The purpose of this Maple worksheet is to show how to use *OreModules* in order to handle the *Pedalvit* exercise given in J.-P. Quadrat and M. Viot, *Introduction à la commande stochastique*, Lecture Notes of Ecole Polytechnique, 1999. The system considered is formed by an equilibrist clown riding a one-wheeled bicycle. This system is modelized by a linearized inverted pendulum around the vertical equilibrum point. We denote by m the mass of the wheel. For sake of simplicity, the mass of the clown's head, the length of the pendulum and the gravity constant g are all supposed to be equal to 1.

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

Define the Weyl algebra $Alg = A_1$, where D acts as differential operator w.r.t. the variable t:

```
> Alg := DefineOreAlgebra(diff=[D,t], polynom=[t], comm=[m,gamma]):
```

Then, the matrix of ordinary differential operators defining the system is the following one:

```
> R := evalm([[(1+m)*D^2, D^2,1],[D^2,D^2-1,0]]);
```

$$R := \begin{bmatrix} (1+m)\,\mathrm{D}^2 & \mathrm{D}^2 & 1 \\ \mathrm{D}^2 & \mathrm{D}^2-1 & 0 \end{bmatrix}$$

The system variables are the horizontal displacement x, the angle with respect to the vertical θ and the force u acting on the one-wheeled bicycle by the clown *Pedalvit*. Therefore, the system equations are then given by:

```
> ApplyMatrix(R, [x(t),theta(t),u(t)], Alg)=evalm([[0]$2]);
```

$$\begin{bmatrix} (1+m)\,(\frac{d^2}{dt^2}\,\mathrm{x}(t)) + (\frac{d^2}{dt^2}\,\theta(t)) + \mathrm{u}(t) \\ (\frac{d^2}{dt^2}\,\mathrm{x}(t)) - \theta(t) + (\frac{d^2}{dt^2}\,\theta(t)) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

First of all, let us study whether or not this system is controllable, parametrizable and flat. In order to do that, we first define the formal adjoint *R_adj* of R.

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

$$R\_adj := \begin{bmatrix} \mathrm{D}^2 + \mathrm{D}^2\,m & \mathrm{D}^2 \\ \mathrm{D}^2 & \mathrm{D}^2-1 \\ 1 & 0 \end{bmatrix}$$

Then, the system is controllable, and thus, parametrizable iff the *Alg*-module associated with R is torsion-free or, equivalently, if the first extension module ext^1 with values in *Alg* of the *Alg*-module associated with *R_adj* is the zero module. Let us check this last point.

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

$$Ext := \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \mathrm{D}^2\,m & 1 & 1 \\ 0 & -1-m+\mathrm{D}^2\,m & -1 \end{bmatrix}, \begin{bmatrix} 1-\mathrm{D}^2 \\ \mathrm{D}^2 \\ -\mathrm{D}^2-\mathrm{D}^2\,m+\mathrm{D}^4\,m \end{bmatrix} \right]$$

$$0.179$$

As the first matrix is the identity, ext^1 of the *Alg*-module associated with *R_adj* is the zero module. Therefore, the system is controllable and $Ext[3]$ defines a parametrization of the system, namely we have:

```
> evalm([[x(t)],[theta(t)],[u(t)]])=Parametrization(R, Alg);
```

$$\begin{bmatrix} \mathrm{x}(t) \\ \theta(t) \\ \mathrm{u}(t) \end{bmatrix} = \begin{bmatrix} \xi_1(t) - \%1 \\ \%1 \\ -\%1 - \%1\, m + m\,(\frac{d^4}{dt^4}\,\xi_1(t)) \end{bmatrix}$$
$$\%1 := \tfrac{d^2}{dt^2}\,\xi_1(t)$$

As the system is time-invariant, the system is also flat. Then, a flat output of the system is defined by computing a left-inverse of the parametrization $Ext[3]$.

```
> S := LeftInverse(Ext[3], Alg);
```
$$S := \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

Hence, we have the following flat output of the system:

```
> xi[1](t)=ApplyMatrix(S, [x(t),theta(t),u(t)], Alg)[1,1];
```
$$\xi_1(t) = \mathrm{x}(t) + \theta(t)$$

which satisfies $(x : \theta : u)^T = Ext[3]\ \xi[1]$. Therefore, we can do some motion planning by determining a trajectory of the system which starts at the position $\mathrm{x}(0) = 1$, $\mathrm{D}\,\mathrm{x}(0) = 0$, $\theta(0)$=0, $\mathrm{D}\,\theta(0) = 0$ and ends at $\mathrm{x}(1) = 0$, D x(1)=0, $\theta(1) = 0$, $\mathrm{D}\,\theta(1) = 0$. In order to do that, we consider a polynomial trajectory $\xi[1]$ of degree 7. By substituting this polynomial trajectory into the parametrization of the system, we obtain:

```
> M1 := ApplyMatrix(Ext[3], [sum(a[i]*t^i,i=0..7)], Alg);
```
$$M1 :=$$
$$\Big[a_0 + a_1\,t + a_2\,t^2 + a_3\,t^3 + a_4\,t^4 + a_5\,t^5 + a_6\,t^6 + a_7\,t^7 - 2\,a_2 - 6\,a_3\,t - 12\,a_4\,t^2$$
$$- 20\,a_5\,t^3 - 30\,a_6\,t^4 - 42\,a_7\,t^5\Big]$$
$$\left[2\,a_2 + 6\,a_3\,t + 12\,a_4\,t^2 + 20\,a_5\,t^3 + 30\,a_6\,t^4 + 42\,a_7\,t^5\right]$$
$$\Big[(-1-m)\,(2\,a_2 + 6\,a_3\,t + 12\,a_4\,t^2 + 20\,a_5\,t^3 + 30\,a_6\,t^4 + 42\,a_7\,t^5)$$
$$+ m\,(24\,a_4 + 120\,a_5\,t + 360\,a_6\,t^2 + 840\,a_7\,t^3)\Big]$$

In order to impose the conditions on the derivative of x and θ, we also need to differentiate the previous result.

```
> M2 := ApplyMatrix(Mult(D, Ext[3], Alg), [sum(a[i]*t^i,i=0..7)], Alg);
```
$$M2 :=$$
$$\Big[a_1 + 2\,a_2\,t + 3\,a_3\,t^2 + 4\,a_4\,t^3 + 5\,a_5\,t^4 + 6\,a_6\,t^5 + 7\,a_7\,t^6 - 6\,a_3 - 24\,a_4\,t - 60\,a_5\,t^2$$
$$- 120\,a_6\,t^3 - 210\,a_7\,t^4\Big]$$
$$\left[6\,a_3 + 24\,a_4\,t + 60\,a_5\,t^2 + 120\,a_6\,t^3 + 210\,a_7\,t^4\right]$$
$$\Big[(-1-m)\,(6\,a_3 + 24\,a_4\,t + 60\,a_5\,t^2 + 120\,a_6\,t^3 + 210\,a_7\,t^4)$$
$$+ m\,(120\,a_5 + 720\,a_6\,t + 2520\,a_7\,t^2)\Big]$$

Then, by substituting $t = 0$

```
> N1 := subs(t=0, evalm(M1)); N2 := subs(t=0, evalm(M2));
```

$$N1 := \begin{bmatrix} a_0 - 2\,a_2 \\ 2\,a_2 \\ 2\,(-1-m)\,a_2 + 24\,m\,a_4 \end{bmatrix}$$

$$N2 := \begin{bmatrix} a_1 - 6\,a_3 \\ 6\,a_3 \\ 6\,(-1-m)\,a_3 + 120\,m\,a_5 \end{bmatrix}$$

and $t = 1$ into the previous matrices

```
> N3 := subs(t=1, evalm(M1)); N4 := subs(t=1, evalm(M2));
```

$$N3 := \left[a_0 + a_1 - a_2 - 5\,a_3 - 11\,a_4 - 19\,a_5 - 29\,a_6 - 41\,a_7\right]$$
$$\left[2\,a_2 + 6\,a_3 + 12\,a_4 + 20\,a_5 + 30\,a_6 + 42\,a_7\right]$$
$$\Big[(-1-m)\,(2\,a_2 + 6\,a_3 + 12\,a_4 + 20\,a_5 + 30\,a_6 + 42\,a_7)$$
$$+ m\,(24\,a_4 + 120\,a_5 + 360\,a_6 + 840\,a_7)\Big]$$

$$N4 := \begin{bmatrix} a_1 + 2\,a_2 - 3\,a_3 - 20\,a_4 - 55\,a_5 - 114\,a_6 - 203\,a_7 \\ 6\,a_3 + 24\,a_4 + 60\,a_5 + 120\,a_6 + 210\,a_7 \\ (-1-m)\,(6\,a_3 + 24\,a_4 + 60\,a_5 + 120\,a_6 + 210\,a_7) + m\,(120\,a_5 + 720\,a_6 + 2520\,a_7) \end{bmatrix}$$

we finally obtain the conditions on the coefficients $a[i]$ of the polynomial trajectory $\xi[1]\_ref$. Solving this system, we obtain:

```
> Sol := solve({N1[1,1]=1, N1[2,1]=0, N2[1,1]=0, N2[2,1]=0, N3[1,1]=0,
> N3[2,1]=0, N4[1,1]=0, N4[2,1]=0});
```

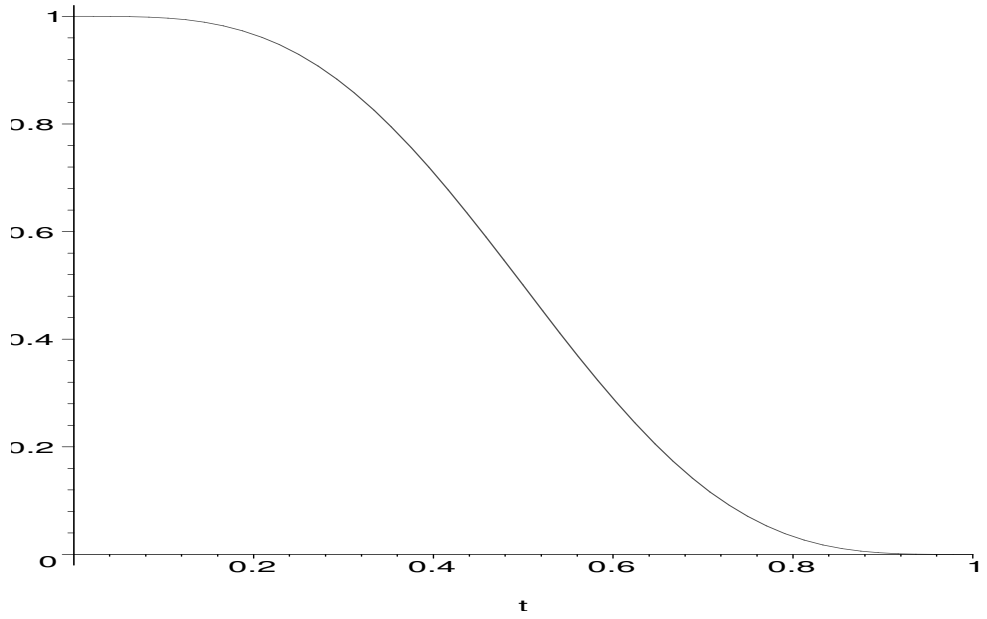
$$Sol := \{a_6 = -70,\ a_4 = -35,\ a_2 = 0,\ a_3 = 0,\ a_5 = 84,\ a_1 = 0,\ a_0 = 1,\ a_7 = 20\}$$

Hence, the particular polynomial trajectory for $\xi[1]$, called $Path$, is then defined by:

```
> Path := subs(Sol, sum(a[i]*t^i,i=0..7));
```

$$Path := 1 - 35\,t^4 + 84\,t^5 - 70\,t^6 + 20\,t^7$$

```
> plot(Path, t=0..1);
```



Then, the corresponding trajectory of the system ($x\_ref : \theta\_ref : u\_ref$) is defined by:

```
> P := ApplyMatrix(Ext[3], [Path], Alg);
```

$$
\begin{aligned}
&P := \\
&[1 + 2065\,t^4 - 756\,t^5 - 70\,t^6 + 20\,t^7 + 420\,t^2 - 1680\,t^3] \\
&[-420\,t^2 + 1680\,t^3 - 2100\,t^4 + 840\,t^5] \\
&[(-1 - m)\,(-420\,t^2 + 1680\,t^3 - 2100\,t^4 + 840\,t^5) \\
&\quad + m\,(-840 + 10080\,t - 25200\,t^2 + 16800\,t^3)]
\end{aligned}
$$

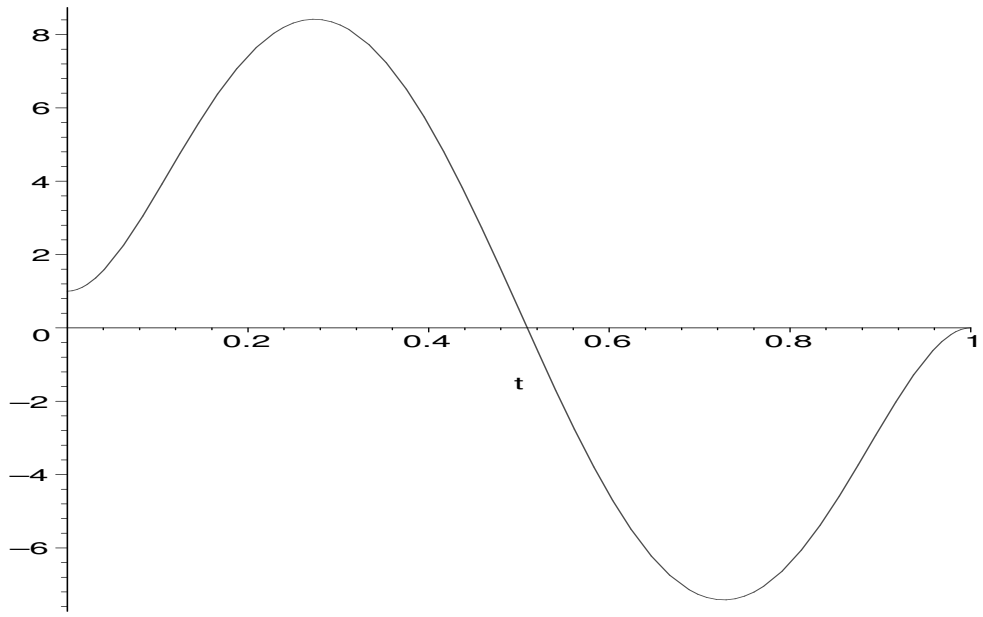
We can check that it is a solution of the system by substituting it into the equations of the system:

```
> simplify(ApplyMatrix(R, P, Alg));
```

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

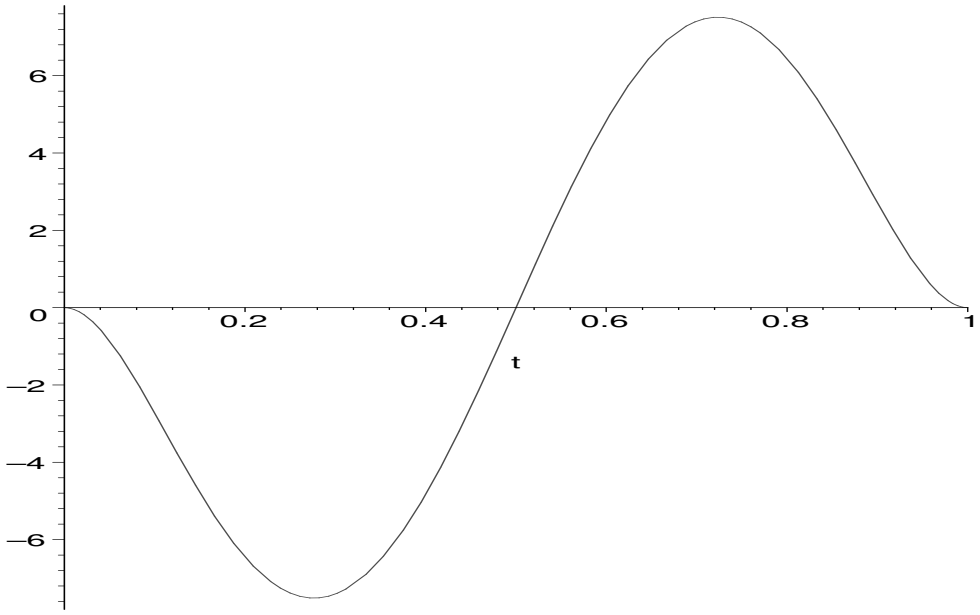
Moreover, we have the following plot for the displacement $x\_\mathrm{ref}(t)$:

```
> plot(P[1,1],t=0..1);
```



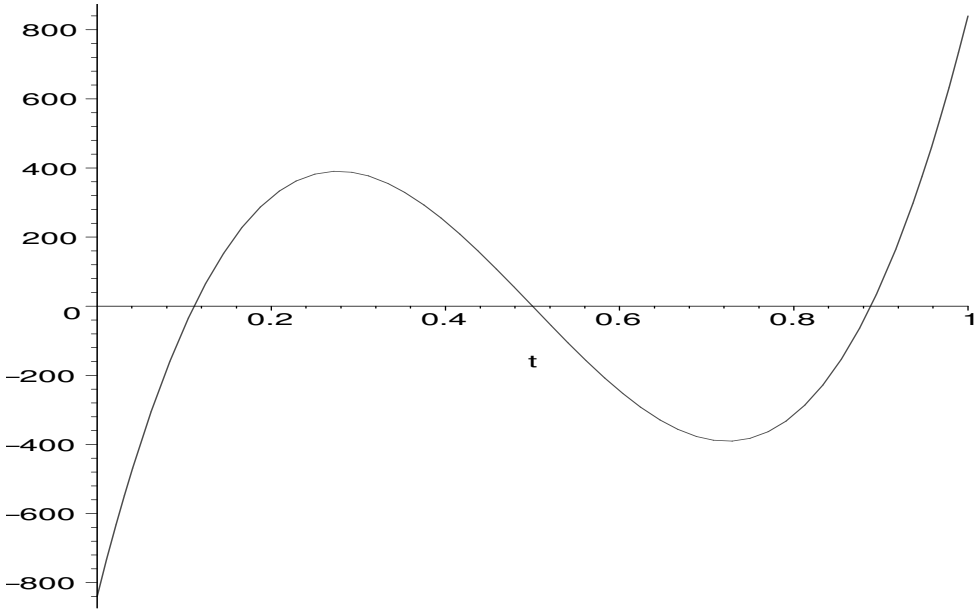
Let us plot the trajectory of the angle $\theta\_\mathrm{ref}(t)$:

```
> plot(P[2,1], t=0..1);
```



Finally, if we choose $m = 1$, then the open-loop input $u\_\mathrm{ref}(t)$ is defined by:

```
> plot(subs(m=1, P[3,1]), t=0..1);
```



Now, if we take the previous input u_ref, then the equations of the system become:

```
> System := subs(m=1, ApplyMatrix(R, [x(t),theta(t),P[3,1]], Alg));
```

$$System := \left[\begin{array}{c} 2\,(\frac{d^2}{dt^2}\,\mathrm{x}(t)) + (\frac{d^2}{dt^2}\,\theta(t)) - 24360\,t^2 + 13440\,t^3 + 4200\,t^4 - 1680\,t^5 - 840 + 10080\,t \\ (\frac{d^2}{dt^2}\,\mathrm{x}(t)) - \theta(t) + (\frac{d^2}{dt^2}\,\theta(t)) \end{array}\right]$$

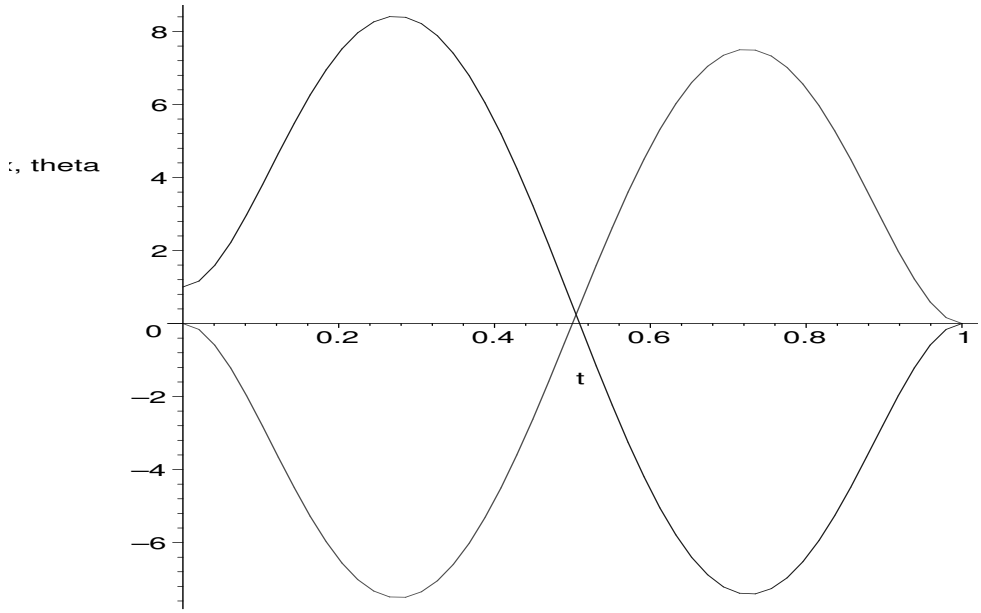
Therefore, if we choose the following initial conditions

```
> Initial := {x(0)=1,D(x)(0)=0,theta(0)=0,D(theta)(0)=0};
```

$$Initial := \{\mathrm{x}(0) = 1,\ \mathrm{D}(x)(0) = 0,\ \theta(0) = 0,\ \mathrm{D}(\theta)(0) = 0\}$$

then we can simulate the open-loop behaviour of the system by numerically integrating it:

```
> dsol := dsolve({System[1,1]=0, System[2,1]=0} union Initial,
> type=numeric, stiff=true):
> plots[odeplot](dsol, [[t,x(t), color=blue], [t,theta(t),color=red]],
> 0..1);
```



In order to stabilize the system around the reference trajectory, we first compute the Brunovský canonical form of the system.

```
> st := time():  B := Brunovsky(R, Alg); time()-st;
```

$$B := \begin{bmatrix} 1 & 1 & 0 \\ \mathrm{D} & \mathrm{D} & 0 \\ 0 & 1 & 0 \\ 0 & \mathrm{D} & 0 \\ 0 & \dfrac{1+m}{m} & \dfrac{1}{m} \end{bmatrix}$$

$$0.290$$

Therefore, if we denote by $z[1]$, ..., $z[4]$ and v the Brunovský canonical variables, then we have the following transformation between the Brunovský and the system variables x, θ and u:

```
> evalm([seq([z[i](t)],i=1..4),[v(t)]])=ApplyMatrix(B, [x(t),theta(t),u(t)], Alg);
```

$$\begin{bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \\ z_4(t) \\ \mathrm{v}(t) \end{bmatrix} = \begin{bmatrix} \mathrm{x}(t) + \theta(t) \\ (\frac{d}{dt}\,\mathrm{x}(t)) + (\frac{d}{dt}\,\theta(t)) \\ \theta(t) \\ \frac{d}{dt}\,\theta(t) \\ \dfrac{(1+m)\,\theta(t)}{m} + \dfrac{\mathrm{u}(t)}{m} \end{bmatrix}$$

Let us check that the new variables $z[1]$, ..., $z[4]$ and v satisfy a Brunovský canonical form:

```
> E := Elimination(linalg[stackmatrix](B, R), [x,theta,u],
> [seq(z[i],i=1..4),v,0,0], Alg):
> ApplyMatrix(E[1], [x(t),theta(t),u(t)], Alg)=ApplyMatrix(E[2],
> [seq(z[i](t),i=1..4),v(t)], Alg);
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathrm{u}(t) \\ \theta(t) \\ \mathrm{x}(t) \end{bmatrix} = \begin{bmatrix} -(\frac{d}{dt}\,z_4(t)) + \mathrm{v}(t) \\ -(\frac{d}{dt}\,z_3(t)) + z_4(t) \\ -(\frac{d}{dt}\,z_2(t)) + z_3(t) \\ -(\frac{d}{dt}\,z_1(t)) + z_2(t) \\ (-1-m)\,z_3(t) + m\,\mathrm{v}(t) \\ z_3(t) \\ z_1(t) - z_3(t) \end{bmatrix}$$

From the first four equations, we check that we have a Brunovský canonical form in the variables $z[1]$, ..., $z[4]$ and v. Let us also notice that the last three equations give the transformation of the system variables x, θ and u in terms of the Brunovský variables z[i], ..., $z[4]$, and v. In the new variables, the system matrix becomes

```
> R2 := evalm(linalg[submatrix](evalm(-E[2]), 1..4, 1..5));
```

$$R2 := \begin{bmatrix} 0 & 0 & 0 & \mathrm{D} & -1 \\ 0 & 0 & \mathrm{D} & -1 & 0 \\ 0 & \mathrm{D} & -1 & 0 & 0 \\ \mathrm{D} & -1 & 0 & 0 & 0 \end{bmatrix}$$

and we have the following corresponding equations:

```
> ApplyMatrix(R2, [seq(z[i](t),i=1..4),v(t)], Alg)=evalm([[0]$4]);
```

$$\begin{bmatrix} (\frac{d}{dt}\,z_4(t)) - \mathrm{v}(t) \\ (\frac{d}{dt}\,z_3(t)) - z_4(t) \\ (\frac{d}{dt}\,z_2(t)) - z_3(t) \\ (\frac{d}{dt}\,z_1(t)) - z_2(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

```
> P2 := Parametrization(R2, Alg);
```

$$P2 := \begin{bmatrix} \xi_1(t) \\ \frac{d}{dt}\,\xi_1(t) \\ \frac{d^2}{dt^2}\,\xi_1(t) \\ \frac{d^3}{dt^3}\,\xi_1(t) \\ \frac{d^4}{dt^4}\,\xi_1(t) \end{bmatrix}$$

Let us point out that if we substitute the parametrization $Ext[3]$ of the first system into the Brunovský matrix B, then we obtain the first *Spencer operator*, namely:

```
> J4 := Mult(B, Ext[3], Alg);
```

$$J4 := \begin{bmatrix} 1 \\ \mathrm{D} \\ \mathrm{D}^2 \\ \mathrm{D}^3 \\ \mathrm{D}^4 \end{bmatrix}$$

See A. Quadrat, *Analyse algébrique des systèmes de contrôle linéaires multidimensionnels*, PhD Thesis, Ecole Nationale des Ponts et Chaussées, 1999, for more details about the links between the Spencer operator and the Brunovský canonical form. Hence, if we consider the reference polynomial trajectory defined above, then we obtain the corresponding $z[i]$ and v by differentiation:

```
> ZVref := ApplyMatrix(J4, [Path], Alg);
```

$$ZVref := \begin{bmatrix} 1 - 35\,t^4 + 84\,t^5 - 70\,t^6 + 20\,t^7 \\ -140\,t^3 + 420\,t^4 - 420\,t^5 + 140\,t^6 \\ -420\,t^2 + 1680\,t^3 - 2100\,t^4 + 840\,t^5 \\ -840\,t + 5040\,t^2 - 8400\,t^3 + 4200\,t^4 \\ -840 + 10080\,t - 25200\,t^2 + 16800\,t^3 \end{bmatrix}$$

More precisely, we have:

```
> evalm([seq([z_ref[i](t)],i=1..4),[v_ref(t)]])=evalm(ZVref);
```

$$\begin{bmatrix} z\_ref_1(t) \\ z\_ref_2(t) \\ z\_ref_3(t) \\ z\_ref_4(t) \\ \mathrm{v\_ref}(t) \end{bmatrix} = \begin{bmatrix} 1 - 35\,t^4 + 84\,t^5 - 70\,t^6 + 20\,t^7 \\ -140\,t^3 + 420\,t^4 - 420\,t^5 + 140\,t^6 \\ -420\,t^2 + 1680\,t^3 - 2100\,t^4 + 840\,t^5 \\ -840\,t + 5040\,t^2 - 8400\,t^3 + 4200\,t^4 \\ -840 + 10080\,t - 25200\,t^2 + 16800\,t^3 \end{bmatrix}$$

We easily check that $ZVref$ is a trajectory of the system as we have:

```
> ApplyMatrix(R2, ZVref, Alg);
```

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

If we denote by T the matrix expressing the system variable x, θ and u in terms of the Brunovský variables $z[1]$, ..., $z[4]$ and v,

```
> T := linalg[submatrix](evalm(E[2]), 5..7, 1..5);
```

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

then, by substitution, we find again the reference trajectory computed previously.

```
> evalm([[u_ref(t)],[theta_ref(t)],[x_ref(t)]])=ApplyMatrix(T, ZVref, Alg);
```

$$\begin{bmatrix} \mathrm{u\_ref}(t) \\ \mathrm{theta\_ref}(t) \\ \mathrm{x\_ref}(t) \end{bmatrix} =$$
$$[(-1-m)\,(-420\,t^2 + 1680\,t^3 - 2100\,t^4 + 840\,t^5)$$
$$+\,m\,(-840 + 10080\,t - 25200\,t^2 + 16800\,t^3)]$$
$$[-420\,t^2 + 1680\,t^3 - 2100\,t^4 + 840\,t^5]$$
$$[1 + 2065\,t^4 - 756\,t^5 - 70\,t^6 + 20\,t^7 + 420\,t^2 - 1680\,t^3]$$

Hence, for $m = 1$ and v_ref, we obtain the following open-loop system:

```
> Sys_openloop := ApplyMatrix(R2, [seq(z[i](t),i=1..4),ZVref[5,1]], Alg);
```

$$Sys\_openloop := \begin{bmatrix} (\frac{d}{dt} z_4(t)) + 840 - 10080\,t + 25200\,t^2 - 16800\,t^3 \\ (\frac{d}{dt} z_3(t)) - z_4(t) \\ (\frac{d}{dt} z_2(t)) - z_3(t) \\ (\frac{d}{dt} z_1(t)) - z_2(t) \end{bmatrix}$$

The previous initial conditions for x and θ give the following initial conditions for the $z[i]$:

```
> IC := {z[1](0)=1, seq(z[i](0)=0,i=2..4)};
```

$$IC := \{z_2(0) = 0,\ z_3(0) = 0,\ z_1(0) = 1,\ z_4(0) = 0\}$$

Then, we can numerically integrate the corresponding system.

```
> dsol_openloop := dsolve({seq(Sys_openloop[i,1]=0,i=1..4)} union IC,
> type=numeric, stiff=true):
```

However, due to some disturbances, pertubations or modelizing errors, we need to stabilize the system around the trajectory $ZVref$. In order to compare with the closed-loop system thereafter, we numerically integrate the previous one but with a constant perturbation equal to $\frac{1}{10}$ in the right hand side of the equations $Sys\_openloop$.

```
> dsol_openloopnoise := dsolve({seq(Sys_openloop[i,1]=1/10,i=1..4)}
> union IC, type=numeric, stiff=true):
```

The Brunovský system is in particular a Kalman system $\mathrm{D}\,z = F\,z + G\,v$, where F and G are defined by:

```
> F := linalg[JordanBlock](0, 4); G := evalm([[0],[0],[0],[1]]);
```

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

$$G := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Then, we have $\mathrm{D}\,z\_ref = F\,z\_ref + G\,v\_ref$ and, if we define $e = z - z\_ref$, then the error satisfies the system $\mathrm{D}\,e = F\,e + G\ (v - v\_ref)$. Using the fact that the system is controllable, we can stabilize the system by using a feedback $v = v\_ref\ + K\ e$. Hence, we have $\mathrm{D}\,e = (F + GK)\ e$ and

$$D\,z = (F + GK)\,z + G\,v\_ref - GK\,z\_ref.$$

Therefore, we need to choose a constant feedback K such that the error asymptotically tends to 0.

```
> K := evalm([[k1,k2,k3,k4]]);
```

$$K := \begin{bmatrix} k1 & k2 & k3 & k4 \end{bmatrix}$$

Then, the matrix $F + GK$ is defined by:

```
> Fclos := evalm(F + G &* K);
```

$$Fclos := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ k1 & k2 & k3 & k4 \end{bmatrix}$$

Its characteristic polynomial is defined by:

```
> linalg[charpoly](Fclos, lambda);
```

$$\lambda^4 - \lambda^3\, k4 - \lambda^2\, k3 - \lambda\, k2 - k1$$

If we want that the closed-loop characteristic polynomial equals

```
> expand((lambda+10)^4);
```

$$\lambda^4 + 40\,\lambda^3 + 600\,\lambda^2 + 4000\,\lambda + 10000$$

then we need to choose $k1 = -10^4$, $k2 = -4000$, $k3 = -600$ and $k4 = -40$. Then, the corresponding K and $F + GK$ become:

```
> Kval := subs(k1=-10^4,k2=-4000,k3=-600,k4=-40, evalm(K));
```

$$Kval := \begin{bmatrix} -10000 & -4000 & -600 & -40 \end{bmatrix}$$

```
> Fclosval := subs(k1=-10^4,k2=-4000,k3=-600,k4=-40, evalm(Fclos));
```

$$Fclosval := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -10000 & -4000 & -600 & -40 \end{bmatrix}$$

Then, the system $\mathrm{D}\,z = (F + GK)\,z + G\,v$ is defined by the following matrix:

```
> Rclos := linalg[augment](evalm(D-(F + G &* Kval)), -G);
```

$$Rclos := \begin{bmatrix} \mathrm{D} & -1 & 0 & 0 & 0 \\ 0 & \mathrm{D} & -1 & 0 & 0 \\ 0 & 0 & \mathrm{D} & -1 & 0 \\ 10000 & 4000 & 600 & 40 + \mathrm{D} & -1 \end{bmatrix}$$

Moreover, $GK\,v\_ref$ is defined by:

```
> GKvalz_ref := evalm((G &* Kval) &* linalg[submatrix](ZVref, 1..4, 1..1));
```

$$GKvalz\_ref :=$$
$$[0]$$
$$[0]$$
$$[0]$$
$$[-10000 - 238000\,t^4 + 336000\,t^5 + 140000\,t^6 - 200000\,t^7 - 112000\,t^3 + 50400\,t^2 + 33600\,t]$$

Hence, the system $\mathrm{D}\,z = (F + GK)\,z + G\,v\_ref - GK\,z\_ref$ is then defined by:

```
> Sys_closedloop := evalm(ApplyMatrix(Rclos,
> [seq(z[i](t),i=1..4),ZVref[5,1]], Alg)+GKvalz_ref);
```

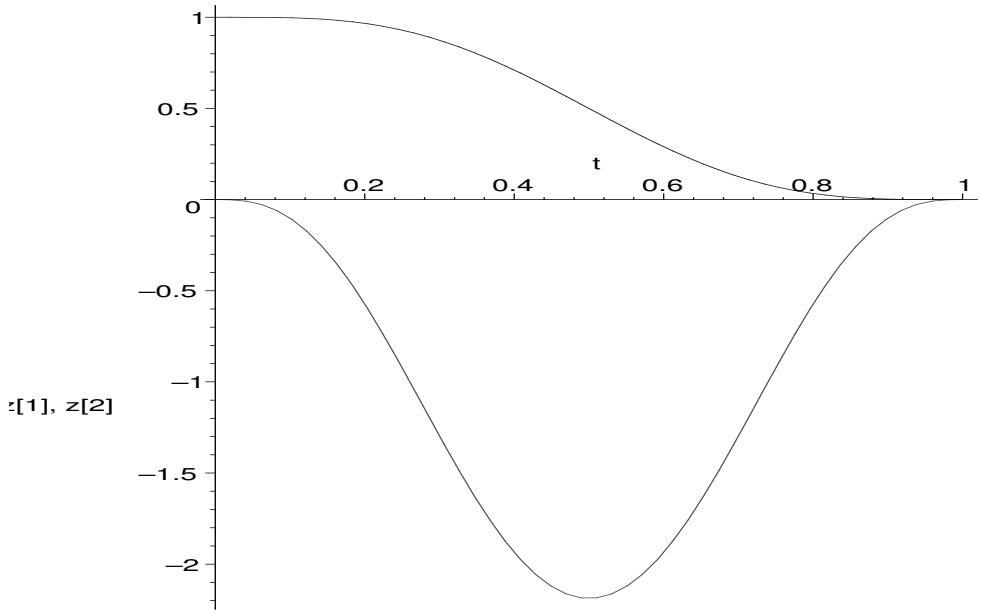
$$
\begin{aligned}
&Sys\_closedloop := \\
&\left[\left(\tfrac{d}{dt}\, z_1(t)\right) - z_2(t)\right] \\
&\left[\left(\tfrac{d}{dt}\, z_2(t)\right) - z_3(t)\right] \\
&\left[\left(\tfrac{d}{dt}\, z_3(t)\right) - z_4(t)\right] \\
&\Big[10000\, z_1(t) + 4000\, z_2(t) + 600\, z_3(t) + 40\, z_4(t) + \left(\tfrac{d}{dt}\, z_4(t)\right) - 9160 + 23520\, t \\
&+ 75600\, t^2 - 128800\, t^3 - 238000\, t^4 + 336000\, t^5 + 140000\, t^6 - 200000\, t^7\Big]
\end{aligned}
$$

Let us integrate it numerically with the same right hand side perturbation $\frac{1}{10}$.

```
> dsol_closedloop := dsolve({seq(Sys_closedloop[i,1]=1/10,i=1..4)}
> union IC, type=numeric, stiff=true):
```

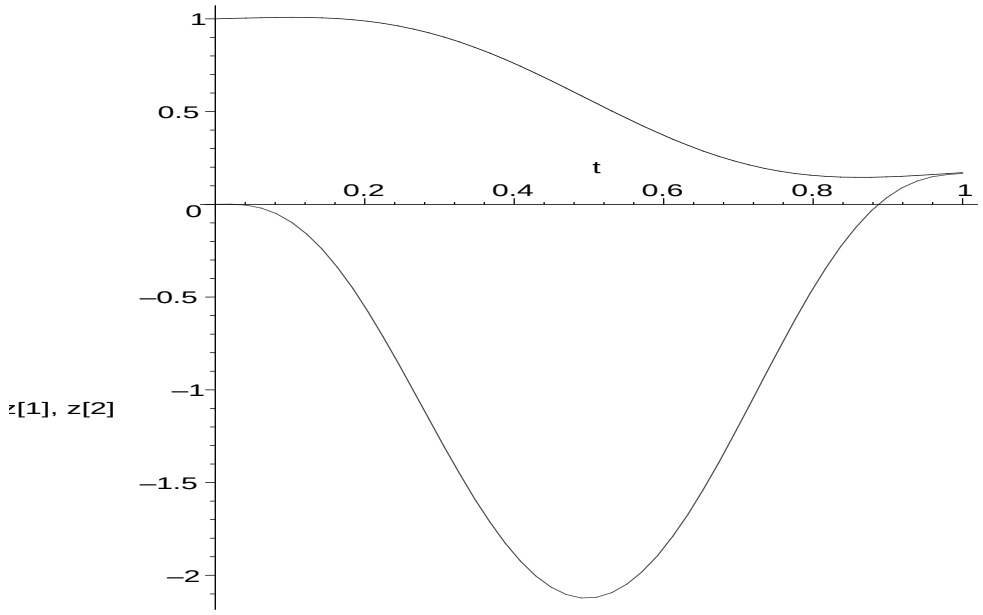
The open-loop $z[1]$ and $z[2]$ are the following ones:

```
> plots[odeplot](dsol_openloop, [[t,z[1](t),color=blue],
> [t,z[2](t),color=red]], 0..1);
```



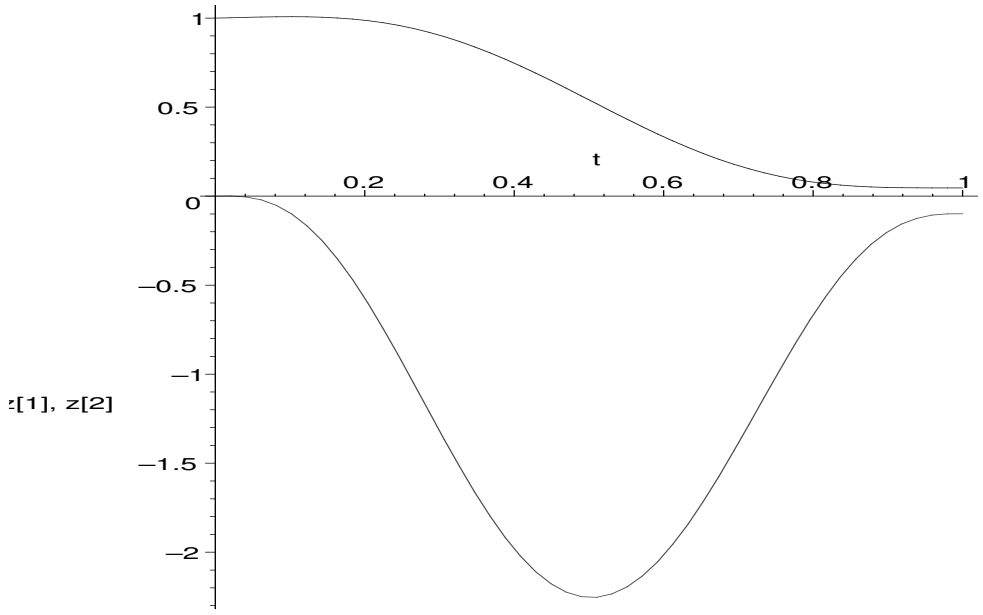
The plots of the open-loop $z[1]$ and $z[2]$ with the perturbation are the following ones:

```
> plots[odeplot](dsol_openloopnoise, [[t,z[1](t),color=blue],
> [t,z[2](t),color=red]], 0..1);
```



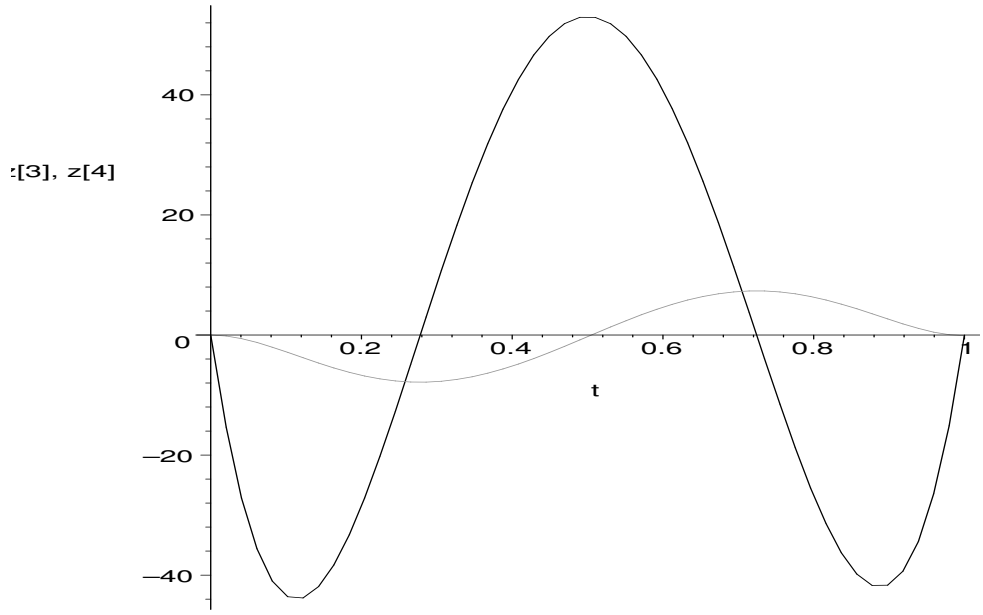
Finally, we have the following closed-loop $z[1]$ and $z[2]$ in the case where the perturbation occurs:

```
> plots[odeplot](dsol_closedloop, [[t,z[1](t),color=blue],
> [t,z[2](t),color=red]], 0..1);
```



We easily understand why we need to stabilize the system around the desired trajectory using a feedback law. In all the three previous cases, the components $z[3]$ and $z[4]$ closely follow the same trajectories. We only plot here the case with the feedback law.

```
> plots[odeplot](dsol_closedloop, [[t,z[3](t),color=green],
> [t,z[4](t),color=black]], 0..1);
```



Now, we study the optimal control for the *Pedalvit* example following the exercise given in J.-P. Quadrat and M. Viot, *Introduction à la commande stochastique*, Lecture Notes of Ecole Polytechnique, 1999. For more details on the theory, we refer the reader to A. Quadrat, *Analyse algébrique des systèmes de contrôle linéaires multidimensionnels*, PhD Thesis, Ecole Nationale des Ponts et Chaussées, 1999, as well as J.-F. Pommaret, A. Quadrat, *A differential operator approach to multidimensional optimal control*, to appear in International Journal of Control, 2004.

We first want to optimize the trajectory of the system defined by *R2* $(z[1] : ... : z[4] : v)^T = 0$ with respect to the quadratic criterion formed by the integration the differential form $\frac{1}{2}\ v^2(\mathrm{t})\ \mathrm{dt}$ between 0 and 1. We first need to define the cost matrix Q which is such that

$$v^2 = (z[1] : \ldots : z[4] : v)\, Q\, (z[1] : \ldots : z[4] : v)^T :$$

```
> q := evalm([[1]]);
```

$$q := \begin{bmatrix} 1 \end{bmatrix}$$

```
> r := evalm([seq([0$4], i=1..4)]);
```

$$r := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Then, the corresponding matrix Q is defined by:

```
> Q := evalm(linalg[diag](r, q));
```

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

As the system $R2$ $(z[1] : ... : z[4] : v)^T = 0$ is parametrizable, we can substitute the parametrization of the system $(z[1] : ... : z[4] : v)^T = P2\ \xi[1]$ into the Lagrangian in order to transform the constrained variational problem into a free/unconstrained one. Hence, instead of the Pontryagin approach, we only need to solve a classical variational problem by computing the Euler-Lagrange equations for the new problem. This last computation can be done by using the command *LQEquations*.

```
> E1 := LQEquations(R2, Q, Alg);
```

$$E1 := \left[\begin{array}{c} \left[\begin{array}{c} \frac{d^8}{dt^8}\,\xi_1(t) \end{array}\right], \\ -\%1\,(\frac{d^7}{dt^7}\,\xi_1(t)) - (\frac{d^2}{dt^2}\,\%1)\,(\frac{d^5}{dt^5}\,\xi_1(t)) + (\frac{d}{dt}\,\%1)\,(\frac{d^6}{dt^6}\,\xi_1(t)) + (\frac{d^3}{dt^3}\,\%1)\,(\frac{d^4}{dt^4}\,\xi_1(t)), \\ \left[\begin{array}{c} \xi_1(t) \\ \frac{d}{dt}\,\xi_1(t) \\ \frac{d^2}{dt^2}\,\xi_1(t) \\ \frac{d^3}{dt^3}\,\xi_1(t) \\ \frac{d^4}{dt^4}\,\xi_1(t) \end{array}\right] \end{array}\right]$$

$$\%1 := \delta_{\xi_1}(t)$$

The first matrix $E1[1]$ corresponds to the Euler-Lagrange equations in the parameter $\xi[1]$ of the parametrization of the controllable system.

```
> E1[1];
```

$$\left[\begin{array}{c} \frac{d^8}{dt^8}\,\xi_1(t) \end{array}\right]$$

The parametrization of the system is given by the third matrix $E1[3]$.

```
> E1[3];
```

$$\left[\begin{array}{c} \xi_1(t) \\ \frac{d}{dt}\,\xi_1(t) \\ \frac{d^2}{dt^2}\,\xi_1(t) \\ \frac{d^3}{dt^3}\,\xi_1(t) \\ \frac{d^4}{dt^4}\,\xi_1(t) \end{array}\right]$$

Finally, the second matrix $E1[2]$ gives the boundary terms coming from the integrations by parts.

```
> E1[2];
```

$$-\%1\,(\frac{d^7}{dt^7}\,\xi_1(t)) - (\frac{d^2}{dt^2}\,\%1)\,(\frac{d^5}{dt^5}\,\xi_1(t)) + (\frac{d}{dt}\,\%1)\,(\frac{d^6}{dt^6}\,\xi_1(t)) + (\frac{d^3}{dt^3}\,\%1)\,(\frac{d^4}{dt^4}\,\xi_1(t))$$

$$\%1 := \delta_{\xi_1}(t)$$

Therefore, depending on the problem (e.g., fixed terminal position, free terminal position, terminal error...), we can easily obtain the corresponding intial and final conditions. Note that we also need to translate the initial conditions on $z[1]$, ..., $z[4]$ to initial conditions on $\xi[1]$.

For instance, let us find the optimal trajectory passing from the initial point $\mathrm{x}(0) = 1$, $\mathrm{D}\,\mathrm{x}(0) = 0$, $\theta(0) = 0$, $\mathrm{D}\,\theta(0) = 0$ to the terminal point $\mathrm{x}(1) = 0$, $\mathrm{D}\ \mathrm{x}(1) = 0$, $\theta(1) = 0$, $\mathrm{D}\,\theta(1) = 0$. We easily check that the corresponding conditions yield the following initial and final conditions:

```
> IC := {xi[1](0)=1,xi[1](1)=0,seq((D@@i)(xi[1])(0)=0, i=1..3),
> seq((D@@i)(xi[1])(1)=0, i=1..3)};
```

$$IC := \{\xi_1(0) = 1,\ \xi_1(1) = 0,\ (\mathrm{D}^{(2)})(\xi_1)(0) = 0,\ \mathrm{D}(\xi_1)(0) = 0,\ (\mathrm{D}^{(2)})(\xi_1)(1) = 0,\ (\mathrm{D}^{(3)})(\xi_1)(1) = 0,\ (\mathrm{D}^{(3)})(\xi_1)(0) = 0,\ \mathrm{D}(\xi_1)(1) = 0\}$$

Hence, we integrate symbolically the corresponding Euler-Lagrange equations with the previous conditions. We obtain:

```
> ODEsol := dsolve({E1[1][1,1]=0} union IC, xi[1](t));
```

$$ODEsol := \xi_1(t) = 1 - 35\,t^4 + 84\,t^5 - 70\,t^6 + 20\,t^7$$

In particular, we find again the trajectory *Path* previously computed. This trajectory is optimal for the Lagrangian defined by the form $\frac{1}{2}\ v^2(t)\ dt$ between $t = 0$ to $t = 1$.

Finally, the value of the cost is given by.

```
> Cost := simplify(int(P[3,1]^2, t=0..1)/2);
```

$$Cost := \frac{140}{11} + \frac{12600}{11}\,m + \frac{566860}{11}\,m^2$$

Now, let us study the problem of minimizing the previous Lagrangian but without any constraint on the terminal point at $t = 1$. Using *E2*[3], we easily check that we must impose the following initial and final conditions:

```
> IC2 := {xi[1](0)=1,seq((D@@i)(xi[1])(0)=0, i=1..7),
> seq((D@@i)(xi[1])(1)=0, i=4..7)};
```

$$IC2 := \{\xi_1(0) = 1,\ (\mathrm{D}^{(2)})(\xi_1)(0) = 0,\ \mathrm{D}(\xi_1)(0) = 0,\ (\mathrm{D}^{(3)})(\xi_1)(0) = 0,\ (\mathrm{D}^{(6)})(\xi_1)(1) = 0,\ (\mathrm{D}^{(4)})(\xi_1)(1) = 0,\ (\mathrm{D}^{(5)})(\xi_1)(1) = 0,\ (\mathrm{D}^{(7)})(\xi_1)(1) = 0,\ (\mathrm{D}^{(4)})(\xi_1)(0) = 0,\ (\mathrm{D}^{(5)})(\xi_1)(0) = 0,\ (\mathrm{D}^{(6)})(\xi_1)(0) = 0,\ (\mathrm{D}^{(7)})(\xi_1)(0) = 0\}$$

Then, we integrate symbolically the corresponding ordinary differential equation. We obtain:

```
> ODEsol2 := dsolve({E1[1][1,1]=0} union IC2, xi[1](t));
```

$$ODEsol2 := \xi_1(t) = 1$$

Therefore, the new optimal trajectory becomes:

```
> evalm([[x2_opt(t)],[theta2_opt(t)],[u2_opt(t)]])=ApplyMatrix(Ext[3], [1], Alg);
```

$$\begin{bmatrix} \mathrm{x2\_opt}(t) \\ \mathrm{theta2\_opt}(t) \\ \mathrm{u2\_opt}(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Hence, the new optimal trajectory means to stay at rest. Indeed, the value of the cost is now 0:

```
> Cost2 := int(ApplyMatrix(Ext[3], [1], Alg)[3,1]^2, t=0..1);
```

$$Cost2 := 0$$

Let us now consider the case where there is an additional cost on the state $z[1]$. Let us define the new matrix *Q2* as follows:

```
> r2 := evalm([[gamma,0,0,0],seq([0$4],i=1..3)]):
> Q2 := evalm(linalg[diag](r2, q));
```

$$Q2 := \begin{bmatrix} \gamma & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, we can compute the corresponding Euler-Lagrange equations and boundary terms:

```
> E2 := LQEquations(R2, Q2, Alg):
> E2[1];
```

$$\left[\ \gamma\,\xi_1(t) + \left(\tfrac{d^8}{dt^8}\,\xi_1(t)\right)\ \right]$$

```
> E2[2];
```

$$-\%1\left(\tfrac{d^7}{dt^7}\,\xi_1(t)\right) - \left(\tfrac{d^2}{dt^2}\,\%1\right)\left(\tfrac{d^5}{dt^5}\,\xi_1(t)\right) + \left(\tfrac{d}{dt}\,\%1\right)\left(\tfrac{d^6}{dt^6}\,\xi_1(t)\right) + \left(\tfrac{d^3}{dt^3}\,\%1\right)\left(\tfrac{d^4}{dt^4}\,\xi_1(t)\right)$$
$$\%1 := \delta_{\xi_1}(t)$$

If we study the case where the final time is infinity, then we need to compute the spectral factorization of the differential operator $\mathrm{D}^8 + \gamma$, namely we need to write $\mathrm{D}^8 + \gamma = \mathit{Ftilde} \circ F$, where *Ftilde* denotes the formal adjoint of F and all roots of $F(\mathrm{D}) = 0$ have negative real parts. The roots of the equations $s^8 + 1 = 0$ are given by:

```
> Unity8 := [seq(exp(I*(Pi/8+k*Pi/4)),k=0..7)];
```

$$\mathit{Unity8} := [e^{(1/8\,I\,\pi)},\ e^{(3/8\,I\,\pi)},\ e^{(5/8\,I\,\pi)},\ e^{(7/8\,I\,\pi)},\ e^{(-7/8\,I\,\pi)},\ e^{(-5/8\,I\,\pi)},\ e^{(-3/8\,I\,\pi)},\ e^{(-1/8\,I\,\pi)}]$$

In particular, the real parts of the previous roots are

```
> evalf(map(Re, Unity8));
```

$$[0.9238795325,\ 0.3826834325,\ -0.3826834325,\ -0.9238795325,\ -0.9238795325,$$
$$-0.3826834325,\ 0.3826834325,\ 0.9238795325]$$

Hence, we may only select the roots with negative real parts, namely

```
> Realneg := [seq(Unity8[i], i=3..6)];
```

$$\mathit{Realneg} := [e^{(5/8\,I\,\pi)},\ e^{(7/8\,I\,\pi)},\ e^{(-7/8\,I\,\pi)},\ e^{(-5/8\,I\,\pi)}]$$

or equivalently:

```
> Realnegc := map(evalc, Realneg);
```

$$\mathit{Realnegc} := [-\cos(\tfrac{3\,\pi}{8}) + \sin(\tfrac{3\,\pi}{8})\,I,\ -\cos(\tfrac{\pi}{8}) + \sin(\tfrac{\pi}{8})\,I,\ -\cos(\tfrac{\pi}{8}) - \sin(\tfrac{\pi}{8})\,I,$$
$$-\cos(\tfrac{3\,\pi}{8}) - \sin(\tfrac{3\,\pi}{8})\,I]$$

Then, the spectral factor $F(z)$ of $z^8 + 1$ is defined by:

```
> Pol := simplify(product(z-gamma^(1/8)*Realnegc[i], i=1..4)):
> F := sort(collect(convert(Pol, radical), {gamma,z}), z);
```

$$F := z^4 + (\sqrt{2+\sqrt{2}} + \sqrt{2-\sqrt{2}})\,\gamma^{(1/8)}\,z^3 + (\sqrt{2+\sqrt{2}}\,\sqrt{2-\sqrt{2}} + 2)\,\gamma^{(1/4)}\,z^2 + (\sqrt{2+\sqrt{2}} + \sqrt{2-\sqrt{2}})\,\gamma^{(3/8)}\,z + \sqrt{\gamma}$$

Finally, the feedback law $v = K\,z$ which is such that the dynamics of the closed-loop system

$$D\,z = (F + GK)\,z$$

equals $F(\mathrm{D})\ z[1] = 0$ is defined by:

```
> K := -evalm([seq(coeff(F,z,i), i=0..4)]);
```

$$K := -\left[\sqrt{\gamma},\ (\sqrt{2+\sqrt{2}} + \sqrt{2-\sqrt{2}})\,\gamma^{(3/8)},\ (\sqrt{2+\sqrt{2}}\,\sqrt{2-\sqrt{2}} + 2)\,\gamma^{(1/4)},\ (\sqrt{2+\sqrt{2}} + \sqrt{2-\sqrt{2}})\,\gamma^{(1/8)},\ 1\right]$$