# ELEC 3035: Lab 2 — state space and polynomial representations

**Lecturer: Ivan Markovsky**

The exercises below use the Control System Toolbox of Matlab. (Matlab's toolboxes are libraries of functions grouped together for solving a specific type of problems.) As the name suggests, the Control System Toolbox is designed for linear time-invariant system analysis and synthesis. The list of functions and short descriptions are printed by typing `help control` in Matlab.

1. *Conversion between transfer function and state space representations* Consider the continuous-time second order system, defined by the transfer function

$$H(s) = \frac{s-1}{s^2+2s+10}.$$

   (a) Using the function `tf`, define a transfer function representation `sys_tf` of the system.

   (b) Using the function `ss`, convert the transfer function representation to a state space representation `sys_ss` and verify that the computed state space representation indeed corresponds to the transfer function $H$. (Hint: use the formula for state space to transfer function conversion on page 19 of Lecture 2.)

   (c) Choose an arbitrary $2 \times 2$ nonsingular matrix $T$ and apply the change of basis transformation (see, page 17 of Lecture 2). Make sure that the newly obtained representation also corresponds to the transfer function $H$.

*Solution:*

(a) `sys_tf = tf([1 -1],[1 2 10])`

(b) `sys_ss = ss(sys_tf)`

```
a =
          x1    x2
   x1     -2  -2.5
   x2      4     0

b =
        u1
   x1    1
   x2    0

c =
           x1      x2
   y1       1   -0.25

d =
        u1
   y1    0
```

Substituting the numerical values of $A, B, C, D$ in the formula $C(sI-A)^{-1}B+D$, we have

$$\begin{bmatrix} 1 & -0.25 \end{bmatrix} \begin{bmatrix} s+2 & 2.5 \\ -4 & s \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{s(s+2)+10} \begin{bmatrix} 1 & -0.25 \end{bmatrix} \begin{bmatrix} s & -2.5 \\ 4 & s+2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{s-1}{s^2+2s+10}.$$

(c)
```
t = rand(2); % random 2x2 matrix
inv_t = inv(t);
sys_ss2 = ss(inv_t * sys_ss.a * t, inv_t * sys_ss.b, sys_ss.c * t, sys_ss.d);
tf(sys_ss2) % check the transfer function
```

```
Transfer function:
    s - 1
--------------
s^2 + 2 s + 10
```

□

2. *Poles and zeros of the system* For the system of exercise 1 compute

   (a) the poles, i) using the `pole` function, and ii) analytically,

   (b) the zeros, i) using the `zero` function, and ii) analytically.

   *Solution:*

   (a) Numerically:

   ```
   pole(sys_tf)

   ans =

     -1.0000 + 3.0000i
     -1.0000 - 3.0000i
   ```

   Analytically: The poles are the roots of the denominator of the transfer function, so we have

   $$s_{1,2} = -1 \pm \sqrt{1-10} = -1 \pm i3.$$

   (b) Numerically:

   ```
   zero(sys_tf)

   ans =

        1
   ```

   Analytically: the zeros are the roots of the numerator of the transfer function, so we have $z = 1$.

□

3. *Discretization* Convert the continuous-time model of exercise 1 to a discrete-time model, using the function `c2d`. Use the following sampling times $t_s = 0.001$, 0.01, 0.1, and 1. Using the function `impulse`, compute and plot the impulse response of the continuous-time model and the discretized models. Comment on the results.

   *Solution:*

   ```
   impulse(sys_tf)
   sys_d1 = c2d(sys_tf,0.001); figure, impulse(sys_d1,6)
   sys_d2 = c2d(sys_tf,0.01); figure, impulse(sys_d2)
   sys_d3 = c2d(sys_tf,0.1); figure, impulse(sys_d3)
   sys_d4 = c2d(sys_tf,1); figure, impulse(sys_d4)
   ```

   Appropriate sampling time is $t_s \leq 0.01$.

□

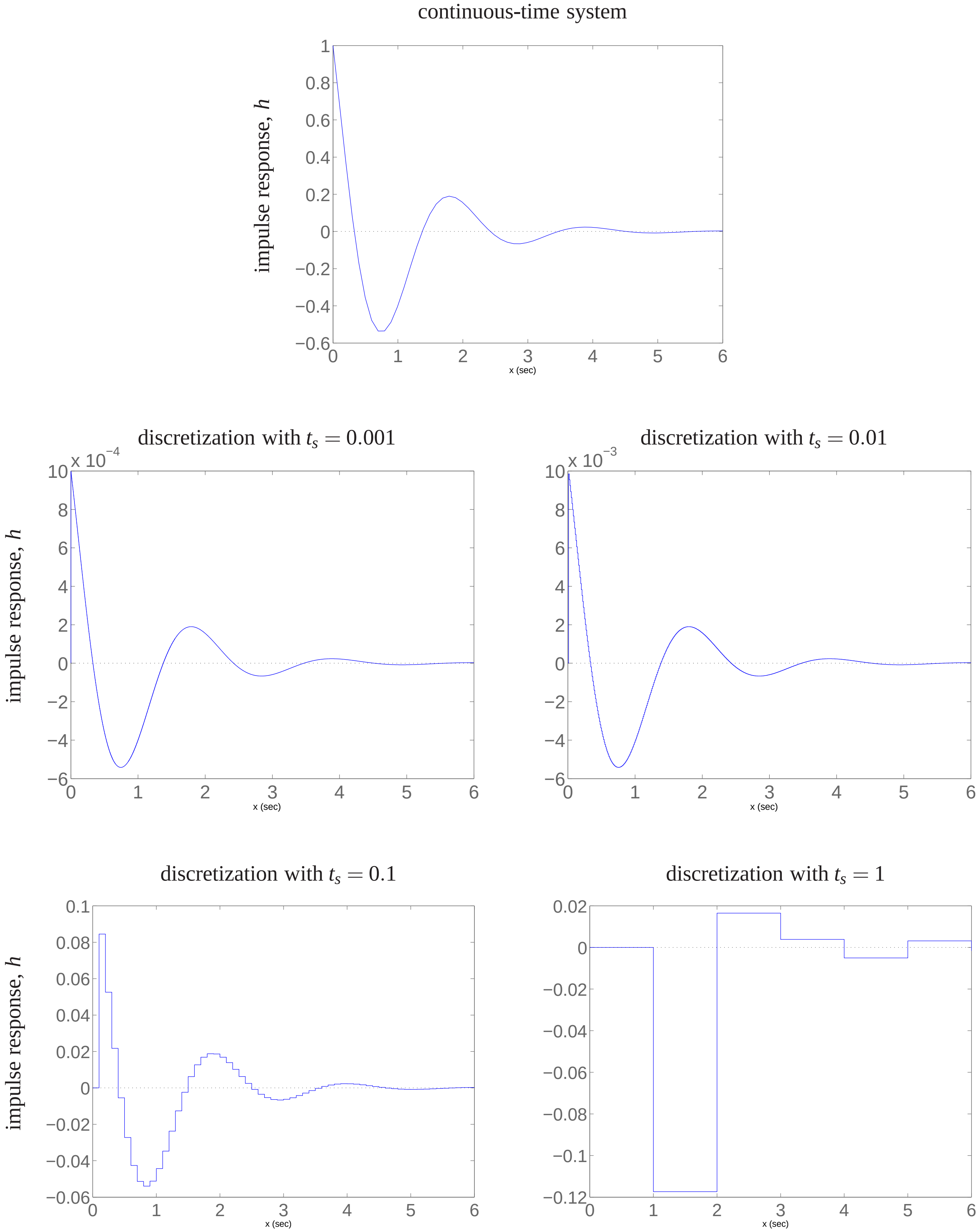


Figure 1: Impulse responses of the continuous-time system and its discretizations with sampling times $t_s = 0.001$, 0.01, 0.1, and 1.