In this worksheet, we study the control of a flexible rod considered in H. Mounier, *Propriétés structurelles des systèmes linéaires à retards: aspects théoriques et pratiques*, PhD thesis, University of Orsay, France, 1995. See also H. Mounier, J. Rudolph, M. Petitot, M. Fliess, *A flexible rod as a linear delay systems*, in the proceedings of the 3rd European Control Conference, Rome (Italy), 1995.

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

We define the Ore Algebra *Alg* as follows.

```
> Alg := DefineOreAlgebra(diff=[Dt,t], dual_shift=[delta,s],
> polynom=[t,s], shift_action=[delta,t,h]):
```

We enter the matrix which defines the system of the flexible rod:

```
> R := evalm([[Dt, -Dt*delta, -1], [2*Dt*delta, -Dt-Dt*delta^2, 0]]);
```

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

Let us define the adjoint of R:

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

Compute the first extension module ext^1 with values in *Alg* of the left *Alg*-module N associated with *R_adj*:

```
> Ext := Exti(R_adj, Alg, 1);
```

$$Ext := \left[\begin{bmatrix} Dt & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -2\,\delta & 1+\delta^2 & 0 \\ -Dt & Dt\,\delta & 1 \\ Dt\,\delta & -Dt & \delta \end{bmatrix}, \begin{bmatrix} 1+\delta^2 \\ 2\,\delta \\ -Dt\,\delta^2 + Dt \end{bmatrix}\right]$$

The torsion submodule $\mathrm{t}(M)$ of the module M, which is associated with R, is generated by the row of *Ext*[1] with first entry Dt. That means that r := [-2 δ, 1 + δ^2, 0] generates the torsion submodule $\mathrm{t}(M)$ of M and r is killed by the operator Dt, i.e., $Dt\, r = 0$ in M. More precisely, if we denote by $y1$, $y2$ and u the system variables, then the torsion element is defined by:

```
> TorsionElements(R, [y1(t),y2(t),u(t)], Alg);
```

$$[[\ \mathrm{D}(\theta_1)(t) = 0\ ],\ [\ \theta_1(t) = -2\,\mathrm{y1}(t-h) + \mathrm{y2}(t) + \mathrm{y2}(t-2\,h)\ ]]$$

Let us point out that *Ext*[3] is a parametrization of the torsion-free module M / $\mathrm{t}(M)$, i.e., of the controllable part of the system which is defined by *Ext*[2].

We compute the second extension module ext^2 with values in *Alg* of N:

```
> Exti(R_adj, Alg, 2);
```

$$[[\ 1\ ],\ [\ 1\ ],\ \mathrm{SURJ}(1)]$$

Hence, ext^2 is the zero module.

Let us check wether or not *Ext*[3] admits a left-inverse:

```
> L := LeftInverse(Ext[3], Alg);
```

$$L := \begin{bmatrix} 1 & -\frac{\delta}{2} & 0 \end{bmatrix}$$

```
> Mult(L, Ext[3], Alg);
```

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

Therefore, we obtain that $M \;/\; \mathrm{t}(M)$ is a free *Alg*-module and a basis of $M \;/\; \mathrm{t}(M)$ is defined by $\xi = L\,(y1 : y2 : u)^T$, where $(y1 : y2 : u)^T = Ext1[3]\; z$. In particular, ξ is a flat output of the system defined by the matrix *Ext*[2], i.e., of the controllable part of the flexible rod. More precisely, we have:

```
> xi(t)=ApplyMatrix(L, [y1(t),y2(t),u(t)], Alg)[1,1];
```

$$\xi(t) = \mathrm{y1}(t) - \frac{1}{2}\,\mathrm{y2}(t-h)$$

```
> evalm([[y1(t)],[y2(t)],[u(t)]])=ApplyMatrix(Ext[3], [xi(t)], Alg);
```

$$\begin{bmatrix} \mathrm{y1}(t) \\ \mathrm{y2}(t) \\ \mathrm{u}(t) \end{bmatrix} = \begin{bmatrix} \xi(t) + \xi(t-2\,h) \\ 2\,\xi(t-h) \\ -\mathrm{D}(\xi)(t-2\,h) + \mathrm{D}(\xi)(t) \end{bmatrix}$$

Let us compute a free resolution of the module over *Alg* which is generated by the rows of *Ext*[3]:

```
> FreeResolution(Ext[3], Alg);
```

$$\mathrm{table}([1 = \begin{bmatrix} 1+\delta^2 \\ 2\,\delta \\ -Dt\,\delta^2 + Dt \end{bmatrix}, 2 = \begin{bmatrix} -2\,\delta & 1+\delta^2 & 0 \\ -Dt & Dt\,\delta & 1 \\ Dt\,\delta & -Dt & \delta \end{bmatrix}, 3 = \begin{bmatrix} Dt & -\delta & 1 \end{bmatrix}, 4 = \mathrm{INJ}(1)])$$

We find that the second entry of this table, i.e., the first syzygy module of *Ext*[3] is *Ext*[2] again, which is another confirmation for the fact that *Ext*[3] gives a parametrization of the torsion-free part which is defined by *Ext*[2]. Moreover, the third module in the free resolution gives the relation that the rows of *Ext*[2] satisfy. In particular, the matrix *Ext*[2] does not have full row rank. We remember that the torsion-free part $M \;/\; \mathrm{t}(M)$ is free, and thus, projective. A simple criterion for projectiveness of a module associated with a full row rank matrix R is that R must have a right-inverse with entries in *Alg*. As *Ext*[2] does not have full row rank, this criterion is not applicable. Indeed, although $M \;/\; \mathrm{t}(M)$ is projective, we have:

```
> RightInverse(Ext[2], Alg);
```

$$[]$$

But the fact the *Alg*-module $M \;/\; \mathrm{t}(M)$, associated with non-full row rank matrix *Ext*[2], is projective is equivalent to the existence of a generalized inverse S which fulfills $Ext[2]\; S\; Ext[2] = Ext[2]$. Let us check whether or not a generalized inverse exists for *Ext*[2]:

```
> S := GeneralizedInverse(Ext[2], Alg);
```

$$S := \begin{bmatrix} \frac{\delta}{2} & 0 & 0 \\ 1 & 0 & 0 \\ -\frac{Dt\,\delta}{2} & 1 & 0 \end{bmatrix}$$

```
> Mult(Ext[2], S, Ext[2], Alg)-Ext[2];
```

$$0$$

Let us consider the example of a flexible rod with a mass considered in H. Mounier, *Propriétés structurelles des systèmes linéaires à retards: aspects théoriques et pratiques*, PhD thesis, University of Orsay, France, 1995. See also M. Fliess, H. Mounier, P. Rouchon, J. Rudolph, *Controllability and motion planning for linear delay systems with an application to a flexible rod*, in the proceedings of the 34th Conference on Decision & Control, New Orleans, 1995. We enter the matrix which defines the system:

```
> R2 := evalm([[Dt^2+Dt^2*delta^2+Dt-Dt*delta^2, -delta]]);
```

$$R2 := \begin{bmatrix} Dt^2 + Dt^2\,\delta^2 + Dt - Dt\,\delta^2 & -\delta \end{bmatrix}$$

Let us check whether or not the *Alg*-module associated with *R2* is torsion-free.

```
> ext1 := Exti(Involution(R2, Alg), Alg, 1);
```

$$ext1 := [\begin{bmatrix} 1 \end{bmatrix}, \begin{bmatrix} Dt^2 + Dt^2\,\delta^2 + Dt - Dt\,\delta^2 & -\delta \end{bmatrix}, \begin{bmatrix} \delta \\ Dt^2 + Dt^2\,\delta^2 + Dt - Dt\,\delta^2 \end{bmatrix}]$$

We obtain that the *Alg*-module associated with *R2* is torsion-free, and thus, the system is controllable and parametrizable. A parametrization of the system is defined by *ext1*[3] or, equivalently, by:

```
> Parametrization(R2, Alg);
```

$$\begin{bmatrix} \xi_1(t-h) \\ (\mathrm{D}^{(2)})(\xi_1)(t) + (\mathrm{D}^{(2)})(\xi_1)(t-2\,h) + \mathrm{D}(\xi_1)(t) - \mathrm{D}(\xi_1)(t-2\,h) \end{bmatrix}$$

Let us check whether or not the system is flat. In order to do that, we check whether or not the *Alg*-module associated with *R2* is projective.

```
> ext2 := Exti(Involution(R2, Alg), Alg, 2);
```

$$ext2 := [\begin{bmatrix} \delta \\ Dt^2 + Dt \end{bmatrix}, \begin{bmatrix} 1 \end{bmatrix}, \mathrm{SURJ}(1)]$$

We obtain that ext^2 is not zero, and thus, the system is not flat. Let us compute the obstruction of flatness as a polynomial in the time-delay operator δ.

```
> PiPolynomial(R2, Alg, [delta]);
```

$$[\delta]$$

Therefore, if we invert the operator δ, i.e., if we use the time-advance operator, then the $Alg[\delta^{-1}]$-module associated with *R2* becomes free. In particular, a basis of the $Alg[\delta^{-1}]$-module associated with *R2* is defined by $\xi = S\ (y\ :\ v)^T$, where y and v are the system variables and S is a left-inverse of the parametrization *ext1*[3], namely:

```
> S := LocalLeftInverse(ext1[3], [delta], Alg);
```

$$S := \begin{bmatrix} \frac{1}{\delta} & 0 \end{bmatrix}$$

Therefore, we have:

```
> xi(t)=ApplyMatrix(S, [y(t),v(t)], Alg)[1,1];
```

$$\xi(t) = \mathrm{y}(t+h)$$

```
> evalm([[y(t)],[v(t)]])=ApplyMatrix(ext1[3], [xi(t)], Alg);
```

$$\left[\begin{array}{c} \mathrm{y}(t) \\ \mathrm{v}(t) \end{array}\right] = \left[\begin{array}{c} \xi(t-h) \\ (\mathrm{D}^{(2)})(\xi)(t) + (\mathrm{D}^{(2)})(\xi)(t-2\,h) + \mathrm{D}(\xi)(t) - \mathrm{D}(\xi)(t-2\,h) \end{array}\right]$$

Moreover, we can substitute the flat output $\xi = S\ (y\,:\,v)^T$ into the parametrization of the system $(y : v)^T = ext1[3]\,\xi$, in order to obtain $(y\,:\,v)^T = Q\ (y\,:\,v)^T$, where Q is the following matrix:

```
>  Q := simplify(evalm(ext1[3] &* S));
```

$$Q := \left[\begin{array}{cc} 1 & 0 \\ \dfrac{Dt\,(Dt + Dt\,\delta^2 + 1 - \delta^2)}{\delta} & 0 \end{array}\right]$$

From the matrix Q, we easily see that we can also use $\xi = y$ as a flat output of the system as we have:

```
>  evalm([[y(t)],[v(t)]])=ApplyMatrix(Q, [y(t),v(t)], Alg);
```

$$\left[\begin{array}{c} \mathrm{y}(t) \\ \mathrm{v}(t) \end{array}\right] = \left[\begin{array}{c} \mathrm{y}(t) \\ (\mathrm{D}^{(2)})(y)(t+h) + (\mathrm{D}^{(2)})(y)(t-h) + \mathrm{D}(y)(t+h) - \mathrm{D}(y)(t-h) \end{array}\right]$$

In particular, we have obtained the input $v(t)$ in terms of the output $y(t)$ and an advance operator δ^{-1}. Therefore, we can do some motion planning as it is shown in H. Mounier, *Propriétés structurelles des systèmes linéaires à retards: aspects théoriques et pratiques*, PhD thesis, University of Orsay, France, 1995, and M. Fliess, H. Mounier, P. Rouchon, J. Rudolph, *Controllability and motion planning for linear delay systems with an application to a flexible rod*, in the proceedings of the 34th Conference on Decision & Control, New Orleans, 1995. Let also point out that one of the main difficulty is to stabilize the open-loop system by means of a suitable stabilizing controller. As we know from the theory of stabilization problems, this problem is generally a difficult one, especially for neutral differential time-delay systems.