm(P[3]+Mult(Lambda[1],S,A)));
> Q[4]:=simplify(evalm(Q[3]+Mult(S,Lambda[1],A)));
```

$$P_4 := \begin{bmatrix} {\sigma_1}^2 & -{\sigma_1}^2 & -{\sigma_1}^2\,\sigma_1 & 0 \\ {\sigma_1}^2-1 & -{\sigma_1}^2+1 & -{\sigma_1}^2\,\sigma_1 & 0 \\ 0 & 0 & 1 & 0\ 0 \\ 0 & 0 & 0 & 1\ 0 \\ 0 & 0 & 0 & 0\ 1 \end{bmatrix} \quad Q_4 := \begin{bmatrix} 1 & \eta_1 - d + \eta_2 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

We now compute bases of the free A-modules $\ker_A(.P_4)$, $\ker_A(.Q_4)$,, $\mathrm{im}_A(.P_4) = \ker_A(.(I_5 - P_4))$, and $\mathrm{im}_A(.Q_4) = \ker_A(.(I_3 - Q_4))$ and we get the following two unimodular matrices X and Y:

```
> X1:=SyzygyModule(P[4],A):
> X2:=SyzygyModule(evalm(1-P[4]),A):
> X:=stackmatrix(X1,X2);
> Y1:=SyzygyModule(Q[4],A):
> Y2:=SyzygyModule(evalm(1-Q[4]),A):
> Y:=stackmatrix(Y1,Y2);
```

$$X := \begin{bmatrix} {\sigma_1}^2-1 & -{\sigma_1}^2 & -{\sigma_1}^2\,\sigma_1 & 0 \\ 1 & -1 & 0 & 0\ 0 \\ 0 & 0 & 1 & 0\ 0 \\ 0 & 0 & 0 & 1\ 0 \\ 0 & 0 & 0 & 0\ 1 \end{bmatrix} \quad Y := \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & d-\eta_2-\eta_1 \\ 0 & 1 & 1 \end{bmatrix}$$

Then, we obtain the following decomposition $Y\,S\,X^{-1}$ of the matrix S:

```
> S_dec:=Mult(Y,S,LeftInverse(X,A),A);
```

$$S\_dec := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2\,\eta_1 & -d-\eta_2-\eta_1+{\sigma_2}^2\,d-\eta_2\,{\sigma_2}^2-{\sigma_2}^2\,\eta_1 & 0 & (\eta_1-d+\eta_2)\,\sigma_2 \\ 0 & {\sigma_1}^2-1 & -{\sigma_1}^2+{\sigma_2}^2 & \sigma_1 & -\sigma_2 \end{bmatrix}$$

We continue by considering the second diagonal block matrix T of S_dec:

```
> T:=submatrix(S_dec,2..3,2..5):
```

We apply the same technique as above:

```
> P[5]:=diag(1$4): Q[5]:=diag(1$2):
> Z[5]:=matrix(4,2,[0$8]):
```

We compute the solutions of the Riccati equation $\Lambda\,T\,\Lambda + \Lambda = 0$:

```
> Mu1:=RiccatiConstCoeff(T,P[5],Q[5],Z[5],A,0,alpha):
```

We choose one solution Λ_2 of the previous algebraic Riccati equation:

```
> Lambda[2]:=subs({b311=0},Mu1[1,1]);
```

$$\Lambda_2 := \begin{bmatrix} -1/(2\,\eta_1) & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Hence, we get an idempotent of the endomorphism ring of the A-module finitely presented by T defined by the following matrices P_6 and Q_6:

```
> P[6]:=simplify(evalm(P[5]+Mult(Lambda[2],T,A)));
> Q[6]:=simplify(evalm(Q[5]+Mult(T,Lambda[2],A)));
```

$$P_6 = \begin{bmatrix} 0 & 1/2\,\frac{\eta_1+\eta_2+d-\sigma_2{}^2\,d+\eta_2\,\sigma_2{}^2+\sigma_2{}^2\eta_1}{\eta_1} & 0 & -1/2\,\frac{(\eta_1-d+\eta_2)\sigma_2}{\eta_1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q_6 = \begin{bmatrix} 0 & 0 \\ -1/2\,\frac{\sigma_1{}^2-1}{\eta_1} & 1 \end{bmatrix}$$

We now compute bases of the free A-modules $\ker_A(.P_6)$, $\ker_A(.Q_6)$, $\operatorname{im}_A(.P_6) = \ker_A(.(I_4 - P_6))$ and $\operatorname{im}_A(.Q_6) = \ker_A(.(I_2 - Q_6))$ and we obtain the following unimodular matrices G and H:

```
> G1:=SyzygyModule(P[6],A):
> G2:=SyzygyModule(evalm(1-P[6]),A):
> G:=stackmatrix(G1,G2);
> H1:=SyzygyModule(Q[6],A):
> H2:=SyzygyModule(evalm(1-Q[6]),A):
> H:=stackmatrix(H1,H2);
```

$$G := \begin{bmatrix} 2\,\eta_1 & -d-\eta_2-\eta_1+\sigma_2{}^2\,d-\eta_2\,\sigma_2{}^2-\sigma_2{}^2\eta_1 & 0 & \sigma_2\,\eta_1-\sigma_2\,d+\eta_2\,\sigma_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H := \begin{bmatrix} 1 & 0 \\ \sigma_1{}^2-1 & -2\,\eta_1 \end{bmatrix}$$

Then, we obtain the following decomposition $H\,T\,G^{-1}$ of the matrix T:

```
> T_dec:=Mult(H,T,LeftInverse(G,A),A);
```

$$T\_dec := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \left((-\eta_1 + d - \eta_2)\,\sigma_2{}^2 + \eta_1 - d - \eta_2\right)\sigma_1{}^2 + (-d - \eta_1 + \eta_2)\,\sigma_2{}^2 + d + \eta_2 + \eta_1 & -2\,\eta_1\,\sigma_1 & (\eta_1 - d + \eta_2)\,\sigma_2\,\sigma_1{}^2 + (d - \eta_2 + \eta_1)\,\sigma_2 \end{bmatrix}$$

From the previous three invertible transformations, we can deduce the unimodular matrices that perform all this decomposition process in one step:

```
> W[1]:=Mult(diag(1,1,G),diag(1,X),U,A):
> W[2]:=Mult(diag(1,1,H),diag(1,Y),V,A):
```

The system matrix R is equivalent to the matrix $L = W_2\,R\,W_1^{-1}$.

```
> L:=Mult(W[2],R,LeftInverse(W[1],A),A)):
```

The matrix L has then the form

```
> ShapeOfMatrix(L);
```

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

where the stars * denote non-trivial elements of A respectively defined by:

```
> collect(L[4,4],{d,sigma[1],sigma[2]});
```

$$\left((-\eta_1 + d - \eta_2)\,\sigma_2{}^2 + \eta_1 - d - \eta_2\right)\sigma_1{}^2 + (-d - \eta_1 + \eta_2)\,\sigma_2{}^2 + d + \eta_2 + \eta_1$$

```
> collect(L[4,5],{d,sigma[1],sigma[2]});
```

$$-2\,\eta_1\,\sigma_1$$

```
> collect(L[4,6],{d,sigma[1],sigma[2]});
```

$$(\eta_1 - d + \eta_2)\,\sigma_2\,\sigma_1{}^2 + (d - \eta_2 + \eta_1)\,\sigma_2$$

The entries of the last row of L can be reduced by means of elementary column operations. Hence, if we consider the following unimodular matrix

$$J :=$$
$$\begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & {\sigma_1}^2 - 1 & -{\sigma_1}^2 & -{\sigma_1}^2 & \sigma_1 & 0 \\ 0 & 2\,\eta_1 & -2\,\eta_1 & -\eta_1 - \eta_2 - d + {\sigma_2}^2\,d - {\sigma_2}^2\,\eta_1 - \eta_2\,{\sigma_2}^2 & 0 & -\left(d - \eta_1 - \eta_2\right)\sigma_2 \\ 0 & 0 & 0 & 1 - {\sigma_2}^2 & 0 & \sigma_2 \\ 0 & 0 & 0 & \sigma_1\left({\sigma_2}^2\,d - {\sigma_2}^2\,\eta_1 - \eta_2\,{\sigma_2}^2 - d - \eta_2 + \eta_1\right) & -2\,\eta_1 & -\sigma_2\,\sigma_1\left(d - \eta_1 - \eta_2\right) \\ 0 & 0 & 0 & 2\,\sigma_2\,\eta_2 & 0 & -2\,\eta_2 \end{bmatrix}$$

obtained from W_1 by means of elementary operations (see [7]), we finally get the following simpler decomposition $W_2\,R\,J^{-1}$ of R:

```
>  R_final:=Mult(W[2],R,LeftInverse(J,A),A);
```

$$R\_final\ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & d + \eta_1 + \eta_2 & \sigma_1 & \sigma_2 \end{bmatrix}$$

Hence, the differential time-delay system (2) formed by 4 equations in 6 unknowns is equivalent to the following sole equation in 3 unknowns:

$$\dot{x}_1(t) + (\eta_1 + \eta_2)\,x_1(t) + x_2(t - h_1) + x_3(t - h_2) = 0. \tag{3}$$

Using the simple form of (3), we can easily study its structural properties (e.g., controllability, parametrizability, flatness, π-freeness, stability, stabilizability), and thus, those of (2). In particular, we obtain that (3), and thus, (2) is controllable, parametrizable, σ_1-free and σ_2-free (see [3, 15] for the corresponding definitions). Parametrizations of (2) can directly be obtained from the ones of (3) by means of the matrix J^{-1} ([3, 4, 15, 17]). System (3) admits an unstable pole at $-(\eta_1 + \eta_2)$, where the η_i's are two positive parameters of (2). Its stabilizability can be studied using, e.g., [17, Proposition 3.8].

3 Others systems decomposed with OreMorphisms

In Table 2, we gather a list of different kinds of systems appearing in control theory, mathematical physics and engineering sciences that we have decomposed using OreMorphisms. We give the system matrix R, the unimodular matrices U and V and the decomposed equivalent matrix $\overline{R} = V\,R\,U^{-1}$. For more examples, we refer the reader to the OreMorphisms web-pages ([7]).

Table 2. List of examples computed with OreMorphisms

$\overline{R} = V\,R\,U^{-1}$	U & V
[16] $V \begin{pmatrix} \partial & -\partial\,\delta^2 & \alpha\,\partial^2\,\delta \\ \partial\,\delta^2 & -\partial & \alpha\,\partial^2\,\delta \end{pmatrix} U^{-1}$ $=$ $\begin{pmatrix} \partial\,(1-\delta)\,(1+\delta) & 0 & 0 \\ 0 & \partial\,(\delta^2+1) & 2\,\alpha\,\partial^2\,\delta \end{pmatrix}$	$U = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $V = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$
[16] $V \begin{pmatrix} \partial & -\partial\,\delta^2 & \alpha\,\partial^2\,\delta \\ \partial\,\delta^2 & -\partial & \alpha\,\partial^2\,\delta \end{pmatrix} U^{-1}$ $=$ $\begin{pmatrix} \partial & 0 & 0 \\ 0 & \partial\,(\delta^4-1) & \alpha\,\partial^2\,(\delta^3-\delta) \end{pmatrix}$	$U = \begin{pmatrix} \delta^2 & -1 & \alpha\,\partial\,\delta \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $V = \begin{pmatrix} 0 & 1 \\ -1 & \delta^2 \end{pmatrix}$
$V \begin{pmatrix} \partial & -\partial\,\delta & -1 \\ 2\,\partial\,\delta & -\partial\,(\delta^2+1) & 0 \end{pmatrix} U^{-1}$ $=$ $\begin{pmatrix} \partial & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ [14]	$U = \begin{pmatrix} -2\,\delta & \delta^2+1 & 0 \\ 2\,\partial\,(1-\delta^2) & \partial\,\delta\,(\delta^2-1) & -2 \\ -1 & \delta/2 & 0 \end{pmatrix}$ $V = \begin{pmatrix} 0 & -1 \\ 2 & -\delta \end{pmatrix}$
$\begin{pmatrix} \partial + \frac{1}{2\,\theta} & 0 & -1 & -1 \\ 0 & \partial + \frac{1}{\theta} & -\frac{d_1}{V_0}\,\delta & -\frac{d_2}{V_0}\,\delta \end{pmatrix}$ $=$ $V^{-1} \begin{pmatrix} \partial + \frac{1}{\theta} & -\frac{1}{V_0}\,\delta & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} U$ [12]	$U = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & d_1 & d_2 \\ \partial + \frac{1}{2\,\theta} & 0 & -1 & -1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$ $V = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
$\begin{pmatrix} \partial + a & -k\,a\,\delta & 0 & 0 \\ 0 & \partial & -1 & 0 \\ 0 & \omega^2 & \partial + 2\,\zeta\,\omega & -\omega^2 \end{pmatrix}$ $= V^{-1} \begin{pmatrix} \partial + a & -a\,k\,\omega^2\,\delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} U$ [13]	$U =$ $\begin{pmatrix} \omega^2 & \partial & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \omega^2\,(\partial + a) & -\omega^2\,(k\,a\,\delta + 1) & -(\partial + 2\,\omega\,\zeta\,) & \omega^2 \\ 0 & \partial & -1 & 0 \end{pmatrix}$ $V = \begin{pmatrix} \omega^2 & \partial + a & 0 \\ \omega^2 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$

$\overline{R} = V\, R\, U^{-1}$	U & V
$V \begin{pmatrix} \partial_x & a\,\partial_t \\ \partial_t & b\,\partial_x \end{pmatrix} U^{-1}$ $= \begin{pmatrix} \partial_t - 2\,\alpha\,b\,\partial_x & 0 \\ 0 & \partial_t + 2\,\alpha\,b\,\partial_x \end{pmatrix}$ [8]	$U = \frac{1}{2\,\alpha} \begin{pmatrix} -2\,\alpha & 1 \\ 2\,\alpha & 1 \end{pmatrix}$ $V = \begin{pmatrix} 2\,b\,\alpha & -1 \\ 2\,b\,\alpha & 1 \end{pmatrix}$ $4\,a\,b\,\alpha^2 - 1 = 0$
$V \begin{pmatrix} \partial_x & L\,\partial_t + R \\ C\,\partial_t + G & \partial_x \end{pmatrix} U^{-1}$ $=$ $\begin{pmatrix} 1 & 0 \\ 0 & (R + L\,\partial_t)\,(G + C\,\partial_t) - \partial_x^2 \end{pmatrix}$ [8]	$U =$ $\begin{pmatrix} C\,(\partial_x - \alpha\,\partial_t) - \alpha\,G & C\,(L\,\partial_t + R) - \alpha\,\partial_x \\ C & -\alpha \end{pmatrix}$ $V =$ $\begin{pmatrix} C & -\alpha \\ -C\,(\partial_x + \alpha\,\partial_t) - \alpha\,G & \alpha\,\partial_x + C\,(L\,\partial_t + R) \end{pmatrix}$ $\alpha^2 - L\,C = 0$
$V \begin{pmatrix} u\,\rho\,\partial_x & c^2\,\partial_x & 0 \\ 0 & c^2\,\partial_y & u\,\rho\,\partial_x \\ \rho\,\partial_x & u\,\partial_x & \rho\,\partial_y \end{pmatrix} U^{-1}$ $=$ $\begin{pmatrix} \partial_x - 2\,\alpha\,c\,\partial_y & 0 & 0 \\ 0 & \partial_x + 2\,\alpha\,c\,\partial_y & 0 \\ 0 & 0 & \partial_x \end{pmatrix}$ [8]	$U = \begin{pmatrix} 0 & 2\,\alpha\,c\,(c^2 - u^2) & u\,\rho \\ 0 & 2\,\alpha\,c\,(c^2 - u^2) & -u\,\rho \\ u\,\rho & c^2 & 0 \end{pmatrix}$ $V = \begin{pmatrix} 2\,\alpha\,c & 1 & -2\,\alpha\,c\,u \\ 2\,\alpha\,c & -1 & -2\,\alpha\,c\,u \\ 1 & 0 & 0 \end{pmatrix}$ $1 + 4\,(c^2 - u^2)\,\alpha^2 = 0$
$\begin{pmatrix} \partial_t & 0 & -i\,\partial_3 & -(i\,\partial_1 + \partial_2) \\ 0 & \partial_t & -i\,\partial_1 + \partial_2 & i\,\partial_3 \\ i\,\partial_3 & i\,\partial_1 + \partial_2 & -\partial_t & 0 \\ i\,\partial_1 - \partial_2 & -i\,\partial_3 & 0 & -\partial_t \end{pmatrix}$ $= V^{-1}$ $\begin{pmatrix} i\,\partial_3 - \partial_t & -i\,\partial_1 - \partial_2 & 0 & 0 \\ i\,\partial_1 - \partial_2 & i\,\partial_3 + \partial_t & 0 & 0 \\ 0 & 0 & i\,\partial_3 + \partial_t & -i\,\partial_1 - \partial_2 \\ 0 & 0 & -i\,\partial_1 + \partial_2 & -i\,\partial_3 + \partial_t \end{pmatrix}$ U [8]	$U = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$ $V = -\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 \end{pmatrix}$

Conclusion

In this paper, we have demonstrated the homological algebraic package OreMorphisms dedicated to the factorization, reduction and decomposition problems of linear functional systems. The increasing role of homological algebra in mathematical systems theory, mathematical physics and other fields has recently motivated the development of packages based on more and more powerful homological algebraic techniques as, for instance, OreModules ([2, 4]), OreMorphisms and homalg ([1]). We are convinced that this phenomenon is a precursory sign of a new era where computer algebra and symbolic computation will play the equivalent role for pure mathematics as the one played by numerical analysis in applied mathematics and engineering sciences.

References

1. M. Barakat, D. Robertz, "An abstract package for homological algebra", *J. Algebra Appl.*, 7 (2008), 299-317 (`http://wwwb.math.rwth-aachen.de/homalg/`).
2. F. Chyzak, A. Quadrat, D. Robertz, OreModules project (`http://wwwb.math.rwth-aachen.de/OreModules`).
3. F. Chyzak, A. Quadrat, D. Robertz, "Effective algorithms for parametrizing linear control systems over Ore algebras", *Appl. Algebra Engrg. Comm. Comput.*, 16 (2005), 319-376.
4. F. Chyzak, A. Quadrat, D. Robertz, "OreModules: A symbolic package for the study of multidimensional linear systems", in *Applications of Time-Delay Systems*, J. Chiasson and J.-J. Loiseau (Eds.), Lecture Notes in Control and Inform. Sci. (LNCIS) 352, Springer, 2007, 233-264.
5. T. Cluzeau, A. Quadrat, "Factoring and decomposing a class of linear functional systems", *Linear Algebra Appl.*, 428 (2008), 324-381.
6. T. Cluzeau, A. Quadrat, "On algebraic simplifications of linear functional systems", in *Topics in Time-Delay Systems: Analysis, Algorithms and Control*, J.-J. Loiseau, W. Michiels, S.-I. Niculescu, R. Sipahi (Eds.), Lecture Notes in Control and Inform. Sci. (LNCIS), Springer (2009).
7. T. Cluzeau, A. Quadrat, OreMorphisms project (`http://www.ensil.unilim.fr/~cluzeau/`, `http://www-sop.inria.fr/personnel/Alban.Quadrat/index.html`).
8. R. Courant, D. Hilbert, *Methods of Mathematical Physics*, Wiley Classics Library, Wiley, 1989.
9. G. Culianez, *Formes de Hermite et de Jacobson: Implémentations et applications*, Internship with A. Quadrat, INRIA Sophia Antipolis, 2005.
10. A. Fabiańska, QuillenSuslin project (`http://wwwb.math.rwth-aachen.de/QuillenSuslin`).
11. A. Fabiańska, A. Quadrat, "Applications of the Quillen-Suslin theorem in multidimensional systems theory", in *Gröbner Bases in Control Theory and Signal Processing*, H. Park and G. Regensburger (Eds.), Radon Series on Computation and Applied Mathematics 3, de Gruyter publisher, 2007, 23-106.
12. H. Kwakernaak, R. Sivan, *Linear Optimal Control Systems*, Wiley, 1972.

13. A. Manitius, "Feedback controllers for a wind tunnel model involving a delay: analytical design and numerical simulations", *IEEE Trans. Autom. Contr.*, 29 (1984), 1058-1068.
14. H. Mounier, J. Rudolph, M. Petitot, M. Fliess, "A flexible rod as a linear delay system", Proceedings of *European Control Conference (ECC)*, Rome, 1995.
15. H. Mounier, J. Rudolph, M. Fliess, P. Rouchon, "Tracking control of a vibrating string with an interior mass viewed as delay system", *ESAIM Control Optim. Calc. Var.*, 3 (1998), 315-321.
16. N. Petit, P. Rouchon, "Dynamics and solutions to some control problems for water-tank systems", *IEEE Trans. Automatic Control*, 47 (2002), 595-609.
17. A. Quadrat, "The fractional representation approach to synthesis problems: an algebraic analysis viewpoint. Part II: Internal stabilization", *SIAM J. Control & Optimization*, 42 (2003), 300-320.
18. A. Quadrat, D. Robertz, "Computation of bases of free modules over the Weyl algebras", *Journal Symbolic Comput.*, 42 (2007), 1113-1141.