```
<<
 "C:\\Users\\Jamal\\Desktop\\main\\Research\\Codes\\IDOs\\ArtesteinsReduction\\
   TenReS.m"
```

```
Package TenReS version 0.2.4
written by Clemens G. Raab
Copyright 2016, RICAM, Austrian Academy of Sciences
```

A few definitions need to be made by the user. Type ?CoeffQ, ?Specialization, and ?CyclicModules for more information.

Basic definitions

Reduction system for IDOLS

Auxiliary functions

4) Recovering Artstein's Transformation

Consider the differential time-delay control system of the form

$$\dot{x}(t) = A_0(t)\,x(t) + B_0(t)\,u(t) + B_1(t)\,u(t-h) \tag{1}$$

which can be written in terms of blocks as

$$\begin{pmatrix} I_n & 0 \end{pmatrix} \begin{pmatrix} \dot{x}(t) \\ \dot{u}(t) \end{pmatrix} = \begin{pmatrix} A(t) & B_0(t) \end{pmatrix} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} + \begin{pmatrix} 0 & B_1(t) \end{pmatrix} \begin{pmatrix} x(t-h) \\ u(t-h) \end{pmatrix}. \tag{2}$$

We show how we can use our package to find by ansatz a transformation from the system above to the following differential system

$$\dot{z}(t) = E(t)\,z(t) + F(t)\,v(t), \tag{3}$$

which can be expressed in terms of blocks as

$$\begin{pmatrix} I_n & 0 \end{pmatrix} \begin{pmatrix} \dot{z}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} E(t) & F(t) \end{pmatrix} \begin{pmatrix} z(t) \\ v(t) \end{pmatrix}. \tag{4}$$

Corresponding to the systems (2) and (4) we consider the following operators

$$R' = R'_0 \cdot \partial + R'_1 + R'_2 \cdot \delta, \qquad R = R_0 \cdot \partial + R_1, \qquad (5), (6)$$

where the coefficients have the block structures as

$$R'_0 = \begin{pmatrix} I_n & 0 \end{pmatrix},\ R'_1 = \begin{pmatrix} -A & -B_0 \end{pmatrix},\ R'_2 = \begin{pmatrix} 0 & -B_1 \end{pmatrix},\ R_0 = \begin{pmatrix} I_n & 0 \end{pmatrix},\ R_1 = \begin{pmatrix} -E & -F \end{pmatrix}.$$

Our goal is to find the operators P and Q such that

$$R \cdot P = Q \cdot R'. \tag{7}$$

For the transformation we choose $Q = Q_0$ where Q_0 is a multiplication operator and P as the simplified ansatz

$$P = P_0 \cdot \delta \cdot \int \cdot P_1 + P_2 \cdot \int \cdot P_3 + P_4 \cdot \delta + P_5. \qquad (8)$$

Moreover, the operators P_0, P_1, P_2, P_3, P_4, and P_5 have undetermined blocks P_{11}, P_{22}, a_0, a_1, a_2, a_3, a_4, and a_5 as follows

$$P_0 = \begin{pmatrix} a_0 \\ 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0 & a_1 \end{pmatrix}, \quad P_2 = \begin{pmatrix} a_2 \\ 0 \end{pmatrix}, \quad P_3 = \begin{pmatrix} 0 & a_3 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 0 & a_4 \\ 0 & 0 \end{pmatrix}, \quad P_5 = \begin{pmatrix} P_{11} & a_5 \\ 0 & P_{22} \end{pmatrix}.$$

First we need to introduce to our package the coefficients R'_0, R'_1, R'_2, P_0, P_1, P_2, P_3, P_4, P_5, Q_0, and Φ which we use as elements of the coefficient ring R, the parameter h and the notation $\delta = \sigma_{1,h}$.

```
MemberQR[ OR_Integer | ORR_Integer | P_Integer | Q_Integer | Φ] := True
MemberQZ[h] := True
δ := S[1, h]
```

1) Normal form computations for the lhs and the rhs of $R \cdot P = Q \cdot R'$.

```
R1 := Prod[OR0, Diff] + Prod[OR1]
P1 := Prod[P0, δ, Int, P1] + Prod[P2, Int, P3] + Prod[P4, δ] + Prod[P5]
LHS := Prod[R1, P1]
```

```
(LeftSide = ApplyRules[LHS, RedSys]) // Coefficients // TableForm
```

```
Prod[]                    mul[OR0, Diff[P5]] + mul[OR1, P5] + mul[OR0, P2, P3]
Prod[Diff]                mul[OR0, P5]
Prod[S[1, h]]             mul[OR0, Diff[P4]] + mul[OR1, P4] + mul[OR0, P0, S[1, h][P1]]
Prod[Int, P3]             mul[OR0, Diff[P2]] + mul[OR1, P2]
Prod[S[1, h], Diff]       mul[OR0, P4]
Prod[S[1, h], Int, P1]    mul[OR0, Diff[P0]] + mul[OR1, P0]
```

```
Q1  := Prod[Q0]
RR1 := Prod[ORR0, Diff] + Prod[ORR1] + Prod[ORR2, δ]
RHS := Prod[Q1, RR1]
```

```
(RightSide = ApplyRules[RHS, RedSys]) // Coefficients // TableForm
```

```
Prod[]            mul[Q0, ORR1]
Prod[Diff]        mul[Q0, ORR0]
Prod[S[1, h]]     mul[Q0, ORR2]
```

2) Extracting conditions by coefficient comparison.

```
BlockStructure :=
 {OR0 → ( 1  0 ), OR1 → ( -EE  -FF ), ORR0 → ( 1  0 ), ORR1 → ( -A  -B0 ),
  ORR2 → ( 0  -B1 ), P0 → (a0; 0), P1 → ( 0  a1 ), P2 → (a2; 0), P3 → ( 0  a3 ), P4 → (0  a4; 0  0),
  P5 → (P11  a5; 0  P22), Q0 → {{Q0}}}
```

```
Coefficients[LeftSide - RightSide][[All, 2]] /. BlockStructure // TableForm
```

```
Diff[P11] - mul[EE, P11] + mul[Q0, A]
Diff[a5] - mul[EE, a5] - mul[FF, P22] + mul[a2, a3] + mul[Q0, B0]
P11 - Q0
a5
0
Diff[a4] - mul[EE, a4] + mul[a0, S[1, h][a1]] + mul[Q0, B1]
Diff[a2] - mul[EE, a2]
0
a4
Diff[a0] - mul[EE, a0]
```

```
equations = DeleteCases[Flatten[%], 0]
```

```
{Diff[P11] - mul[EE, P11] + mul[Q0, A],
 Diff[a5] - mul[EE, a5] - mul[FF, P22] + mul[a2, a3] + mul[Q0, B0],
 P11 - Q0, a5, Diff[a4] - mul[EE, a4] + mul[a0, S[1, h][a1]] + mul[Q0, B1],
 Diff[a2] - mul[EE, a2], a4, Diff[a0] - mul[EE, a0]}
```

3) Solving obtained system and finding *P*.

For solving the obtained equations, we set $a_4 = a_5 = 0$ and let P_{11} be such that the following equation holds:

$$\partial P_{11} - E P_{11} = -Q_0 A.$$

We also set $Q_0 = P_{11}$ and let Φ invertible such that

$$\partial \Phi = E \Phi.$$

Then for arbitrary constants c_0 and c_2 we assume that

$$a_0 = \Phi c_0 \quad \text{and} \quad a_2 = \Phi c_2 .$$

```
Diff[c0 | c2] := 0
Diff[Φ] := mul[EE, Φ]
```

This solves six of the above equations.

```
Solution1 := {a4 → 0, a5 → 0, Q0 → P11,
   Diff[P11] → mul[EE, P11] - mul[P11, A], a0 → mul[Φ, c0], a2 → mul[Φ, c2]}
RemainingEquations = DeleteCases[equations /. Solution1, 0]
```

```
{-mul[FF, P22] + mul[P11, B0] + mul[Φ, c2, a3], mul[P11, B1] + mul[Φ, c0, S[1, h][a1]]}
```

The remaining equations

$$a_0\, \delta a_1 + \partial a_4 - E a_4 = -Q_0 B_1 \quad \text{and} \quad \partial a_5 + a_2 a_3 - E a_5 - F P_{22} = -Q_0 B_0$$

can be written respectively as

$$c_0\, a_1 = -\, \delta^{-1} \Phi P_{11} B_1 \quad \text{and} \quad c_2\, a_3 = -\Phi^{-1} (F P_{22} - P_{11} B_0).$$

We assume that c_0, c_2, a_1, a_3 are such that they satisfy the above equations.

```
Solution2 :=
  {mul[Φ, c0, S[1, h][a1]] → -mul[P11, B1], mul[Φ, c2, a3] → mul[FF, P22] - mul[P11, B0]}
```

After entering these assumptions, our package verifies the remaining equations are solved.

```
RemainingEquations /. Solution2
{0, 0}
```

Considering these assumptions, the operator P can be written as

$$P = -\begin{pmatrix}\Phi\\0\end{pmatrix}\cdot\delta\cdot\int\cdot(0 \quad \delta^{-1}\ \Phi P_{11} B_1) + \begin{pmatrix}\Phi\\0\end{pmatrix}\cdot\int\cdot(0 \quad \Phi^{-1}\ (F P_{22} - P_{11} B_0)) + \begin{pmatrix}P_{11} & 0\\0 & P_{22}\end{pmatrix}.$$

```
Solution := {P0 → (Φ; 0), P1 → ( 0  -mul[inv[δ][mul[inv[Φ], P11, B1]]] ),
   P2 → (Φ; 0), P3 → ( 0  mul[inv[Φ], mul[FF, P22] - mul[P11, B0]] ),
   P4 → (0 0; 0 0), P5 → (P11 0; 0 P22), Q0 → {{P11}}};
OP = Prod[P0, δ, Int, P1] + Prod[P2, Int, P3] + Prod[P4, δ] + Prod[P5] /. Solution
Prod[{{P11, 0}, {0, P22}}] +
 Prod[{{Φ}, {0}}, Int, {{0, mul[inv[Φ], FF, P22] - mul[inv[Φ], P11, B0]}}] +
 Prod[{{Φ}, {0}}, S[1, h], Int,
  {{0, -mul[inv[S[1, -h][Φ]], S[1, -h][P11], S[1, -h][B1]]}}]
```

4) Verifying correctness of the solution.

In the following we prove that for the obtained operators P and Q the identity $R\cdot P = Q\cdot R'$ holds.

```
MemberQR[ A | B_Integer | EE | FF] := True
MemberQR[P11] := True
Diff[P11] := mul[EE, P11] - mul[P11, A]

ApplyRules[LHS - RHS, RedSys] /.
    {OR0 → ( 1  0 ), OR1 → ( -EE  -FF ), ORR0 → ( 1  0 ), ORR1 → ( -A  -B0 ),
     ORR2 → ( 0  -B1 )} /. Solution // Coefficients // TableForm
Prod[]                                                                         0  (
Prod[Int, {{0, mul[inv[Φ], FF, P22] - mul[inv[Φ], P11, B0]}}]                  0
Prod[S[1, h], Int, {{0, -mul[inv[S[1, -h][Φ]], S[1, -h][P11], S[1, -h][B1]]}}] 0
```