

Two pendula mounted on a cart (Polderman, Willems)

Definition of Ore algebra and ODE system:

```
> with(OreModules):
> Alg := diff_algebra([Dt,t], polynom={t}, comm={m1, m2, M, L1, L2, g}):
> R := evalm([[m1*L1*Dt^2, m2*L2*Dt^2, -1, (M+m1+m2)*Dt^2],
>             [m1*L1^2*Dt^2-m1*L1*g, 0, 0, m1*L1*Dt^2], [0, m2*L2^2*Dt^2-m2*L2*g, 0, m2*L2*Dt^2]]);
```

$$R := \begin{bmatrix} m1 L1 Dt^2 & m2 L2 Dt^2 & -1 & (M + m1 + m2) Dt^2 \\ m1 L1^2 Dt^2 - m1 L1 g & 0 & 0 & m1 L1 Dt^2 \\ 0 & m2 L2^2 Dt^2 - m2 L2 g & 0 & m2 L2 Dt^2 \end{bmatrix}$$

Computing $\text{ext}_D^1(N, D)$, where $D = Alg$: (CPU: < 1s)

```
> Ext1 := exti(adjoint(R, Alg), Alg, 1): Ext1[1];
```

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

$\iff \text{ext}_D^1(N, D) \cong t(M) = \{0\}$.

\iff System is controllable \iff System is parametrizable.

`exti` also constructs a parametrization of the system:

```
> Ext1[3];
```

$$\left[\begin{array}{c} gDt^2 - L2Dt^4 \\ gDt^2 - Dt^4L1 \\ -Dt^4gL1m2 - L2Dt^4gm1 - L2Dt^4gM + L2Dt^6L1M + g^2m2Dt^2 - gL1MDt^4 + g^2m1Dt^2 + g^2MDt^2 \\ -L2Dt^2g + L2Dt^4L1 - gL1Dt^2 + g^2 \end{array} \right]$$

If this parametrization admits a left inverse, we can construct a *flat output*:

```
> L := LI(Ext1[3], Alg);
```

$$L := \left[\begin{array}{cccc} \frac{L1^2}{g^2(L1 - L2)} & -\frac{L2^2}{g^2(L1 - L2)} & 0 & -\frac{-L1 + L2}{g^2(L1 - L2)} \end{array} \right]$$

Now

$$L \cdot (w_1, w_2, w_3, w_4)^T = \frac{L1^2 w_1 - L2^2 w_2 + (L1 - L2) w_4}{g^2 (L1 - L2)}$$

is a flat output.

Linear DAE

Definition of Ore algebra and ODE system:

```
> with(OreModules):
> Alg := diff_algebra([D[t],t], polynom={t}):
> R := evalm([[ -t*D[t]+1, t^2*D[t], -1, 0], [-D[t], t*D[t]+1, 0, -1]]);
```

$$R := \begin{bmatrix} -t D_t + 1 & t^2 D_t & -1 & 0 \\ -D_t & t D_t + 1 & 0 & -1 \end{bmatrix}$$

```
> Ext1 := exti(adjoint(R, Alg), Alg, 1);
```

$$\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & D_t & -t D_t \\ 1 & -t & -1 & t \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ t^2 D_t & -t D_t + 1 \\ t D_t + 1 & -D_t \end{bmatrix}, \begin{bmatrix} 0 & 1 & -t^2 D_t - 2t & -t D_t \\ 1 & 0 & t D_t + 2 & D_t \end{bmatrix} \right]$$

```
> mult(R, Ext1[3], Alg);
```

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

```
> LI(Ext1[3], Alg);
```

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

Wind tunnel model (Manitius)

Definition of Ore algebra and differential time-delay system:

```
> with(OreModules):
> Alg := diff_algebra([D[1],x[1]], [delta,x[2]],
>   polynom={x[1],x[2]}, comm={a, omega, zeta, k}):
> R := evalm([[D[1]+a,-k*a*delta,0,0], [0,D[1],-1,0],
>   [0,omega^2,D[1]+2*zeta*omega,-omega^2]]);
```

$$R := \begin{bmatrix} D_1 + a & -k a \delta & 0 & 0 \\ 0 & D_1 & -1 & 0 \\ 0 & \omega^2 & D_1 + 2 \zeta \omega & -\omega^2 \end{bmatrix}$$

Computing $\text{ext}_D^1(N, D)$, where $D = \text{Alg}$: (CPU: < 1s)

```
> Ext1 := exti(adjoint(R, Alg), Alg, 1): Ext[1];
```

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

$$\iff \text{ext}_D^1(N, D) \cong t(M) = \{0\}.$$

$$\iff \text{System is controllable} \iff \text{parametrizable}.$$

exti also constructs a parametrization of the system:

```
> Ext1[3];
```

$$\begin{bmatrix} -\omega^2 k a \delta \\ -D_1 \omega^2 - a \omega^2 \\ -\omega^2 D_1^2 - \omega^2 a D_1 \\ -D_1^3 - 2 D_1^2 \zeta \omega - a D_1^2 - D_1 \omega^2 - 2 a D_1 \zeta \omega - a \omega^2 \end{bmatrix}$$

Computing $\text{ext}_D^2(N, D)$, where $D = \text{Alg}$: (CPU: $< 1s$)

```
> Ext2 := exti(adjoint(R, Alg), Alg, 2): Ext2[1];
```

$$\begin{bmatrix} \delta \\ D_1 + a \end{bmatrix}$$

We find: $\text{ext}_D^2(N, D) \neq \{0\}$.

$\left. \begin{array}{l} \text{ext}_D^1(N, D) = \{0\} \\ \text{ext}_D^2(N, D) \neq \{0\} \end{array} \right\} \implies$ System controllable, but neither reflexive, nor projective, nor free (which are all equivalent in this case).

Hence, we find a polynomial $\pi(\delta)$ such that the system is π -flat:

```
> pi_polynomial(adjoint(R, Alg), Alg, [delta]);
```

$$[\delta]$$

A two reflector antenna (Mounier)

Definition of Ore algebra and differential delay system:

```

> with(OreModules):
> Alg:=diff_algebra([D[1],x[1]], [delta,x[2]], polynom={x[1],x[2]}, comm={K1,K2,Te,Kp,Kc}):
> R := evalm([[D[1],-K1,0$7], [0,D[1]+K2/Te,0$4,-Kp/Te*delta,-Kc/Te*delta,-Kc/Te*delta],
>           [0,0,D[1],-K1,0$5], [0$3,D[1]+K2/Te,0,0,-Kc/Te*delta,-Kp/Te*delta,-Kc/Te*delta],
>           [0$4,D[1],-K1,0$3], [0$5,D[1]+K2/Te,-Kc/Te*delta,-Kc/Te*delta,-Kp/Te*delta]]);

```

$$R := \begin{bmatrix}
D_1 & -K_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & D_1 + \frac{K_2}{T_e} & 0 & 0 & 0 & 0 & -\frac{K_p \delta}{T_e} & -\frac{K_c \delta}{T_e} & -\frac{K_c \delta}{T_e} \\
0 & 0 & D_1 & -K_1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & D_1 + \frac{K_2}{T_e} & 0 & 0 & -\frac{K_c \delta}{T_e} & -\frac{K_p \delta}{T_e} & -\frac{K_c \delta}{T_e} \\
0 & 0 & 0 & 0 & D_1 & -K_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & D_1 + \frac{K_2}{T_e} & -\frac{K_c \delta}{T_e} & -\frac{K_c \delta}{T_e} & -\frac{K_p \delta}{T_e}
\end{bmatrix}$$

Computing $\text{ext}_D^1(N, D)$, where $D = \text{Alg}$: (CPU: 1s)

```
> Ext1 := exti(adjoint(R, Alg), Alg, 1): Ext1[1];
```

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

$\iff \text{ext}_D^1(N, D) \cong t(M) = \{0\}$.

\iff System is controllable

\iff System is parametrizable.

`exti` also constructs a parametrization of the system:

```
> Ext1[3];
```

$$\begin{bmatrix} 0 & 0 & \%4 \\ 0 & 0 & \%3 \\ 0 & \%4 & 0 \\ 0 & \%3 & 0 \\ \%4 & 0 & 0 \\ \%3 & 0 & 0 \\ \%1 & \%1 & \%2 \\ \%1 & \%2 & \%1 \\ \%2 & \%1 & \%1 \end{bmatrix} \quad \begin{array}{l} \%1 := Te D_1^2 Kc + K2 D_1 Kc \\ \%2 := -Kp D_1^2 Te - Kp D_1 K2 \\ - Te D_1^2 Kc - K2 D_1 Kc \\ \%3 := -Kp^2 D_1 \delta - Kp Kc \delta D_1 \\ + 2 \delta Kc^2 D_1 \\ \%4 := 2 \delta Kc^2 K1 - Kp Kc \delta K1 \\ - Kp^2 K1 \delta \end{array}$$

Computing $\text{ext}_D^2(N, D)$, where $D = \text{Alg}$: (CPU: 1s)

```
> Ext2 := exti(adjoint(R, Alg), Alg, 2): Ext2[1];
```

$$\begin{bmatrix} \delta & 0 & 0 \\ Te D_1^2 + K2 D_1 & 0 & 0 \\ 0 & \delta & 0 \\ 0 & Te D_1^2 + K2 D_1 & 0 \\ 0 & 0 & \delta \\ 0 & 0 & Te D_1^2 + K2 D_1 \end{bmatrix}$$

We find: $\text{ext}_D^2(N, D) \neq \{0\}$.

$\left. \begin{array}{l} \text{ext}_D^1(N, D) = \{0\} \\ \text{ext}_D^2(N, D) \neq \{0\} \end{array} \right\} \implies$ System controllable, but neither reflexive, nor projective, nor free (which are all equivalent in this case).

Hence, we find a polynomial $\pi(\delta)$ such that the system is π -flat:

```
> pi_polynomial(adjoint(R, Alg), Alg, [delta]);
```

$[\delta]$

Flexible Rod (Mounier)

Definition of Ore algebra and differential time-delay system:

```
> with(OreModules):
> Alg := diff_algebra([D[1],x[1]], [delta,x[2]], polynom={x[1],x[2]}):
> R := evalm([[D[1], -D[1]*delta, -1], [2*D[1]*delta, -D[1]-D[1]*delta^2, 0]]);
```

$$R := \begin{bmatrix} D_1 & -D_1 \delta & -1 \\ 2 D_1 \delta & -D_1 - D_1 \delta^2 & 0 \end{bmatrix}$$

Computing $\text{ext}_D^1(N, D)$, where $D = Alg$: (CPU: < 1s)

```
> exti(adjoint(R, Alg), Alg, 1);
```

$$\left[\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & D_1 \end{array} \right], \left[\begin{array}{ccc} -D_1 & D_1 \delta & 1 \\ D_1 \delta & -D_1 & \delta \\ -2 \delta & 1 + \delta^2 & 0 \end{array} \right], \left[\begin{array}{c} 1 + \delta^2 \\ 2 \delta \\ D_1 - D_1 \delta^2 \end{array} \right], \left[\begin{array}{ccc} 1 + \delta^2 & -2 \delta & -D_1 + D_1 \delta^2 \end{array} \right] \right]$$

We find: $t(M) \cong \text{ext}_D^1(N, D) \neq \{0\}$.

E.g. $m := (-2 \delta) y_1 + (1 + \delta^2) y_2 \in M$ is torsion element: $D_1 m = 0$.

An electric transmission line (Mounier)

Definition of Ore algebra and differential delay system:

```

> with(OreModules):
> Alg := diff_algebra([Dt,t], [delta,s], polynom={t,s},
>   comm={a[0],a[1],a[2],a[3],a[4],a[5],b[0]}):
> R := evalm([[Dt+a[0],-(a[4]*Dt+a[0])*delta,-a[0],0,-b[0]*Dt],
>   [-delta*(a[5]*Dt+a[1]),Dt+a[1],0,a[1],0],
>   [a[2],-a[2]*a[4]*delta,Dt,0,-a[2]*b[0]],
>   [a[3]*a[5]*delta,-a[3],0,Dt,0]]);

```

$$R := \begin{bmatrix} Dt + a_0 & -(a_4 Dt + a_0) \delta & -a_0 & 0 & -b_0 Dt \\ -\delta (a_5 Dt + a_1) & Dt + a_1 & 0 & a_1 & 0 \\ a_2 & -a_2 a_4 \delta & Dt & 0 & -a_2 b_0 \\ a_3 a_5 \delta & -a_3 & 0 & Dt & 0 \end{bmatrix}$$

Computing $\text{ext}_D^1(N, D)$, where $D = Alg$: (CPU: $\approx 10s$)

```

> Ext1 := exti(adjoint(R, Alg), Alg, 1): Ext1[1];

```

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

$$\iff \text{ext}_D^1(N, D) \cong t(M) = \{0\}.$$

$$\iff \text{System is controllable} \iff \text{parametrizable.}$$

`exti` also constructs a parametrization of the system:

```
> Ext1[3];
```

$$\left[\begin{array}{l} Dt^4 b_0 + a_1 b_0 Dt^3 + a_0 a_2 b_0 Dt^2 + a_1 b_0 Dt^2 a_3 + a_0 a_1 a_2 b_0 Dt + a_1 b_0 a_3 a_2 a_0 \\ a_0 a_1 a_2 Dt \delta b_0 + a_0 a_1 a_2 a_5 b_0 a_3 \delta + a_0 a_2 a_5 Dt^2 \delta b_0 + a_5 Dt^4 \delta b_0 + a_1 a_5 b_0 \delta Dt^2 a_3 + a_1 Dt^3 \delta b_0 \\ a_1 b_0 a_3 a_2 a_0 - a_0 a_1 a_2 a_5 b_0 \delta^2 a_3 + a_0 a_1 a_2 b_0 Dt - a_0 a_1 a_2 Dt \delta^2 b_0 - a_0 a_2 a_5 Dt^2 \delta^2 b_0 + a_0 a_2 b_0 Dt^2 \\ a_0 a_1 a_2 b_0 a_3 \delta - a_0 a_1 a_2 a_5 b_0 a_3 \delta + a_1 b_0 a_3 \delta Dt^2 - a_1 a_5 b_0 \delta Dt^2 a_3 \\ a_1 a_0 Dt^2 + Dt^4 + a_1 Dt^3 + a_0 Dt^3 - a_1 Dt^3 \delta^2 a_4 + a_1 a_3 Dt^2 + a_0 a_2 Dt^2 - a_0 a_1 a_2 \delta^2 Dta_4 \\ - a_0 a_1 \delta^2 Dta_3 a_5 - a_0 a_2 a_5 \delta^2 Dt^2 a_4 - a_0 Dt^3 \delta^2 a_5 + a_0 a_1 a_2 a_3 + a_0 a_1 a_2 Dt \\ + a_0 a_1 a_3 Dt - a_0 a_1 \delta^2 Dt^2 - a_0 a_1 a_2 a_5 \delta^2 a_3 a_4 - a_5 \delta^2 Dt^4 a_4 - a_1 a_5 \delta^2 Dt^2 a_3 a_4 \end{array} \right]$$

Computing $\text{ext}_D^2(N, D)$, where $D = \text{Alg}$: (CPU: 5s)

```
> Ext2 := exti(adjoint(R, Alg), Alg, 2): Ext2[1];
```

$$\left[\begin{array}{l} -Dta_0 a_1 a_2 \delta + \delta a_0 a_5 a_2 a_1 Dt - \delta a_1^2 a_5 a_3 Dt + \delta Dta_1^2 a_3 + \delta^3 a_1^2 a_2 a_0 + a_0^2 a_5^2 a_2^2 \delta^3 \\ -2 a_0 a_5^2 a_2 a_1 \delta^3 a_3 + a_1^2 a_5^2 a_3^2 \delta^3 + 2 a_1 a_5 a_3 a_0 a_2 \delta - \delta a_1^2 a_5 a_3^2 - a_1^2 \delta a_2 a_0 - a_0^2 a_5 a_2^2 \delta \\ \delta^2 Dta_1 - Dt^2 - a_5 \delta^2 a_0 a_2 + a_1 a_5 \delta^2 a_3 - a_1 Dt - a_1 a_3 \\ \delta a_2 a_0 + Dt^2 \delta \\ a_1 Dt^3 + a_1^2 Dt^2 + a_0 a_2 a_5 Dt^2 - Dt^2 a_1 a_5 a_3 + \delta^2 a_1^2 a_2 a_0 + a_0^2 a_5^2 a_2^2 \delta^2 - 2 a_0 a_5^2 a_2 a_1 \delta^2 a_3 \\ + a_1^2 a_5^2 a_3^2 \delta^2 + Dta_1^2 a_3 + a_0 a_5 a_2 a_1 Dt - a_1^2 a_5 a_3 Dt + a_0 a_2 a_5 a_1 a_3 - a_1^2 a_5 a_3^2 \end{array} \right]$$

Einstein equations (lin. Ricci equations in vacuum)

Underdetermined PDE system – is it *parametrizable*?

\iff J. Wheeler: Exist potentials for Einstein equations?

Answer (Pommaret 1995): *No.* Let us prove this by using *OreModules*:

Definition of Ore algebra and PDE system:

```
> with(OreModules):
> Alg := diff_algebra([D[1],x1], [D[2],x2], [D[3],x3], [D[4],x4], polynom={x1,x2,x3,x4}):
> R := evalm([ ... ]);
```

$$R := \begin{bmatrix} \%1 & D_1^2 & D_1^2 & -D_1^2 & -2 D_1 D_2 & 0 & 0 & -2 D_1 D_3 & 0 & 2 D_1 D_4 \\ D_2^2 & \%2 & D_2^2 & -D_2^2 & -2 D_1 D_2 & -2 D_2 D_3 & 0 & 0 & 2 D_2 D_4 & 0 \\ D_3^2 & D_3^2 & \%3 & -D_3^2 & 0 & -2 D_2 D_3 & 2 D_3 D_4 & -2 D_1 D_3 & 0 & 0 \\ D_4^2 & D_4^2 & D_4^2 & \%4 & 0 & 0 & -2 D_3 D_4 & 0 & -2 D_2 D_4 & -2 D_1 D_4 \\ 0 & 0 & D_1 D_2 & -D_1 D_2 & D_3^2 - D_4^2 & -D_1 D_3 & 0 & -D_2 D_3 & D_1 D_4 & D_2 D_4 \\ D_2 D_3 & 0 & 0 & -D_2 D_3 & -D_1 D_3 & D_1^2 - D_4^2 & D_2 D_4 & -D_1 D_2 & D_3 D_4 & 0 \\ D_3 D_4 & D_3 D_4 & 0 & 0 & 0 & -D_2 D_4 & D_1^2 + D_2^2 & -D_1 D_4 & -D_2 D_3 & -D_1 D_3 \\ 0 & D_1 D_3 & 0 & -D_1 D_3 & -D_2 D_3 & -D_1 D_2 & D_1 D_4 & D_2^2 - D_4^2 & 0 & D_3 D_4 \\ D_2 D_4 & 0 & D_2 D_4 & 0 & -D_1 D_4 & -D_3 D_4 & -D_2 D_3 & 0 & D_1^2 + D_3^2 & -D_1 D_2 \\ 0 & D_1 D_4 & D_1 D_4 & 0 & -D_2 D_4 & 0 & -D_1 D_3 & -D_3 D_4 & -D_1 D_2 & D_2^2 + D_3^2 \end{bmatrix}$$

$$\%1 := D_2^2 + D_3^2 - D_4^2, \quad \%2 := D_1^2 + D_3^2 - D_4^2, \quad \%3 := D_1^2 + D_2^2 - D_4^2, \quad \%4 := D_1^2 + D_2^2 + D_3^2$$

Computing $\text{ext}_D^1(N, D)$, where $D = \text{Alg}$: (CPU: 22s)

> `Ext1 := exti(adjoint(R, Alg), Alg, 1): Ext1[1];`

$$\underbrace{\begin{bmatrix} \%1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \%1 \end{bmatrix}}_{20} \quad \%1 := D_4^2 - D_3^2 - D_2^2 - D_1^2$$

We find: $t(M) \cong \text{ext}_D^1(N, D) \neq \{0\}$.

More precisely, $t(M)$ is generated by 20 torsion elements, and these satisfy the *Dalambertian equation* $(\Delta - c^2 \frac{\partial^2}{\partial t^2}) y = 0$.

\Rightarrow The linearized Ricci equations cannot be parametrized.

Of course, $M/t(M)$ can be parametrized:

> `Ext1[3];`

$$\begin{bmatrix} -2D_1 & 0 & 0 & 0 \\ 0 & 0 & -2D_2 & 0 \\ 0 & 0 & 0 & -2D_3 \\ 0 & -2D_4 & 0 & 0 \\ -D_2 & 0 & -D_1 & 0 \\ 0 & 0 & -D_3 & -D_2 \\ 0 & -D_3 & 0 & -D_4 \\ -D_3 & 0 & 0 & -D_1 \\ 0 & -D_2 & -D_4 & 0 \\ -D_4 & -D_1 & 0 & 0 \end{bmatrix}$$

exti also gives a generating set for $t(M)$:

> Ext1[2];

$$\begin{bmatrix}
 D_2^2 & D_1^2 & 0 & 0 & -2D_1D_2 & 0 & 0 & 0 & 0 & 0 \\
 -D_2D_4 & 0 & 0 & 0 & D_1D_4 & 0 & 0 & 0 & -D_1^2 & D_1D_2 \\
 0 & D_1D_4 & 0 & 0 & -D_2D_4 & 0 & 0 & 0 & -D_1D_2 & D_2^2 \\
 0 & 0 & 0 & 0 & 0 & -D_1D_4 & 0 & D_2D_4 & D_1D_3 & -D_2D_3 \\
 -D_2D_3 & 0 & 0 & 0 & D_1D_3 & -D_1^2 & 0 & D_1D_2 & 0 & 0 \\
 0 & D_1D_3 & 0 & 0 & -D_2D_3 & -D_1D_2 & 0 & D_2^2 & 0 & 0 \\
 0 & 0 & 0 & D_1D_2 & D_4^2 & 0 & 0 & 0 & -D_1D_4 & -D_2D_4 \\
 0 & D_4^2 & 0 & D_2^2 & 0 & 0 & 0 & 0 & -2D_2D_4 & 0 \\
 D_4^2 & 0 & 0 & D_1^2 & 0 & 0 & 0 & 0 & 0 & -2D_1D_4 \\
 0 & 0 & 0 & -D_1D_3 & 0 & 0 & D_1D_4 & -D_4^2 & 0 & D_3D_4 \\
 0 & 0 & 0 & -D_2D_3 & 0 & -D_4^2 & D_2D_4 & 0 & D_3D_4 & 0 \\
 0 & 0 & 0 & 0 & D_3D_4 & -D_1D_4 & D_1D_2 & 0 & 0 & -D_2D_3 \\
 0 & D_3D_4 & 0 & 0 & 0 & -D_2D_4 & D_2^2 & 0 & -D_2D_3 & 0 \\
 D_3D_4 & 0 & 0 & 0 & 0 & 0 & D_1^2 & -D_1D_4 & 0 & -D_1D_3 \\
 0 & 0 & D_1D_4 & 0 & 0 & 0 & -D_1D_3 & -D_3D_4 & 0 & D_3^2 \\
 0 & 0 & D_2D_4 & 0 & 0 & -D_3D_4 & -D_2D_3 & 0 & D_3^2 & 0 \\
 0 & 0 & D_1D_2 & 0 & D_3^2 & -D_1D_3 & 0 & -D_2D_3 & 0 & 0 \\
 0 & 0 & D_4^2 & D_3^2 & 0 & 0 & -2D_3D_4 & 0 & 0 & 0 \\
 0 & D_3^2 & D_2^2 & 0 & 0 & -2D_2D_3 & 0 & 0 & 0 & 0 \\
 D_3^2 & 0 & D_1^2 & 0 & 0 & 0 & 0 & -2D_1D_3 & 0 & 0
 \end{bmatrix}$$

Bibliography

- F. Chyzak, A. Quadrat, D. Robertz, **Linear control systems over Ore algebras: effective algorithms for the computation of parametrizations**, Workshop TDS03.
- *OreModules*: <http://wwwb.math.rwth-aachen.de/OreModules/>
- J. W. Polderman, J. C. Willems, **Introduction to Mathematical Systems Theory. A Behavioral Approach**, TAM 26, Springer, 1998.
- A. Manitius, **Feedback controllers for a wind tunnel model involving a delay: analytical design and numerical simulations**, IEEE Trans. Autom. Contr. vol. 29 (1984), pp. 1058-1068.
- H. Mounier, **Propriétés structurelles des systèmes linéaires à retards: aspects théoriques et pratiques**, PhD Thesis, University of Orsay, France, 1995.
- J.-F. Pommaret, **Dualité différentielle et applications**, C.R. Acad. Sci. Paris, Série I 320 (1995), pp. 1225-1230.
- G. Le Vey, **Some remarks on solvability and various indices for implicit differential equations**, Numerical Algorithms, vol. 19 (1998), pp. 127-145.