Let us consider the example of a high speed network defined in M. Fliess, H. Mounier, *Quasi-finite linear delay systems: theory and applications*, IFAC Linear Time Delay Systems, Grenoble, France 1998, pp. 169-173.

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

We define the Ore algebra *Alg*, which contains the differential operator *Dt* w.r.t. time *t* and the shift operators $\delta 1$, $\delta 2$:

```
> Alg := DefineOreAlgebra(diff=[Dt,t], dual_shift=[delta1,s1],
> dual_shift=[delta2,s2], polynom=[t,s1,s2], shift_action=[delta1,t,tau1],
> shift_action=[delta2,t,tau2]):
```

The high speed network is modeled by the following matrix of operators.

```
> R := evalm([[Dt, 0, 1, -delta1], [0, Dt, -delta2, 0]]);
```

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

The system is controllable because the *Alg*-module *M* associated with the system is torsion-free:

```
> Ext1 := Exti(Involution(R, Alg), Alg, 1);
```

$$Ext1 := \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} Dt & 0 & 1 & -\delta 1 \\ 0 & Dt & -\delta 2 & 0 \end{bmatrix}, \begin{bmatrix} -\delta 1 & 1 \\ 0 & -\delta 2 \\ 0 & -Dt \\ -Dt & 0 \end{bmatrix} \right]$$

A parametrization of the system is defined by *Ext1*[3]. It is a minimal parametrization, as we have:

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

$$\left[\begin{bmatrix} -\delta 1 & 1 \\ 0 & -\delta 2 \\ 0 & -Dt \\ -Dt & 0 \end{bmatrix} \right]$$

The same parametrization can be obtained by using *Parametrization*. The result involves two free functions ξ_1, ξ_2:

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

$$\begin{bmatrix} -\xi_1(t-\tau 1)+\xi_2(t) \\ -\xi_2(t-\tau 2) \\ -\mathrm{D}(\xi_2)(t) \\ -\mathrm{D}(\xi_1)(t) \end{bmatrix}$$

Let us compute the second extension module ext^2 with values in *Alg* of the left *Alg*-module which is associated with *Involution*(*R*, *Alg*):

```
> Ext2 := Exti(Involution(R, Alg), Alg, 2);
```

$$Ext2 := \left[\begin{bmatrix} Dt & 0 \\ \delta 2\,\delta 1 & 0 \\ 0 & \delta 2 \\ 0 & Dt \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathrm{SURJ}(2) \right]$$

The *Alg*-module M associated with the system is not reflexive, because the first matrix *Ext2*[1] of *Ext2* is not an identity matrix. Therefore, M is not a projective, and thus, not a free *Alg*-module. Hence, the corresponding high speed network is not flat.

The matrix R has full row rank, as we can easily check by computing a free resolution of the *Alg*-module M.

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

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

Let us recall that we know that an *Alg*-module M defined by a full row rank matrix R is projective iff R admits a right-inverse. Hence, we already know that R should have no right-inverse. Let us check it:

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

$$[]$$

Moreover, we also know that the parametrization *Ext1*[3] of the system is not injective, as it would imply that the *Alg*-module M is free. Let us check it:

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

$$[]$$

As the *Alg*-module M is torsion-free but not free, we know that the high speed network is π-free. Let us compute a π-polynomial for this system:

```
> PiPolynomial(R, Alg, [delta1,delta2]);
```

$$[\delta 1\, \delta 2]$$

Thus, the system is $\pi = \delta 1\, \delta 2$-free. Let us compute a left-inverse of the parametrization *Ext1*[3] in the ring $Alg[\pi^{-1}]$.

```
> L := LocalLeftInverse(Ext1[3], [delta1*delta2], Alg);
```

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

We easily check that L is a left-inverse of *Ext1*[3]:

```
> simplify(evalm(L &* Ext1[3]));
```

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

We obtain that $(\xi 1(t) : \xi 2(t))^T = L(\xi(t) : \chi(t) : \mathrm{u}(t) : \mu(t))^T$ is a flat output of the system. More precisely, we have:

```
> evalm([[xi1(t)],[xi2(t)]])=ApplyMatrix(L, [xi(t),chi(t),u(t),mu(t)], Alg);
```

$$\begin{bmatrix} \xi 1(t) \\ \xi 2(t) \end{bmatrix} = \begin{bmatrix} -\xi(t + \tau 1) - \chi(t + \tau 2 + \tau 1) \\ -\chi(t + \tau 2) \end{bmatrix}$$

```
> P := simplify(evalm(Ext1[3] &* L));
```

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

The matrix P is such that we have $(\xi(t) : \chi(t) : \mathrm{u}(t) : \mu(t))^T = P\ (\xi(t) : \chi(t) : \mathrm{u}(t) : \mu(t))^T$. From P, we easily see that the system variables $\mathrm{u}(t)$ and $\mu(t)$ can be expressed as $Alg[\pi^{-1}]$-linear combinations of $\xi(t)$ and $\chi(t)$ showing that $\xi(t)$ and $\chi(t)$ are flat outputs of the high speed network over $Alg[\pi^{-1}]$. More precisely, we have:

```
> evalm([[xi(t)], [chi(t)], [mu(t)], [u(t)]])=
> ApplyMatrix(P,[xi(t),chi(t),mu(t),u(t)],Alg);
```

$$\begin{bmatrix} \xi(t) \\ \chi(t) \\ \mu(t) \\ \mathrm{u}(t) \end{bmatrix} = \begin{bmatrix} \xi(t) \\ \chi(t) \\ \mathrm{D}(\chi)(t + \tau 2) \\ \mathrm{D}(\xi)(t + \tau 1) + \mathrm{D}(\chi)(t + \tau 2 + \tau 1) \end{bmatrix}$$