Assignment 4

1. Continuous-time to Discrete-time transformation

- Define the following transfer function in MATLAB:
  - MATLAB command: `tf`
  - **Hint:** Requires the MATLAB Control System Toolbox

$$G(s) = \frac{1}{s^2 + 2s + 4}$$

- Plot the impulse response, step response, bode diagram and pol-zero map of the given system using the following MATLAB commands:
  - `impulse`
  - `step`
  - `bode`
  - `pzmap`
- Determine a corresponding discrete-time transfer function with a sample time T=0.2s with an equal impulse, respectively step response. Use the impulse invariant and zero order hold transformation methods by using the following MATLAB command:
  - `c2d(…,…,’impulse’)`
  - `c2d(…,…,’zoh’)`
- Which effect would have the multiplication of z with the resulting transfer function?
- Compare the transformed discrete system with sample time T=0.2s and T=0.15s.
- Determine a discrete system with the given parameters using the bilinear transformation method. Use the following MATLAB command:
  - `c2d(…,…,’tustin’)`

2. Symbolic computation: Bilinear transform

- Transform the following transfer function by hand:

$$G(s) = \frac{1}{s^2 + 2s + 4}$$

- Check the result with the MATLAB using the following commands:
  - `syms`
  - `simplify`
  - `pretty`
  - **Hint:** Requires the MATLAB Symbolic Toolbox

Home Assignment: Cont. Butterworth for discrete applications

- MATLAB uses the function `butter` (signal processing toolbox) to create discrete butterworth coefficients (b and a).

  Internally it uses the bilinear transform method to create the discrete coefficients from the analog equivalent. In the following we want to verify with our own implementation that the bilinear transformation can be used in this scenario.

Hint: The function `butter` does some processing to make the coefficients more stable, but those optimizations can be neglected in the following. You will get minor differences in the pole/zero map.

- Furthermore, the function `butter` can also be used to create analog butterworth coefficients. See MATLAB manual (F1) for further help.
- Create the following
  - discrete butterworth coefficients with Fs=100Hz and
  - analog butterworth coefficients using
  - common parameters
    - Order: 5
    - Cut-off frequency: 0.1 normalized Nyquist frequency
- Transform the analog butterworth transfer function by using the bilinear transformation.
- Overlay the pole zero maps of both the discrete (MATLAB reference) and the c2d converted (analog matched version) transfer functions.

```
hold on;
pzmap(Hz_mat,'r-'); % MATLAB reference
pzmap(Hz_c2d,'b-'); % c2d matched version
hold off;
```

- Show that the overlayed zeros and poles match very closely.
  - Hint: Really read the help for `butter` function thoroughly. Especially for the creation of analog coefficients.
- Required upload
  - `demobutter.m`