CTDSX – a Collection of Benchmark Examples for State-Space Realizations of Continuous-Time Dynamical Systems[1]

Daniel Kreßner, Volker Mehrmann[2], Thilo Penzl[3]

November 1998

[1]This paper describes Version 1.0 of the benchmark collection CTDSX. It presents research results of the European Community BRITE-EURAM III Thematic Networks Programme NICONET (contract number BRRT-CT97-5040) and is distributed by the Working Group on Software WGS. *WGS secretariat*: Mrs. Ida Tassens, ESAT - Katholieke Universiteit Leuven, K. Mercierlaan 94, 3001-Leuven-Heverlee, BELGIUM. This report is also available by anonymous ftp from *wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN1998-9.ps.Z*

[2]Fakultät für Mathematik, TU Chemnitz, D–09107 Chemnitz, Germany. Email: `volker.mehrmann@mathematik.tu-chemnitz.de`

[3]Fakultät für Mathematik, TU Chemnitz, D–09107 Chemnitz, Germany. Email: `thilo.penzl@mathematik.tu-chemnitz.de`

Contents

1 Introduction

In the analysis of numerical methods and their implementation as numerical software it is extremely important to test the correctness of the implementation as well as the performance of the method. This validation is one of the major steps in the construction of a software library, in particular, if this library is used in practical applications.

In order to carry out such tests it is necessary to have tools that yield an evaluation of the performance of the method as well as the implementation with respect to correctness, accuracy, and speed. Similar tools are needed to compare different numerical methods, to test their robustness, and also to analyze the behaviour of the methods in extreme situations, where the limit of the possible accuracy is reached.

In many application areas benchmark collections have been created that can partially serve for this purpose. Such collections are heavily used. In order to have a fair evaluation and a comparison of methods and software, there should be a standardized set of examples, which are freely available and on which newly developed methods and their implementations can be tested. Moreover, public benchmark collections can be used by developers of algorithms and software as a reference when reporting the results of numerical experiments in publications.

In order to make such collections useful it is important that they cover a wide range of problems. Two kinds of test problems are of particular interest. First, benchmark collections should contain so-called 'real world' examples, i.e., examples reflecting current problems in applications. Second, they must contain test examples which drive numerical methods and their implementations to a limit. These are ideal test cases because errors and failures usually occur only in extreme cases and these are often not covered by standard software validation procedures.

Whereas plenty of test problems for basic linear algebra problems (e.g., systems of linear equations, eigenvalue problems) are provided in several collections, e.g., [29], benchmark examples for control problems are currently harder to find. However, in the last few years collections for control problems containing relatively few test examples have been implemented. In particular, the collection for control system design [13] by E. Davison and the collection for continuous-time Riccati equations [7] by P. Benner, A. Laub, and V. Mehrmann should be mentioned here. Since both collections have been partly included into the benchmark collection described here, a lot of credit ought to be given to their authors.

In Sections 2–5, we provide test examples of time-invariant, continuous-time *descriptor systems*

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) && (1) \\ y(t) &= Cx(t) + Du(t), && (2) \end{aligned}$$

where $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$. However, a considerable part of these examples actually represent *standard systems*, i.e., systems where $E = I_n$. Note that we call the vector-valued functions u, x, and y the *input*, the *state*, and the *output* of the system, respectively.

The examples of our collection are subdivided into four different *groups*:

- Group 1 — parameter-free problems of fixed size,

- Group 2 — parameter-dependent problems of fixed size,
- Group 3 — parameter-free examples of scalable size,
- Group 4 — parameter-dependent examples of scalable size.

The examples in the Groups 2 and 4 depend on several parameters, which have a direct impact on the algebraic properties of the descriptor system. The Groups 3 and 4 contain examples which are freely scalable by a so-called scaling parameter. Of course, this parameter can have an (indirect) influence on the algebraic system properties, too. In some of the examples, the parameters are restricted to certain ranges. These are indicated in the description. Moreover, default values are provided for each parameter.

The benchmark collection CTDSX has been implemented in FORTRAN and MATLAB. The FORTRAN codes conform to the implementation and documentation standards of the software library SLICOT [34].

We intend to augment the benchmark collection in the future. Contributions to forthcoming releases are highly appreciated.

2 Parameter-free problems of fixed size (Group 1)

The first two examples present 'academic' test problems of very small scale.

Example 1.1 [24, Example 1] (see also [7, Example 1])

n	m	p
2	1	2

The system matrices are

$$E = I_2, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = I_2, \quad D = 0_{2\times 1}.$$

Example 1.2 [24, Example 2] (see also [7, Example 2])

n	m	p
2	1	1

This is an example of stabilizable-detectable, but uncontrollable-unobservable data. We have the following system matrices:

$$E = I_2, \quad A = \begin{bmatrix} 4 & 3 \\ -\frac{9}{2} & -\frac{7}{2} \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 2 \end{bmatrix}, \quad D = 0.$$

In the sequel we provide several 'real-world' examples. The description of these problems is kept to the minimum necessary information. For the physical, chemical, or engineering background see the given references and the references given therein.

Example 1.3 [6] (see also [15] and [7, Example 3])

n	m	p
4	2	4

Here the system matrices describe a mathematical model of an L-1011 aircraft.

$$E = I_4, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1.89 & 0.39 & -5.53 \\ 0 & -0.034 & -2.98 & 2.43 \\ 0.034 & -0.0011 & -0.99 & -0.21 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 0.36 & -1.6 \\ -0.95 & -0.032 \\ 0.03 & 0 \end{bmatrix}, \quad C = I_4, \quad D = 0_{4\times 2}.$$

Example 1.4 [8] (see also [15] and [7, Example 4])

n	m	p
8	2	8

A mathematical model of a binary distillation column with condenser, reboiler, and nine plates is given by

$$E = I_8, \quad A = \begin{bmatrix} -0.991 & 0.529 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.522 & -1.051 & 0.596 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.522 & -1.118 & 0.596 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.522 & -1.548 & 0.718 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.922 & -1.640 & 0.799 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.922 & -1.721 & 0.901 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.922 & -1.823 & 1.021 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.922 & -1.943 \end{bmatrix},$$

$$B = 10^{-3} \times \begin{bmatrix} 3.84 & 4.00 & 37.60 & 3.08 & 2.36 & 2.88 & 3.08 & 3.00 \\ -2.88 & -3.04 & -2.80 & -2.32 & -3.32 & -3.82 & -4.12 & -3.96 \end{bmatrix}^T$$

$$C = I_8, \quad D = 0_{8\times 2}.$$

Example 1.5 [30] (see also [15] and [7, Example 5])

n	m	p
9	3	9

This is the data for a continuous state-space model of a tubular ammonia reactor. It should be noted that the underlying model includes a disturbance term which is neglected here.

$$E = I_9, \quad A = \begin{bmatrix} -4.019 & 5.12 & 0 & 0 & -2.082 & 0 & 0 & 0 & 0.87 \\ -0.346 & 0.986 & 0 & 0 & -2.34 & 0 & 0 & 0 & 0.97 \\ -7.909 & 15.407 & -4.069 & 0 & -6.45 & 0 & 0 & 0 & 2.68 \\ -21.816 & 35.606 & -0.339 & -3.87 & -17.8 & 0 & 0 & 0 & 7.39 \\ -60.196 & 98.188 & -7.907 & 0.34 & -53.008 & 0 & 0 & 0 & 20.4 \\ 0 & 0 & 0 & 0 & 94.0 & -147.2 & 0 & 53.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 94.0 & -147.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12.8 & 0 & -31.6 & 0 \\ 0 & 0 & 0 & 0 & 12.8 & 0 & 0 & 18.8 & -31.6 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.010 & 0.003 & 0.009 & 0.024 & 0.068 & 0 & 0 & 0 & 0 \\ -0.011 & -0.021 & -0.059 & -0.162 & -0.445 & 0 & 0 & 0 & 0 \\ -0.151 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \quad C = I_9, \quad D = 0_{9\times 3}.$$

Example 1.6 [14] (see also [7, Example 6])

n	m	p
30	3	5

This control problem for a J–100 jet engine is a special case of a multivariable servomechanism problem. The system state is given by the state of the jet engine denoted by $x^{(1)} \in \mathbb{R}^{16}$, the actuators $x^{(2)} \in \mathbb{R}^8$, and the sensors $x^{(3)} \in \mathbb{R}^6$. There are three actuators in this problem: one for the fuel flow (denoted by $x^{(2,1)} \in \mathbb{R}^2$), one for the nozzle jet area ($x^{(2,2)} \in \mathbb{R}^3$), and one for the inlet guide vane position ($x^{(2,3)} \in \mathbb{R}^3$). The fuel flow x^{ff}, nozzle jet area x^{nja}, and inlet guide vane positions x^{igvp} themselves are given by the first component of the corresponding state variables, i.e.,

$$x^{ff} = x_1^{(2,1)}, \qquad x^{nja} = x_1^{(2,2)}, \qquad x^{igvp} = x_1^{(2,3)}.$$

The dynamics of the system are then given by the following set of equations: The state of the jet engine is described by

$$\dot{x}^{(1)} = A^{(1)}x^{(1)} + A^{(1,2,1)}x^{ff} + A^{(1,2,2)}x^{nja} + A^{(1,2,3)}x^{igvp} + B^{(1)}u^{(1)},$$

where $B^{(1)} = 0$ and

$$A^{(1)}_{(:,1:8)} =$$

$$\begin{bmatrix}
-4.328D+0 & 1.714D-1 & 5.376D+0 & 4.016D+2 & -7.246D+2 & -1.933D+0 & 1.020D+0 & -9.820D-1 \\
-4.402D-1 & -5.643D+0 & 1.275D+2 & -2.335D+2 & -4.343D+2 & 2.659D+1 & 2.040D+0 & -2.592D+0 \\
1.038D+0 & 6.073D+0 & -1.650D+2 & -4.483D+0 & 1.049D+3 & -8.245D+1 & -5.314D+0 & 5.097D+0 \\
5.304D-1 & -1.086D-1 & 1.313D+2 & -5.783D+2 & 1.020D+2 & -9.240D+0 & -1.146D+0 & -2.408D+0 \\
8.476D-3 & -1.563D-2 & 5.602D-2 & 1.573D+0 & -1.005D+1 & 1.952D-1 & -8.804D-3 & -2.110D-2 \\
8.350D-1 & -1.249D-2 & -3.567D-2 & -6.074D-1 & 3.765D+1 & -1.979D+1 & -1.813D-1 & -2.952D-2 \\
6.768D-1 & -1.264D-2 & -9.683D-2 & -3.567D-1 & 8.024D+1 & -8.239D-2 & -2.047D+1 & -3.928D-2 \\
-9.696D-2 & 8.666D-1 & 1.687D+1 & 1.051D+0 & -1.023D+2 & 2.966D+1 & 5.943D-1 & -1.997D+1 \\
-8.785D-3 & -1.636D-2 & 1.847D-1 & 2.169D-1 & -8.420D+0 & 7.003D-1 & 5.666D-2 & 6.623D+0 \\
-1.298D-4 & -2.430D-4 & 2.718D-3 & 3.214D-3 & -1.246D-1 & 1.037D-2 & 8.395D-4 & 9.812D-2 \\
-1.207D+0 & -6.717D+0 & 2.626D+1 & 1.249D+1 & -1.269D+3 & 1.030D+2 & 7.480D+0 & 3.684D+1 \\
-2.730D-2 & -4.539D-1 & -5.272D+1 & 1.988D+2 & -2.809D+1 & 2.243D+0 & 1.794D-1 & 9.750D+0 \\
-1.206D-3 & -2.017D-2 & -2.343D+0 & 8.835D+0 & -1.248D+0 & 9.975D-2 & 8.059D-3 & 4.333D-1 \\
-1.613D-1 & -2.469D-1 & -2.405D+1 & 2.338D+1 & 1.483D+2 & 1.638D+0 & 1.385D-1 & 4.488D+0 \\
-1.244D-2 & 3.020D-2 & -1.198D-1 & -4.821D-2 & 5.575D+0 & -4.525D-1 & 1.981D+1 & 1.249D-1 \\
-1.653D+0 & 1.831D+0 & -3.822D+0 & 1.134D+2 & 3.414D+2 & -2.734D+1 & -2.040D+0 & -6.166D-1
\end{bmatrix}$$

$$A^{(1)}_{(:,9:16)} =$$

$$\begin{bmatrix}
9.990D-1 & 1.521D+0 & -4.062D+0 & 9.567D+0 & 1.008D+1 & -6.017D-1 & -1.312D-1 & 9.602D-2 \\
1.132D+1 & 1.090D+1 & -4.071D+0 & -5.739D-2 & -6.063D-1 & -7.488D-2 & -5.936D-1 & -9.602D-2 \\
-9.389D-3 & 1.352D-1 & 5.638D+0 & 2.246D-2 & 1.797D-1 & 2.407D-2 & 1.100D+0 & 2.743D-2 \\
-3.081D+0 & -4.529D+0 & 5.707D+0 & -2.346D-1 & -2.111D+0 & -2.460D-1 & -4.686D-1 & -3.223D-1 \\
2.090D-3 & -5.256D-2 & -4.077D-2 & -9.182D-3 & -5.178D-2 & 3.425D-2 & 4.995D-3 & -1.256D-2 \\
-1.953D-2 & -1.622D-1 & -6.439D-3 & -2.346D-2 & -2.201D-1 & -2.514D-2 & -3.749D-3 & -3.351D-2 \\
1.878D-2 & -2.129D-1 & -9.337D-3 & -3.144D-2 & -2.919D-1 & -3.370D-2 & 8.873D-2 & -4.458D-2 \\
2.253D-2 & 1.701D-1 & 8.371D-3 & 2.645D-2 & 2.560D-1 & 2.835D-2 & -3.749D-2 & 3.635D-2 \\
-4.999D+1 & 6.760D-2 & 3.946D+1 & 4.991D-3 & 8.983D-2 & 5.349D-3 & 0.000D+0 & 1.372D-2 \\
-6.666D-1 & -6.657D-1 & 5.847D-1 & 6.654D-5 & 1.347D-3 & 7.131D-5 & 0.000D+0 & 2.057D-4 \\
2.854D-1 & 2.332D+0 & -4.765D+1 & 3.406D-1 & 3.065D+0 & 3.624D-1 & -4.343D-1 & 4.681D-1 \\
-9.627D+0 & -9.557D+0 & 3.848D+1 & -5.001D+1 & 1.011D-1 & 1.203D-2 & -4.686D-2 & 1.715D-2 \\
-4.278D-1 & -4.245D-1 & 1.710D+0 & -2.000D+0 & -1.996D+0 & 5.349D-4 & -1.999D-3 & 7.544D-4 \\
-4.414D+0 & -4.354D+0 & 1.766D+1 & -3.113D+0 & -3.018D+0 & -1.977D+1 & -4.999D-2 & 1.509D-2 \\
-1.127D-3 & -6.760D-3 & 1.835D-2 & -9.981D-4 & -1.347D-2 & -1.070D-3 & -2.000D+1 & -2.057D-3 \\
5.004D-1 & -1.437D-1 & -2.416D+0 & -1.073D-1 & -1.078D+0 & 3.053D+1 & 1.989D+1 & -5.016D+1
\end{bmatrix},$$

$$A^{(1,2,1)} = \begin{bmatrix} -4.570D{-2} \\ 1.114D{-1} \\ 2.153D{-1} \\ 3.262D{-1} \\ 9.948D{-3} \\ 2.728D{-2} \\ 1.716D{-2} \\ -7.741D{-2} \\ 3.855D{-2} \\ 5.707D{-4} \\ 5.727D{+0} \\ 1.392D{-1} \\ 6.172D{-3} \\ 6.777D{-2} \\ 1.880D{-3} \\ 1.677D{-1} \end{bmatrix}, \quad A^{(1,2,2)} = \begin{bmatrix} -4.516D{+2} \\ -5.461D{+2} \\ 1.362D{+3} \\ 2.080D{+2} \\ -9.839D{+1} \\ 7.162D{+1} \\ 7.171D{+1} \\ -1.412D{+2} \\ -7.710D{+0} \\ -1.144D{-1} \\ -1.745D{+3} \\ -2.430D{+1} \\ -1.082D{+0} \\ 1.660D{+1} \\ 9.147D{+0} \\ 4.358D{+2} \end{bmatrix}, \quad A^{(1,2,3)} = \begin{bmatrix} -1.058D{+2} \\ -6.575D{+0} \\ 1.346D{+1} \\ -2.888D{+0} \\ 5.069D{-1} \\ 9.608D{+0} \\ 8.571D{+0} \\ -8.215D{-1} \\ -4.371D{-2} \\ -6.359D{-4} \\ -8.940D{+0} \\ -2.736D{-1} \\ -1.183D{-2} \\ 3.980D{-1} \\ -8.241D{-1} \\ -5.994D{+1} \end{bmatrix}.$$

The actuator for the fuel flow is defined by

$$\begin{aligned} \dot{x}^{(2,1)} &= A^{(2,1)}x^{(2,1)} + B^{(2,1)}u^{(2,1)} \\ &= \begin{bmatrix} 0 & 1 \\ -500 & -60 \end{bmatrix} x^{(2,1)} + \begin{bmatrix} 0 \\ 500 \end{bmatrix} u^{(2,1)}, \end{aligned}$$

the nozzle jet area actuator is given by

$$\begin{aligned} \dot{x}^{(2,2)} &= A^{(2,2)}x^{(2,2)} + B^{(2,2)}u^{(2,2)} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3600 & -708 & -106.72 \end{bmatrix} x^{(2,2)} + \begin{bmatrix} 0 \\ 0 \\ 3600 \end{bmatrix} u^{(2,2)}, \end{aligned}$$

and the inlet guide vane position actuator is described by

$$\begin{aligned} \dot{x}^{(2,3)} &= A^{(2,3)}x^{(2,3)} + B^{(2,3)}u^{(2,3)} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -12000 & -5240 & -150 \end{bmatrix} x^{(2,3)} + \begin{bmatrix} 0 \\ 0 \\ 12000 \end{bmatrix} u^{(2,3)}. \end{aligned}$$

Since the actuator states are originally given as third-order differential equations, the first entry of $x^{(2,i)}$, $i = 1, 2, 3$, in the first-order model equations above represents the state of the actuators.

Finally, the sensor state is given by

$$\dot{x}^{(3)} = A^{(3)}x^{(3)} + A^{(3,1)}x^{(1)} + B^{(3)}u^{(3)}$$

$$
= \begin{bmatrix} -33.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & -20 & 0 & 0 & 0 & 0 \\ 0 & 0 & -20 & 0 & 0 & 0 \\ 0 & 0 & 0 & -20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.306 & -1.86 \end{bmatrix} x^{(3)} + \begin{bmatrix} 33.3x_1^{(1)} \\ 20x_2^{(1)} \\ 20x_3^{(1)} \\ 20x_5^{(1)} \\ 0.645(x_{12}^{(1)} + x_{13}^{(1)}) \\ -0.894(x_{12}^{(1)} + x_{13}^{(1)}) \end{bmatrix}.
$$

We can thus write the above equations in the standard form (1) with

$$
E = I_{30}, \quad A = \begin{bmatrix} A^{(1)} & \begin{bmatrix} A^{(1,2,1)} & 0 \end{bmatrix} & \begin{bmatrix} A^{(1,2,2)} & 0 & 0 \end{bmatrix} & \begin{bmatrix} A^{(1,2,3)} & 0 & 0 \end{bmatrix} & 0 \\ 0 & A^{(2,1)} & 0 & 0 & 0 \\ 0 & 0 & A^{(2,2)} & 0 & 0 \\ 0 & 0 & 0 & A^{(2,3)} & 0 \\ A^{(3,1)} & 0 & 0 & 0 & A^{(3)} \end{bmatrix},
$$

$$
B = \begin{bmatrix} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ B^{(2,1)} & 0 & 0 \\ 0 & B^{(2,2)} & 0 \\ 0 & 0 & B^{(2,3)} \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}.
$$

The output (2) of the system is given by

$$
\begin{aligned}
y &= Cx \\
&= \begin{bmatrix}
4.865D{-}1 & 1.383D{-}2 & 0.000D{+}0 & 7.418D{-}5 & 1.538D{-}5 \\
-6.741D{-}1 & 2.789D{-}6 & 0.000D{+}0 & 5.496D{-}6 & 1.201D{-}4 \\
5.392D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 4.790D{-}6 & -2.579D{-}3 \\
9.542D{+}1 & 0.000D{+}0 & 0.000D{+}0 & 1.478D{-}4 & -1.609D{-}4 \\
2.403D{+}1 & -1.081D{-}2 & 0.000D{+}0 & -1.504D{-}2 & 1.618D{-}2 \\
1.052D{+}1 & -5.545D{-}5 & 0.000D{+}0 & -6.503D{-}5 & -1.071D{-}3 \\
8.190D{-}1 & 4.722D{-}5 & 0.000D{+}0 & 8.820D{-}5 & -9.561D{-}5 \\
-4.492D{-}1 & 0.000D{+}0 & 0.000D{+}0 & 4.999D{-}6 & -5.503D{-}6 \\
5.195D{-}1 & 0.000D{+}0 & 0.000D{+}0 & 3.434D{-}6 & -3.732D{-}6 \\
8.437D{-}1 & 0.000D{+}0 & 0.000D{+}0 & 2.727D{-}5 & -2.996D{-}5 \\
-1.863D{+}0 & 0.000D{+}0 & 1.000D{+}0 & 1.128D{-}6 & -1.234D{-}6 \\
5.709D{-}2 & 0.000D{+}0 & 0.000D{+}0 & 4.002D{-}6 & -4.380D{-}6 \\
4.815D{-}1 & 0.000D{+}0 & 0.000D{+}0 & 3.673D{-}5 & -4.024D{-}5 \\
3.428D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 4.290D{-}6 & -4.721D{-}6 \\
2.161D{+}0 & 0.000D{+}0 & 0.000D{+}0 & -4.958D{-}6 & 5.324D{-}6 \\
7.681D{-}2 & 0.000D{+}0 & 0.000D{+}0 & 5.609D{-}6 & -6.103D{-}6 \\
-6.777D{-}2 & 1.282D{-}4 & 0.000D{+}0 & 1.030D{-}6 & 8.109D{-}6 \\
0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 \\
-4.205D{+}2 & 3.353D{-}1 & 0.000D{+}0 & -1.193D{-}2 & 2.328D{-}2 \\
0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 \\
0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 \\
3.297D{+}1 & 6.804D{-}1 & 0.000D{+}0 & -5.806D{-}3 & 1.178D{-}4 \\
0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 & 0.000D{+}0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
0.000D{+}0 & \dots & \dots & \dots & 0.000D{+}0
\end{bmatrix}^T
\begin{bmatrix} x^{(1)} \\ x^{(2,1)} \\ x^{(2,2)} \\ x^{(2,3)} \\ x^{(3)} \end{bmatrix}.
\end{aligned}
$$

Example 1.7 [12] (see also [13, #90–01])

n	m	p
11	3	3

This problem describes a model of a binary distillation column with pressure variation. The equation of a binary distillation column with k plates for a general binary system of components is given in [12]. The following linearized model is obtained for a column containing $k = 8$ plates. The original problem has a disturbance input which is ignored here.

$$E = I_{11}, \qquad A_{(:,1:8)} =$$

$$\begin{bmatrix}
-1.40D{-}02 & 4.30D{-}03 & 0 & 0 & 0 & 0 & 0 & 0 \\
9.50D{-}03 & -1.38D{-}02 & 4.60D{-}03 & 0 & 0 & 0 & 0 & 0 \\
0 & 9.50D{-}03 & -1.41D{-}02 & 6.30D{-}03 & 0 & 0 & 0 & 0 \\
0 & 0 & 9.50D{-}03 & -1.58D{-}02 & 1.10D{-}02 & 0 & 0 & 0 \\
0 & 0 & 0 & 9.50D{-}03 & -3.12D{-}02 & 1.50D{-}02 & 2.20D{-}02 & 0 \\
0 & 0 & 0 & 0 & 2.02D{-}02 & -3.52D{-}02 & 2.20D{-}02 & 0 \\
0 & 0 & 0 & 0 & 0 & 2.02D{-}02 & -4.22D{-}02 & 2.80D{-}02 \\
0 & 0 & 0 & 0 & 0 & 0 & 2.02D{-}02 & -4.82D{-}02 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.02D{-}02 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2.55D{-}02 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$A_{(:,9:11)} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 5.00D{-}04 \\
0 & 0 & 2.00D{-}04 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
3.70D{-}02 & 0 & 2.00D{-}04 \\
-5.72D{-}02 & 4.20D{-}02 & 5.00D{-}04 \\
2.02D{-}02 & -4.83D{-}02 & 5.00D{-}04 \\
0 & 2.55D{-}02 & -1.85D{-}02
\end{bmatrix}$$

$$B = \begin{bmatrix}
0 & 0 & 0 \\
5.00D{-}06 & -4.00D{-}05 & 2.50D{-}03 \\
2.00D{-}06 & -2.00D{-}05 & 5.00D{-}03 \\
1.00D{-}06 & -1.00D{-}05 & 5.00D{-}03 \\
0 & 0 & 5.00D{-}03 \\
0 & 0 & 5.00D{-}03 \\
-5.00D{-}06 & 1.00D{-}05 & 5.00D{-}03 \\
-1.00D{-}05 & 3.00D{-}05 & 5.00D{-}03 \\
-4.00D{-}05 & 5.00D{-}06 & 2.50D{-}03 \\
-2.00D{-}05 & 2.00D{-}06 & 2.50D{-}03 \\
4.60D{-}04 & 4.60D{-}04 & 0
\end{bmatrix}, \qquad C = \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} & 0_{3\times 8} & \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix}, \qquad D = 0_{3\times 3}$$

Example 1.8 [9] (see also [13, #90-02])

n	m	p
9	3	2

This problem describes a fairly realistic model of a drum boiler and has the feature of being multivariable, unstable, and non-minimum phase. The original system [9] has a disturbance input which is ignored here.

$$E = I_9, \qquad A_{(:,1:8)} =$$

$$\left[\begin{array}{rrrrrrrr}
-3.93D{+}00 & -3.15D{-}03 & 0 & 0 & 0 & 4.03D{-}05 & 0 & 0 \\
3.68D{+}02 & -3.05D{+}00 & 3.03D{+}00 & 0 & 0 & -3.77D{-}03 & 0 & 0 \\
2.74D{+}01 & 7.87D{-}02 & -5.96D{-}02 & 0 & 0 & -2.81D{-}04 & 0 & 0 \\
-6.47D{-}02 & -5.20D{-}05 & 0 & -2.55D{-}01 & -3.35D{-}06 & 3.60D{-}07 & 6.33D{-}05 & 1.94D{-}04 \\
3.85D{+}03 & 1.73D{+}01 & -1.28D{+}01 & -1.26D{+}04 & -2.91D{+}00 & -1.05D{-}01 & 1.27D{+}01 & 4.31D{+}01 \\
2.24D{+}04 & 1.80D{+}01 & 0 & -3.56D{+}01 & -1.04D{-}04 & -4.14D{-}01 & 9.00D{+}01 & 5.69D{+}01 \\
0 & 0 & 2.34D{-}03 & 0 & 0 & 2.22D{-}04 & -2.03D{-}01 & 0 \\
0 & 0 & 0 & -1.27D{+}00 & 1.00D{-}03 & 7.86D{-}05 & 0 & -7.17D{-}02 \\
-2.20D{+}00 & -1.77D{-}03 & 0 & -8.44D{+}00 & -1.11D{-}04 & 1.38D{-}05 & 1.49D{-}03 & 6.02D{-}03
\end{array}\right],$$

$$A_{(:,9)} = \left[\begin{array}{r}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1.00D{-}10
\end{array}\right], \qquad B = \left[\begin{array}{rrr}
0 & 0 & 0 \\
0 & 0 & 0 \\
1.56D{+}00 & 0 & 0 \\
0 & -5.13D{-}06 & 0 \\
8.28D{+}00 & -1.50D{+}00 & 3.95D{-}02 \\
0 & 1.78D{+}00 & 0 \\
2.33D{+}00 & 0 & 0 \\
0 & -2.45D{-}02 & 2.84D{-}03 \\
0 & 2.94D{-}05 & 0
\end{array}\right],$$

$$C = \left[\begin{array}{cc} 0_{2\times 5} & \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 1 \end{array}\right] \end{array}\right], \qquad D = 0_{2\times 3}$$

Example 1.9 [26, 27, 28] (see also [13, #90–06])

n	m	p
55	2	2

This plant is an aeroelastic model for a modified Boeing B-767 airplane, at a flutter condition, that was used in a research study of active control technology [27] where a linear-quadratic-Gaussian (LQG)-based design has been developed for flutter control and gust load alleviation. The resulting 55th-order controller has been subsequently reduced to a 10th-order controller for practical implementation using a standard modal residualization technique. A description of the design philosophy and related details on final design performance can be found in [27, 28]. Additional results on controller reduction applied to this problem are published in [26]. A more detailed discussion of this model in context with controller design is given in [13]. Note that the original problem has a disturbance input which is ignored here. For briefness we do not print the matrices A, B, and C. The remaining matrices are $E = I_{55}$ and $D = 0_{2\times 2}$.

Example 1.10 [13, #90-08]

n	m	p
8	2	1

We consider in this problem a control surface servo for an underwater vehicle. The original problem has a disturbance input which is ignored here.

$$E = I_8,$$

$$A = \begin{bmatrix} 0 & 8.5D{+}02 & 0 & 0 & 0 & 0 & 0 & 0 \\ -8.5D{+}02 & -1.2D{+}02 & -4.1D{+}03 & 0 & 0 & 0 & 0 & 0 \\ 3.3D{+}01 & 0 & -3.3D{+}01 & 0 & -7.0D{+}02 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.4D{+}03 & 0 & 0 & 0 \\ 0 & 0 & 1.6D{+}03 & -4.5D{+}02 & -1.1D{+}02 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8.1D{+}01 & 0 & -1.0D{+}00 & 0 & -9.0D{+}2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.1D{+}02 \\ 0 & 0 & 0 & 0 & 0 & 1.2D{+}01 & -1.1D{+}00 & -2.2D{+}01 \end{bmatrix},$$

$$B = \begin{bmatrix} 0_{1\times 2} \\ \begin{bmatrix} 4.6D{+}00 & 9.9D{+}04 \end{bmatrix} \\ 0_{6\times 2} \end{bmatrix}, \qquad C = \begin{bmatrix} 0_{1\times 6} & 1 & 0 \end{bmatrix}, \qquad D = 0_{1\times 2}.$$

3 Parameter-dependent problems of fixed size (Group 2)

Example 2.1 [10] (see also [7, Example 13])

n	m	p	parameter	default value
4	1	2	$\varepsilon \in \mathbb{R} \setminus \{0\}$	10^{-6}

The data of this example describes a magnetic tape control problem.

$$E = I_4, \quad A = \begin{bmatrix} 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0.345 & 0 \\ 0 & -0.524/\varepsilon & -0.465/\varepsilon & 0.262/\varepsilon \\ 0 & 0 & 0 & -1/\varepsilon \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/\varepsilon \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D = 0_{2\times 1}.$$

As $\varepsilon \to 0$, (A, B) gets close to an unstabilizable system.

Example 2.2 [2, Example 2] (see also [7, Example 14])

n	m	p	parameter	default value
4	1	1	$\varepsilon \in \mathbb{R}$	10^{-6}

Here, we have the following system matrices:

$$E = I_4, \quad A = \begin{bmatrix} -\varepsilon & 1 & 0 & 0 \\ -1 & -\varepsilon & 0 & 0 \\ 0 & 0 & \varepsilon & 1 \\ 0 & 0 & -1 & \varepsilon \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^T, \quad D = 0.$$

As $\varepsilon \to 0$, the pairs (A, B) and (A^T, C^T) become unstabilizable.

Example 2.3 [13, #90–05]

n	m	p	parameter	default value
3	1	1	$s \in \{1, \ldots, 10\}$	1

This model describes the control of the vertical acceleration of a rigid guided missile. The missile is open loop stable, but has insufficient damping and is non-minimum phase. There are parameters of 10 different operating conditions labelled by s.

$$E = I_3, \qquad A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 1 & a_{22} & a_{23} \\ 0 & 0 & -190 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 190 \end{bmatrix},$$

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \end{bmatrix}, \qquad D = 0.$$

	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$
a_{11}	-0.327	0.391	-0.688	-0.738	-0.886
a_{12}	-63.94	-130.29	-619.27	-651.57	-1068.85
a_{13}	-155.96	-186.5	-552.9	-604.18	-1004.39
a_{22}	-1.0	-1.42	-2.27	-2.75	-3.38
a_{23}	-0.237	-0.337	-0.429	-0.532	-0.582
c_{11}	0.326	0.35	0.65	0.66	0.79
c_{12}	-208.5	-272.38	-651.11	-913.64	-1926.45
c_{13}	90.93	75.06	283.44	250.5	402.96

	$s=6$	$s=7$	$s=8$	$s=9$	$s=10$
a_{11}	-1.364	-0.333	-0.337	-0.369	-0.402
a_{12}	-92.82	-163.24	-224.03	-253.71	-277.2
a_{13}	-128.46	-153.32	-228.72	-249.87	-419.35
a_{22}	-4.68	-0.666	-0.663	-0.80	-0.884
a_{23}	-0.087	-0.124	-0.112	-0.135	-0.166
c_{11}	1.36	0.298	0.319	0.33	0.36
c_{12}	-184.26	-247.75	-375.75	-500.59	-796.18
c_{13}	76.43	63.77	117.4	103.76	178.59

Example 2.4 [5, 17, 33] (see also [13, #90–07])

n	m	p	parameter	default value
3	1	1	$b \in (9000, 16000)$	14000
			$\mu \in (0.05, 0.3)$	0.1287
			$r \in (0.05, 5)$	0.150
			$r_c \in (0, 0.05)$	0.01
			$k_l \in (0.000103, 0.0035)$	0.002
			$\sigma \in (0.001, 15)$	0.24
			$a \in (10.5, 11.10)$	10.75

This model describes the control of the carriage position of a hydraulic positioning system under different loading conditions. The original problem has a disturbance input which is ignored here. The non-variable parameter v is chosen as $v = 874$.

$$E = I_3, \qquad A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\left(r + \frac{4}{\pi} r_c\right)/\mu & a/\mu \\ 0 & -4ab/v & -4b(\sigma + k_l)/v \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ -4b/v \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad D = 0.$$

Example 2.5 [22] (see also [13, #90–09])

n	m	p	parameter	default value
$2m_s$	m_s	m_s	$s \in \{1, \ldots, 7\}$	1

This problem describes a linearized model of a cascade of inverted pendula, and has the feature of being 'highly' unstable and difficult to control. The difficulties of control become more pronounced as the number of links increase. There are seven examples of the orders $2m_s$, where $m_s = s$ for $s = 1, \ldots, 6$ and $m_s = 10$ for $s = 7$. $E = I_{2m_s}$ and $D = 0_{m_s \times m_s}$ for any s. For briefness we do not display the matrices $A \in \mathbb{R}^{2m_s \times 2m_s}$, $B \in \mathbb{R}^{2m_s \times m_s}$, and $C \in \mathbb{R}^{m_s \times 2m_s}$.

Example 2.6 [3, 20, 21] (see also [13, #90–10])

n	m	p	parameter	default value
3	1	1	$s \in \{1, \ldots, 5\}$	1

This problem deals with the design of a controller associated with the regulation of a ship's heading. It is the linearized model of a ship moving under constant velocity. There are parameter sets for five different classes of vessels. These sets are labeled by the parameter s.

	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$
a_{11}	-0.895	-0.770	-0.597	-0.298	-0.454
a_{12}	-0.286	-0.335	-0.372	-0.279	-0.433
a_{21}	-4.367	-3.394	-3.651	-4.370	-4.005
a_{22}	-0.918	-1.627	-0.792	-0.773	-0.807
b_1	0.108	0.170	0.103	0.116	0.097
b_2	-0.918	-1.627	-0.792	-0.773	-0.807

$$E = I_3, \qquad A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \qquad D = 0.$$

Example 2.7 [1, 11] (see also [13, #90–11])

n	m	p	parameter	default value
5	1	1	$\mu \in [9.95, 16.0]$	15
			$\nu \in [1.0, 20.0]$	10

It is desired to control the lateral motion of a city bus such that the bus follows an electromagnetic track imbedded in the street. Here we provide a linearized model for the lateral motion. Besides the parameters μ and ν the system depends on the following constants.

$$\begin{aligned} l_F &= 3.67, \\ l_R &= 1.93, \\ \delta_F &= 99, \\ \delta_R &= 235, \\ c &= 10.86. \end{aligned}$$

Some entries of the matrix A depend on these values:

$$\begin{aligned} a_{11} &= -2\frac{\delta_F + \delta_R}{\nu\mu}, \\ a_{12} &= -1 - 2\frac{\delta_F l_F - \delta_R l_R}{\nu^2\mu}, \\ a_{21} &= -2\frac{\delta_F l_F - \delta_R l_R}{\mu c}, \\ a_{22} &= -2\frac{\delta_F l_F^2 + \delta_R l_R^2}{\nu\mu c}, \\ a_{15} &= 2\frac{\delta_F}{\nu\mu}, \\ a_{25} &= 2\frac{\delta_F l_F}{\mu c}. \end{aligned}$$

The matrices of the system are:

$$E = I_5, \quad A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & a_{15} \\ a_{21} & a_{22} & 0 & 0 & a_{25} \\ \nu & 0 & 0 & \nu & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

$$B = \begin{bmatrix} 0_{4\times 1} \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 1 & 6.12 & 0 \end{bmatrix}, \quad D = 0.$$

4 Parameter-free examples of scalable size (Group 3)

Example 3.1 [24, Example 4], [4] (see also [7, Example 15])

n	m	p	scaling parameter	default value
$2q-1$	q	$q-1$	$q \in \{1, 2, \ldots\}$	20

The matrices presented here describe a mathematical model of position and velocity control for a string of high-speed vehicles. This problem is also known as “smart highway” or “intelligent highway”. If q vehicles are to be controlled, the size of the system matrices will be $n = 2q - 1$.

$$E = I_{2q-1}, \quad A = \begin{bmatrix} A_{11} & A_{12} & 0 & \ldots & & 0 \\ 0 & A_{22} & A_{23} & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & 0 & A_{q-2,q-2} & A_{q-2,q-1} & 0 \\ & & & 0 & A_{q-1,q-1} & \begin{bmatrix} 0 \\ -1 \end{bmatrix} \\ 0 & \ldots & & & 0 \quad 0 & -1 \end{bmatrix} \in \mathbb{R}^{(2q-1)\times(2q-1)},$$

where

$$\begin{aligned} A_{k,k} &= \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{for} \quad 1 \le k \le q-1, \\ A_{k,k+1} &= \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \quad \text{for} \quad 1 \le k \le q-2, \end{aligned}$$

and

$$B = \begin{bmatrix} 1 & 0 & \ldots & \ldots & 0 \\ 0 & 0 & & & \vdots \\ 0 & 1 & & & \\ 0 & 0 & & & \\ \vdots & \vdots & & & \\ & & & & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(2q-1)\times q}, \quad C = \begin{bmatrix} 0 & 1 & 0 & & & \ldots & 0 \\ 0 & 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \ldots & & \ldots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(q-1)\times(2q-1)},$$

$$D = 0_{(q-1)\times q}.$$

Note that the matrix A has $q-1$ zero eigenvalues.

Remark 4.1 Partial differential equations are an important source of scalable test examples, in particular, if large, sparse examples are sought. In the sequel, the heat equation (provided with a control term)

$$\frac{\partial}{\partial t}\mathbf{x}(\xi,t) = \Delta_\xi \mathbf{x}(\xi,t) + \mathbf{b}(\xi)u(t) \qquad \begin{cases} + \text{ initial conditions} \\ + \text{ boundary conditions} \end{cases} \tag{3}$$

is of interest. Here, $\mathbf{x}$, ξ, and t denote the temperature, the space component, and the time component, respectively. A semidiscretization of (3) w.r.t. ξ with *finite differences* delivers a dynamical system (1), i.e., $E = I_n$. In general, the resulting matrix A is symmetric and negative definite. See also Remark 5.1.

Example 3.2 [19, Example 4.1]

n	m	p	scaling parameter	default value
n	1	n	$n \in \{1,2,\ldots\}$	100

This is a very simple one-dimensional example which describes the control of the heat flow in a thin rod. The input u of the system acts on the boundary of the domain, or more precisely at one of the ends of the rod. The tridiagonal matrix $A \in \mathbb{R}^{n\times n}$ is given by

$$A = \begin{bmatrix} -1/h & 1/h & 0 & \cdots & 0 \\ 1/h & -2/h & 1/h & \ddots & \vdots \\ 0 & 1/h & -2/h & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1/h \\ 0 & \cdots & 0 & 1/h & -2/h \end{bmatrix}$$

where $h = 1/(n+1)$. The remaining matrices are

$$E = I_n, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1/h \end{bmatrix}, \quad C = I_n, \quad D = 0_{n\times 1}.$$

Example 3.3 [24, Example 6] (see also [7, Example 17])

n	m	p	scaling parameter	default value
n	1	1	$n \in \{1,2,\ldots\}$	21

This example describes a system of n integrators connected in series and a feedback controller is supposed to be applied to the nth system. For more details about the physical background see [24].

$$E = I_n, \quad A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & & & 0 \\ 0 & & & 0 & 1 \\ 0 & \dots & & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T, \quad D = 0.$$

Remark 4.2 Another source of large-scale descriptor systems is the control of *second-order models* as described for example in [16, 25]. In this type of problems, the dynamical system is given in terms of a second-order differential equation

$$M\ddot{z} + L\dot{z} + Kz = Su \tag{4}$$

and an associated output

$$y = Nz + P\dot{z} \tag{5}$$

or alternatively

$$\tilde{y} = \begin{bmatrix} N & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix}, \tag{6}$$

where $z \in \mathbb{R}^\ell$, $M, L, K \in \mathbb{R}^{\ell\times\ell}$, $S \in \mathbb{R}^{\ell\times m}$, and $N, P \in \mathbb{R}^{p\times\ell}$. Often, M and K are symmetric where M is positive definite, K is positive semidefinite, and L is the sum of a symmetric positive semidefinite and a skew-symmetric matrix. Usually, M is called the *mass matrix*, L is the *Rayleigh* matrix representing damping (the symmetric part) and gyroscopic (the skew-symmetric part) forces, and K is the *stiffness matrix*. Second-order models are often used to model mechanical systems such as large flexible space structures.

A first-order realization of this problem may be obtained by introducing the state vector

$$x = \begin{bmatrix} z \\ \dot{z} \end{bmatrix}.$$

This yields a system of the form

$$\begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \dot{x} = \begin{bmatrix} 0 & I \\ -K & -L \end{bmatrix} x + \begin{bmatrix} 0 \\ S \end{bmatrix} u \tag{7}$$

$$y = [\, N \;\; P \,]\, x, \tag{8}$$

or, with (6),

$$\tilde{y} = \begin{bmatrix} N & 0 \\ 0 & P \end{bmatrix} x. \tag{9}$$

This is a standard system as in (1) with $n = 2\ell$.

Example 3.4 [31, 35] (see also [7, Example 20])

n	m	p	scaling parameter	default value
$2\ell-1$	ℓ	ℓ	$\ell \in \{1,\ldots,211\}$	211

The size of this example is scalable over a wide range, but not freely scalable. However, the structure of the example is similar to that of the test examples in Group 4.

The example describes a problem arising in power plants. We consider a model of a rotating axle with several masses placed upon it. These masses may be parts of turbines or generators and are assumed to be symmetric with respect to the axle. The input to the system consists of changing loads which act on the masses. This causes vibrations in the axle. The aim is to minimize the moments between two neighboring masses in order to maximize the life expectancy of the axle. The system matrices in (4) and (5) are given as

$$M = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_\ell \end{bmatrix}, \qquad K = \begin{bmatrix} \kappa_1 & -\kappa_1 & & & \\ -\kappa_1 & \kappa_1+\kappa_2 & -\kappa_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -\kappa_{n-2} & \kappa_{n-2}+\kappa_{n-1} & -\kappa_{n-1} \\ & & & -\kappa_{n-1} & \kappa_{n-1} \end{bmatrix},$$

$$L = \begin{bmatrix} \delta_1+\gamma_1 & -\gamma_1 & & & \\ -\gamma_1 & \gamma_1+\delta_2+\gamma_2 & -\gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -\gamma_{\ell-2} & \gamma_{\ell-2}+\delta_{\ell-1}+\gamma_{\ell-1} & -\gamma_{\ell-1} \\ & & & -\gamma_{\ell-1} & \gamma_{\ell-1}+\delta_\ell \end{bmatrix}, \qquad S = I_\ell,$$

$$N = \begin{bmatrix} 0 & 0 & & \\ \kappa_1 & -\kappa_1 & & \\ & \ddots & \ddots & \\ & & \kappa_{\ell-1} & -\kappa_{\ell-1} \end{bmatrix}, \qquad P = \begin{bmatrix} 1 & 0 & & \\ \gamma_1 & -\gamma_1 & & \\ & \ddots & \ddots & \\ & & \gamma_{\ell-1} & -\gamma_{\ell-1} \end{bmatrix}.$$

Hence, the mathematical model of this problem is defined by ℓ and the parameter vectors

- $\mu \in \mathbb{R}^\ell$ — the moments of inertia of the masses,
- $\delta \in \mathbb{R}^\ell$ — the outer damping forces,
- $\gamma \in \mathbb{R}^{\ell-1}$ — the damping forces between two neighboring masses, and
- $\kappa \in \mathbb{R}^{\ell-1}$ — the spring constants of the axle part between two neighboring masses.

The resulting system is neither observable nor detectable. We may overcome this problem by eliminating the unobservable state variable as follows.

At first, a linear transformation in the state space is performed. It is known that such a transformation preserves the system properties (i.e., controllability, observability, stabilizability, detectability) if the transformation matrix is regular; see, e.g., [36].

As transformation matrix we choose

$$T = \begin{bmatrix} 0 & \hat{T} \\ \hat{T} & 0 \end{bmatrix} \in \mathbb{R}^{2\ell\times 2\ell},$$

where $\hat{T} \in \mathbb{R}^{\ell \times \ell}$ is the lower triangular matrix

$$\hat{T} = \begin{bmatrix} 1 & 0 & & & \\ 1 & -1 & & & \\ 1 & -1 & -1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ 1 & -1 & -1 & \dots & -1 \end{bmatrix}.$$

The inverse of T is

$$T^{-1} = \begin{bmatrix} 0 & \hat{T}^{-1} \\ \hat{T}^{-1} & 0 \end{bmatrix}, \qquad \hat{T}^{-1} = \begin{bmatrix} 1 & & & & \\ 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix}.$$

The resulting system corresponding to (7) is then given by

$$\begin{aligned} \hat{E}\dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u, \\ y &= \hat{C}\hat{x}, \end{aligned}$$

where $\hat{x} = T^{-1}x$ and

$$\begin{aligned} \hat{E} = T^{-1} \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} T &= \begin{bmatrix} \hat{T}^{-1} M \hat{T} & 0 \\ 0 & I \end{bmatrix}, \\ \hat{A} = T^{-1} \begin{bmatrix} 0 & I \\ -K & -L \end{bmatrix} T &= \begin{bmatrix} -\hat{T}^{-1} L \hat{T} & -\hat{T}^{-1} K \hat{T} \\ I & 0 \end{bmatrix}, \\ \hat{B} = T^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} &= \begin{bmatrix} \hat{T}^{-1} \\ 0 \end{bmatrix}, \\ \hat{C} = [\, N \;\; P \,] T &= [\, P\hat{T} \;\; N\hat{T} \,]. \end{aligned} \tag{10}$$

Now the $(\ell+1)$st columns of $\hat{A}$ and $\hat{C}$ are zero, that is, the $(\ell+1)$st component of $\hat{x}$ is the undetectable state variable. We thus obtain a stabilizable/detectable system with the same input/output behavior as (10) by removing this component from the system. This is equivalent to removing the $(\ell+1)$st columns of $\hat{E}$, $\hat{A}$, and $\hat{C}$ and the $(\ell+1)$st rows of $\hat{E}$, $\hat{A}$, and $\hat{B}$. The resulting system matrices are $E, A \in \mathbb{R}^{(2\ell-1)\times(2\ell-1)}$, $B \in \mathbb{R}^{(2\ell-1)\times\ell}$, and $C \in \mathbb{R}^{\ell\times(2\ell-1)}$. As values for the physical parameters we use data provided by [23] corresponding to a generator axle in a power plant. The dimension of the problem ($n = 2\ell - 1 = 421$) prevents printing the data. If a system with $l < 211$ is desired, then the parameter vectors μ, δ, γ, and κ are truncated when generating the data.

5 Parameter-dependent examples of scalable size (Group 4)

Remark 5.1 In Remark 4.1 we have considered dynamical systems arising from the discretization of parabolic differential equations by means of finite differences. Alternatively, *finite elements* can be used for a semidiscretization of (3) w.r.t. ξ. This results in a descriptor system (1), where $E \neq I_n$ is the *mass matrix* and A is the *stiffness matrix*.

Example 4.1 [32] (see also [7, Example 18])

n	m	p	scaling parameter	default value	parameter	default value
n	1	1	$n \in \{1, 2, \ldots\}$	100	$a \in (0, \infty)$	0.01
					$b \in \mathbb{R}$	1.0
					$c \in \mathbb{R}$	1.0
					$\beta_1 \in [0, 1)$	0.2
					$\beta_2 \in (\beta_1, 1]$	0.3
					$\gamma_1 \in [0, 1)$	0.2
					$\gamma_2 \in (\gamma_1, 1]$	0.3

The data of this example come from a control problem of one-dimensional heat flow in the domain $\Omega = (0, 1)$. The standard finite element approach based on linear B–splines is used to generate the descriptor system (1). If the interval Ω is subdivided into $n + 1$ subintervals of length $h = 1/(n + 1)$, then with this approach we obtain a system of order n. The mass matrix $E \in \mathbb{R}^{n\times n}$ and the stiffness matrix $A \in \mathbb{R}^{n\times n}$ are given by

$$E = \frac{1}{6(n+1)} \begin{bmatrix} 4 & 1 & 0 & & \ldots & 0 \\ 1 & 4 & 1 & & & \\ & \ddots & \ddots & \ddots & & \vdots \\ & & & 1 & 4 & 1 \\ 0 & \ldots & & & 1 & 4 \end{bmatrix}, \quad A = -a(n+1) \begin{bmatrix} 2 & -1 & 0 & & \ldots & 0 \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \vdots \\ & & & -1 & 2 & -1 \\ 0 & \ldots & & & -1 & 2 \end{bmatrix},$$

whereas $B, C^T \in \mathbb{R}^{n\times 1}$ are given by

$$\begin{aligned} (B)_i &= \int_0^1 \beta(\xi)\varphi_i^n(\xi)d\xi, \qquad i = 1, \ldots, n, \\ (C)_i &= \int_0^1 \gamma(\xi)\varphi_i^n(\xi)d\xi, \qquad i = 1, \ldots, n. \end{aligned}$$

Here $\{\varphi_i^n\}_{i=1}^n$ with

$$\varphi_i^n(\xi) := \begin{cases} (n+1)(\xi - (i-1)h) & : \ \xi \in ((i-1)h, ih) \\ (n+1)((i+1)h - \xi) & : \ \xi \in (ih, (i+1)h) \\ 0 & : \ \text{otherwise} \end{cases}$$

is the B–spline basis for the chosen finite-dimensional subspace of the underlying Hilbert space. The functions $\beta, \gamma \in L_2(0, 1)$ used here are defined by

$$\begin{aligned} \beta(\xi) &= \begin{cases} b, & \xi \in [\beta_1, \beta_2] \\ 0, & \text{otherwise} \end{cases}, \\ \gamma(\xi) &= \begin{cases} c, & \xi \in [\gamma_1, \gamma_2] \\ 0, & \text{otherwise} \end{cases}. \end{aligned}$$

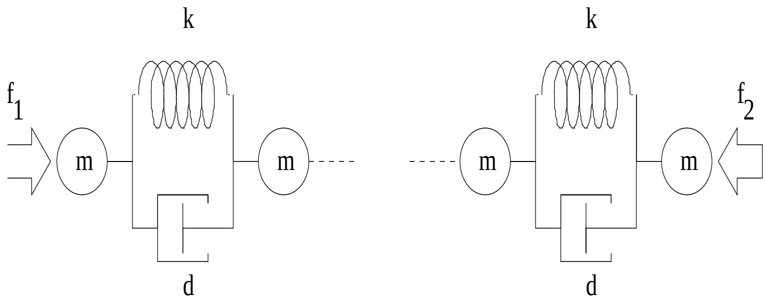
The matrix D is chosen to be zero. Besides the system dimension n, the problem has the parameters a, b, c, β_1, β_2, γ_1, and γ_2. The default values given in the table are taken from [32]. Any other parameter combination may be used for generating the data.

Example 4.2 [18, Example 3] (see also [7, Example 19])

n	m	p	scaling parameter	default value	parameter	default value
2ℓ	2	2ℓ	$\ell \in \{1, 2, \ldots\}$	30	$\mu \in (0, \infty)$	4.0
					$\delta \in (0, \infty)$	4.0
					$\kappa \in (0, \infty)$	1.0

This example represents a second order model, which has been described in Remark 4.2. We refer to the notation introduced there. The example is a model of a string consisting of coupled springs, dashpots, and masses as shown in Figure 1. The inputs are two forces, one acting on the left end of the string, the other one on the right end.

Figure 1: Coupled Spring Experiment ($k \sim \kappa$, $m \sim \mu$, $d \sim \delta$)



For this problem, the matrices in (4) and (6) are

$$M = \mu I_\ell, \qquad L = \delta I_\ell, \qquad N = P = I_\ell,$$

$$K = \kappa \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 & 0 \\ -1 & 2 & -1 & \ldots & 0 & 0 \\ 0 & -1 & 2 & \ldots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ldots \\ 0 & 0 & \ldots & -1 & 2 & -1 \\ 0 & 0 & \ldots & 0 & -1 & 1 \end{bmatrix}, \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & 0 \\ 0 & -1 \end{bmatrix}.$$

The matrix D is chosen to be zero.

Acknowledgements

The authors wish to thank P. Benner and E. Davison (IFAC), who considerably supported the development of the benchmark collection CTDSX.

References

[1] J. Ackermann. The track-guided bus - parametric modelling of its dynamics and analysis of the robust control problem. Technical Report DLR-Report 515-89-15, DLR, 1989.

[2] W.F. Arnold and A.J. Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proc. IEEE*, 72:1746–1754, 1984.

[3] K.J. Astrom and C.G. Kallstrom. Identification of ship steering dynamics. *Automatica*, 12:9–22, 1976.

[4] M. Athans, W.S. Levine, and A. Levis. A system for the optimal and suboptimal position and velocity control for a string of high-speed vehicles. In *Proc. 5th Int. Analogue Computation Meetings*, Lausanne, Switzerland, September 1967.

[5] H. Baumgartner. *Die Nachlaufgenauigkeit des Hydraulischen Zylinderantriebes im Maschinenbau.* PhD thesis, ETH Zürich, 1977. In German.

[6] S. Beale and B. Shafai. Robust control system design with a proportional-integral observer. *Internat. J. Control*, 50:97–111, 1989.

[7] P. Benner, A.J. Laub, and V. Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: Continuous-time case. Preprint SPC95-22, Technische Universität Chemnitz, Chemnitz, Germany, 1995.

[8] S. Bhattacharyya, A. Del Nero Gomes, and J. Howze. The structure of robust disturbance rejection control. *IEEE Trans. Automat. Control*, AC-28:874–881, 1983.

[9] K.L. Chien, E.I. Ergin, C. Ling, and A. Lee. Dynamic analysis of a boiler. *Trans. ASME*, 80:1809–1819, 1958.

[10] J. Chow and P. Kokotovic. A decomposition of near–optimum regulators for systems with slow and fast modes. *IEEE Trans. Automat. Control*, AC-21:701–705, 1976.

[11] W. Darenberg. Automatische Spurführung von Kraftfahrzeugen. *Automobilindustrie*, pages 155–159, 1987.

[12] E.J. Davison. Control of a distillation column with pressure variation. *Trans. Institute of Chemical Engineers*, 45:229–250, 1967.

[13] E.J. Davison. Benchmark problems in control system design. Report of the IFAC theory committee, International Federation of Automatic Control, May 1990.

[14] E.J. Davison and W. Gesing. The systematic design of control systems for the multivariable servomechanism problem. In M.K Sain, J.L. Peczkowsky, and J.M. Melsa, editors, *Alternatives for Linear Multivariable Control.* Nat. Eng. Consortium Inc., Chicago, IL, 1978.

[15] Z. Gajić and X. Shen. *Parallel Algorithms for Optimal Control of Large Scale Linear Systems.* Springer-Verlag, London, 1993.

[16] J.D. Gardiner. Stabilizing control for second-order models and positive real systems. *AIAA J. Guidance, Dynamics and Control*, 15(1):280–282, 1992.

[17] Q.T. Gu. A robust regulator for a hydraulic positioning system. Technische Hogeschool Twente Memorandum, no. 339.

[18] J.J. Hench, C. He, V. Kučera, and V. Mehrmann. Dampening controllers via a Riccati equation approach. Technical Report SPC 95_18, Fakultät für Mathematik, Technische Universität Chemnitz–Zwickau, 09107 Chemnitz, FRG, May 1995.

[19] A.S. Hodel, K.P. Poolla, and B. Tenison. Numerical solution of the Lyapunov equation by approximate power iteration. *Linear Algebra Appl.*, 236:205–230, 1996.

[20] C.G. Kallstrom and K. J. Astrom. Experiences of system identification applied to ship steering. *Automatica*, 17:187–198, 1981.

[21] C.G. Kallstrom, K. J. Astrom, N. E. Thorell, J. Eriksson, and L. Sten. Adaptive autopilots for tankers. *Automatica*, 15:241–254, 1979.

[22] H. Kwakernaak and H. Westdyk. Regulability of a multiple inverted pendulum system. *Control Theory and Advanced Technology*, 1:1–9, 1985.

[23] W. Lang. Personal communication, 1995.

[24] A.J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Control*, AC-24:913–921, 1979. (see also *Proc. 1978 CDC (Jan. 1979)*, pp. 60-65).

[25] A.J. Laub and J.D. Gardiner. Hypercube implementation of some parallel algorithms in control. In M.J. Denham and A.J. Laub, editors, *Advanced Computing Concepts and Techniques in Control Engineering*, pages 361–390. Springer-Verlag, Berlin, 1988.

[26] Y. Liu, B.D.O. Anderson, and U.L. Ly. Coprime factorization controller reduction with Bezout identity induced frequency weighting. *Automatica*, 26:233–249, 1990.

[27] U.L. Ly and D. Gangsaas. Application of modified linear quadratic Gaussian design to active control of a transport airplane. AIAA, Guidance and Control Conference, Boulder, August 1979.

[28] U.L. Ly, D. Gangsaas, and D.C. Norman. Practical gust load alleviation and flutter suppression control laws based on an LQG methodology. AIAA, Aerospace Sciences Meeting, 1981.

[29] Matrix Market. URL: `http://math.nist.gov/MatrixMarket/`.

[30] L.M. Patnaik, N. Viswanadham, and I.G. Sarma. Computer control algorithms for a tubular ammonia reactor. *IEEE Trans. Automat. Control*, AC-25(4):642–651, 1980.

[31] T. Penzl. Vorkonditionierung bei der iterativen Lösung von Lyapunovgleichungen. Diploma thesis, Technische Universität Chemnitz–Zwickau, Fak. f. Mathematik, 09107 Chemnitz, FRG, July 1994.

[32] I.G. Rosen and C. Wang. A multi–level technique for the approximate solution of operator Lyapunov and algebraic Riccati equations. *SIAM J. Numer. Anal.*, 32(2):514–541, 1995.

[33] M. Senning. Robuste Regelung hydraulischer Vorschubantriebe, 1980. Vortrag Seminar Inst. f. Werkzeugmaschinenbau und Fertigungstechnik, ETH-Z, Zürich.

[34] SLICOT Implementation and Documentation Standards. URL: `http://www.win.tue.nl/wgs/stand.html`.

[35] H. Wilharm. Ein praktisch orientiertes Verfahren der Beobachterauslegung bei schwach gedämpften Mehrmassensystemen mit deterministischen Meßstörungen. *Fortschritt–Berichte der VDI–Zeitschriften*, Reihe 8(64).

[36] W.M. Wonham. *Linear Multivariable Control: A Geometric Approach.* Springer-Verlag, New York, second edition, 1979.

A The FORTRAN 77 subroutine CTDSX

The FORTRAN 77 subroutine ctdsx has the following calling sequence and in-line documentation.

```
      SUBROUTINE CTDSX(DEF, NR, DPAR, IPAR, VEC, N, M, P, E, LDE, A,
     1                 LDA, B, LDB, C, LDC, D, LDD, NOTE, DWORK, LDWORK,
     2                 INFO)
C
C     PURPOSE
C
C     This routine is an implementation of the benchmark library
C     CTDSX (Version 1.0) described in [1].
C
C     It generates benchmark examples for time-invariant,
C     continuous-time dynamical systems
C
C          .
C       E x(t) = A x(t) + B u(t)
C
C         y(t) = C x(t) + D u(t)
C
C     E, A are real N-by-N matrices, B is N-by-M, C is P-by-N, and
C     D is P-by-M. In many examples, E is the identity matrix and D is
C     the zero matrix.
C
C
C     ARGUMENTS
C
C     Mode Parameters
C
C     DEF     CHARACTER*1
C             Specifies the kind of values used as parameters when
C             generating parameter-dependent and scalable examples
C             (i.e., examples with NR(1) = 2, 3, or 4):
C             DEF = 'D' or 'd': Default values defined in [1] are used.
C             DEF = 'N' or 'n': Values set in DPAR and IPAR are used.
C             This parameter is not referenced if NR(1) = 1.
C             Note that the scaling parameter of examples with
C             NR(1) = 3 or 4 is considered as a regular parameter in
C             this context.
C
C     Input/Output Parameters
C
C     NR      (input) INTEGER array, dimension 2
```

```
C            Specifies the index of the desired example according
C            to [1].
C            NR(1) defines the group:
C                  1 : parameter-free problems of fixed size
C                  2 : parameter-dependent problems of fixed size
C                  3 : parameter-free problems of scalable size
C                  4 : parameter-dependent problems of scalable size
C            NR(2) defines the number of the benchmark example
C            within a certain group according to [1].
C
C    DPAR    (input/output) DOUBLE PRECISION array, dimension 7
C            On entry, if DEF = 'N' or 'n' and the desired example
C            depends on real parameters, then the array DPAR must
C            contain the values for these parameters.
C            For an explanation of the parameters see [1].
C            For Examples 2.1 and 2.2, DPAR(1) defines the parameter
C            'epsilon'.
C            For Example 2.4, DPAR(1), ..., DPAR(7) define 'b', 'mu',
C            'r', 'r_c', 'k_l', 'sigma', 'a', respectively.
C            For Example 2.7, DPAR(1) and DPAR(2) define 'mu' and 'nu',
C            respectively.
C            For Example 4.1, DPAR(1), ..., DPAR(7) define 'a', 'b',
C            'c', 'beta_1', 'beta_2', 'gamma_1', 'gamma_2',
C            respectively.
C            For Example 4.2, DPAR(1), ..., DPAR(3) define 'mu',
C            'delta', 'kappa', respectively.
C            On exit, if DEF = 'D' or 'd' and the desired example
C            depends on real parameters, then the array DPAR is
C            overwritten by the default values given in [1].
C
C    IPAR    (input/output) INTEGER array of DIMENSION at least 1
C            On entry, if DEF = 'N' or 'n' and the desired example
C            depends on integer parameters, then the array IPAR must
C            contain the values for these parameters.
C            For an explanation of the parameters see [1].
C            For Examples 2.3, 2.5, and 2.6, IPAR(1) defines the
C            parameter 's'.
C            For Example 3.1, IPAR(1) defines 'q'.
C            For Examples 3.2 and 3.3, IPAR(1) defines 'n'.
C            For Example 3.4, IPAR(1) defines 'l'.
C            For Example 4.1, IPAR(1) defines 'n'.
C            For Example 4.2, IPAR(1) defines 'l'.
C            On exit, if DEF = 'D' or 'd' and the desired example
C            depends on integer parameters, then the array IPAR is
C            overwritten by the default values given in [1].
```

```
C
C     VEC     (output) LOGICAL array, dimension 8
C             Flag vector which displays the availabilty of the output
C             data:
C             VEC(1), ..., VEC(3) refer to N, M, and P, respectively,
C             and are always .TRUE.
C             VEC(4) is .TRUE. iff E is NOT the identity matrix.
C             VEC(5), ..., VEC(7) refer to A, B, and C, respectively,
C             and are always .TRUE.
C             VEC(8) is .TRUE. iff D is NOT the zero matrix.
C
C     N       (output) INTEGER
C             The actual state dimension, i.e., the order of the
C             matrices E and A.
C
C     M       (output) INTEGER
C             The number of columns in the matrices B and D.
C
C     P       (output) INTEGER
C             The number of rows in the matrices C and D.
C
C     E       (output) DOUBLE PRECISION array, dimension (LDE,N)
C             The leading N-by-N part of this array contains the
C             matrix E.
C             NOTE that this array is overwritten (by the identity
C             matrix), if VEC(4) = .FALSE.
C
C     LDE     INTEGER
C             The leading dimension of array E. LDE >= N
C
C     A       (output) DOUBLE PRECISION array, dimension (LDA,N)
C             The leading N-by-N part of this array contains the
C             matrix A.
C
C     LDA     INTEGER
C             The leading dimension of array A. LDA >= N
C
C     B       (output) DOUBLE PRECISION array, dimension (LDB,M)
C             The leading N-by-M part of this array contains the
C             matrix B.
C
C     LDB     INTEGER
C             The leading dimension of array B. LDB >= N
C
C     C       (output) DOUBLE PRECISION array, dimension (LDC,N)
```

```
C             The leading P-by-N part of this array contains the
C             matrix C.
C
C     LDC     INTEGER
C             The leading dimension of array C. LDC >= P
C
C     D       (output) DOUBLE PRECISION array, dimension (LDD,M)
C             The leading P-by-M part of this array contains the
C             matrix D.
C             NOTE that this array is overwritten (by the zero
C             matrix), if VEC(8) = .FALSE.
C
C     LDD     INTEGER
C             The leading dimension of array D. LDD >= P
C
C     NOTE    (output) CHARACTER*70
C             String containing short information about the chosen
C             example.
C
C     Workspace
C
C     DWORK   DOUBLE PRECISION array, dimension (LDWORK)
C
C     LDWORK  INTEGER
C             The length of the array DWORK.
C             For Example 3.4, LDWORK >= 4*IPAR(1) is required.
C             For the other examples, no workspace is needed, i.e.,
C             LDWORK >= 1.
C
C     Error Indicator
C
C     INFO    INTEGER
C             = 0:  successful exit;
C             < 0:  if INFO = -i, the i-th argument had an illegal
C                   value; in particular, INFO = -3 or -4 indicates
C                   that at least one of the parameters in DPAR or
C                   IPAR, respectively, has an illegal value;
C             = 1:  data file can not be opened or has wrong format.
C
C
C     REFERENCES
C
C     [1]  D. Kressner, V. Mehrmann, and T. Penzl.
C          CTDSX - a Collection of Benchmark Examples for State-Space
C          Realizations of Continuous-Time Dynamical Systems.
```

```
C           SLICOT working note 1998-9. 1998.
C
```

B The MATLAB function CTDSX

The MATLAB function `ctdsx` has the following calling sequence and in-line documentation.

```
function [E,A,B,C,D] = ctdsx(nr,parin)
%CTDSX
%
% Usage:  [E,A,B,C,D] = ctdsx(nr,parin)
%         [E,A,B,C,D] = ctdsx(nr)
%
% Main routine of the benchmark library CTDSX (Version 1.0) described
% in [1]. It generates benchmark examples for time-invariant,
% continuous-time, dynamical systems
%         .
%       E x(t)  =  A x(t) + B u(t)
%                                                                 (1)
%         y(t)  =  C x(t) + D u(t)
%
% E, A are real n-by-n matrices, B is n-by-m, C is p-by-n, and
% D is p-by-m.
%
% Input:
%  - nr    : index of the desired example according to [1];
%            nr is a 1-by-2 matrix;
%            nr(1) defines the group:
%             = 1 : parameter-free problems of fixed size
%             = 2 : parameter-dependent problems of fixed size
%             = 3 : parameter-free problems of scalable size
%             = 4 : parameter-dependent problems of scalable size
%            nr(2) defines the number of the benchmark example within
%            a certain group.
%  - parin : parameters of the chosen example;
%            referring to [1], the entries in parin have the following
%            meaning:
%            Ex. 2.1 : parin(1)   = epsilon
%            Ex. 2.2 : parin(1)   = epsilon
%            Ex. 2.3 : parin(1)   = s
%            Ex. 2.4 : parin(1:7) = [b mu r r_c k_l sigma a]
%            Ex. 2.5 : parin(1)   = s
%            Ex. 2.6 : parin(1)   = s
%            Ex. 2.7 : parin(1:2) = [mu nu]
%            Ex. 3.1 : parin(1)   = q
%            Ex. 3.2 : parin(1)   = n
%            Ex. 3.3 : parin(1)   = n
%            Ex. 3.4 : parin(1)   = l
```

```
%            Ex. 4.1 : parin(1)   = n
%                      parin(2:4) = [a, b, c]
%                      parin(5:6) = [beta_1,beta_2]
%                      parin(7:8) = [gamma_1,gamma_2]
%            Ex. 4.2 : parin(1)   = l
%                      parin(2:4) = [mu delta kappa].
%            parin is optional; default values as defined in [1] are
%            used as example parameters if parin is omitted. Note that
%            parin is not referenced if nr(1) = 1.
%
% Output:
%  - E, A, B, C, D :  matrices of the dynamical system (1).
%
% References:
%
% [1] D. Kressner, V. Mehrmann, and T. Penzl.
%     CTDSX - a Collection of Benchmark Examples for State-Space
%     Realizations of Continuous-Time Dynamical Systems.
%     SLICOT working note 1998-9. 1998.
```