MATLAB commands to analyse linear time-invariant (LTI) systems

Creation of a "sys" object

```
sys = ss(A,B,C,D)      % creates object "sys" of the continuous-time state-space model:
                       % dx/dt = Ax(t) + Bu(t)
                       % y(t) = Cx(t) + Du(t)
```

Eigenvalues, eigenfrequencies,damping

```
lambda = eig(A);                          % calculate the eigenvalues of matrix A
freq = imag(lambda)/(2*pi);               % calculate damped eigenfrequencies
dr = -real(lambda)/(norm(lambda))*100;    % critical damping ratio in percent
```

Transfer function

```
freq=[0.1:0.1:5];                         % frequency vector in Hz (0.1 to 5 Hz)
w= 2*pi*freq;                             % create vector w in rad/s
[mag,phase] = bode(sys,w);                % transfer functions of "sys" at frequencies "w"
```

"mag" gives the magnitude, "phase" the phase angle in degrees. Note that "mag" and "phase" have three dimensions, "mag(2,1,:)" represents the magnitude of the second output to the first input of the object "sys" for all frequencies specified in the vector "w". The MATLAB "plot" command does not accept a signal with three dimensions, use the "squeeze" command to remove the singleton dimensions, example:

```
plot( freq, squeeze(mag(2,1,:) ); % plot magnitude of the transfer function, input 1 - output 2
```

To plot the transfer function on a log-log scale, use "loglog" instead of "plot". "semilogx" give a logarithmic x-axis.

As an alternative to "bode" it is also possible to use "freqresp".

```
H = freqresp(sys,w);          % complex values of the transfer function H.
```

Just as with the "bode" command H has three dimensions: output, input, frequency.

```
mag21 = norm(H(2,1,:);        % magnitude of the transfer function, input 1 - output 2
phase21 = angle(H(2,1,:);     % phase angle (radians!) of the transfer function, input 1 -
output 2
```

Step response

```
time=[0:0.01:5];              % time vector (0 to 5 sec.)
y = step(sys,time);           % calculate step response
```

The vector "y" has three dimensions: time, output, input

```
plot(time, squeeze(y(:,2,1) );  % plot time history of output 2 to a step on the input  1
```

Response to an arbitrary input signal

```
time=[0:0.01:5];              % time vector (0 to 5 sec.)
u = 0.1*sin(2*pi*time);       % input signal  of 1Hz with amplitude of 0.1
y = lsim(sys,u,time);         % perform the linear simulation
```

In case of multiple inputs, "u" should have as many columns as there are input to the system!