Transfer Function

OpenRails

Resources

- https://en.wikibooks.org/wiki/Signals_and_Systems/Second_Order_Transfer_Function
- https://en.wikipedia.org/wiki/Bilinear_transform

- Real-world examples:
  - https://nl.mathworks.com/help/control/ug/analyzing-the-response-of-an-rlc-circuit.html
  - https://nl.mathworks.com/help/control/ug/dc-motor-control.html
  - …

Code example

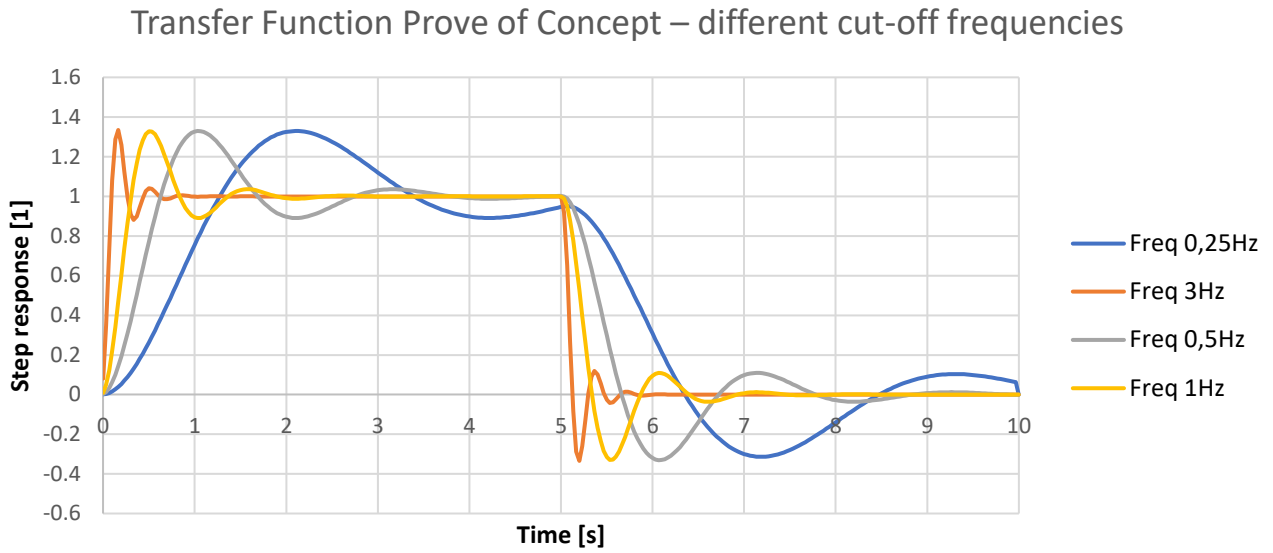
- Based on bilinear transformation of a 2nd-order system

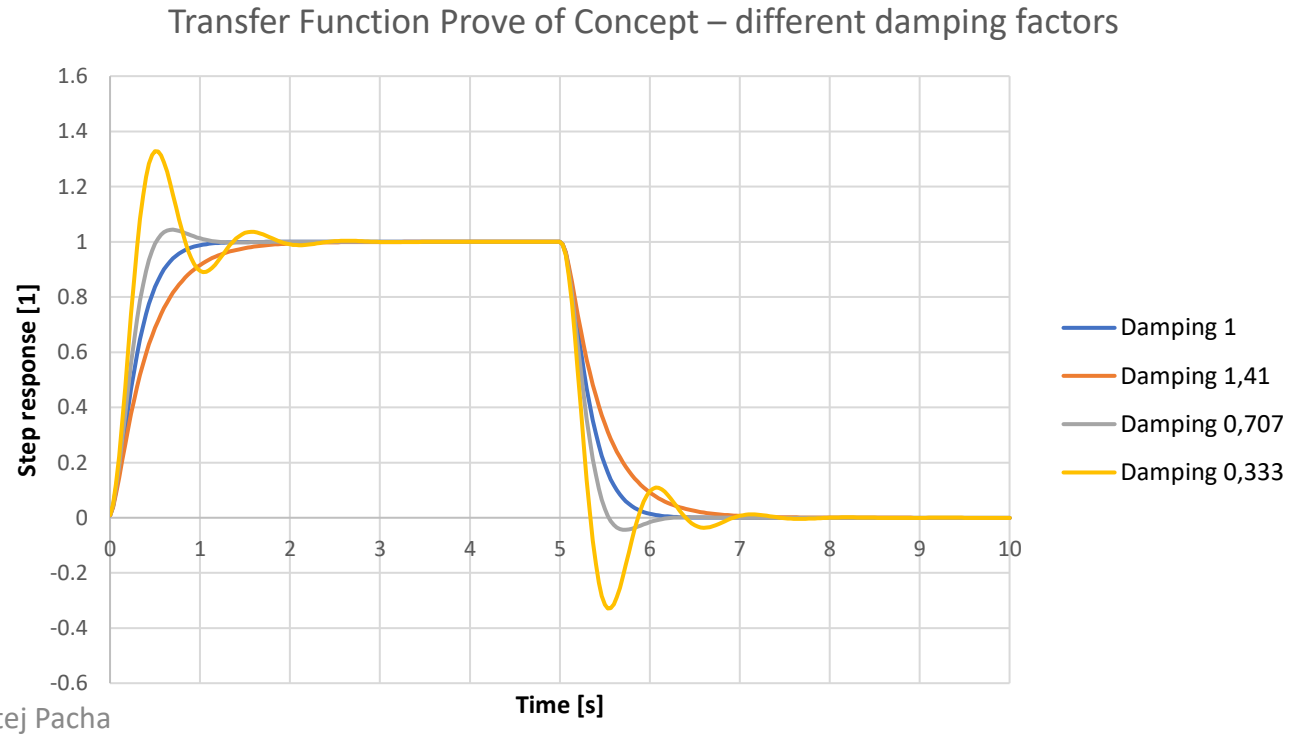
$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 \cdot s + \omega_0^2}$$

- ω_0 … cut-off angular frequency
- ζ …… damping factor
- Written in Console C#

User performance

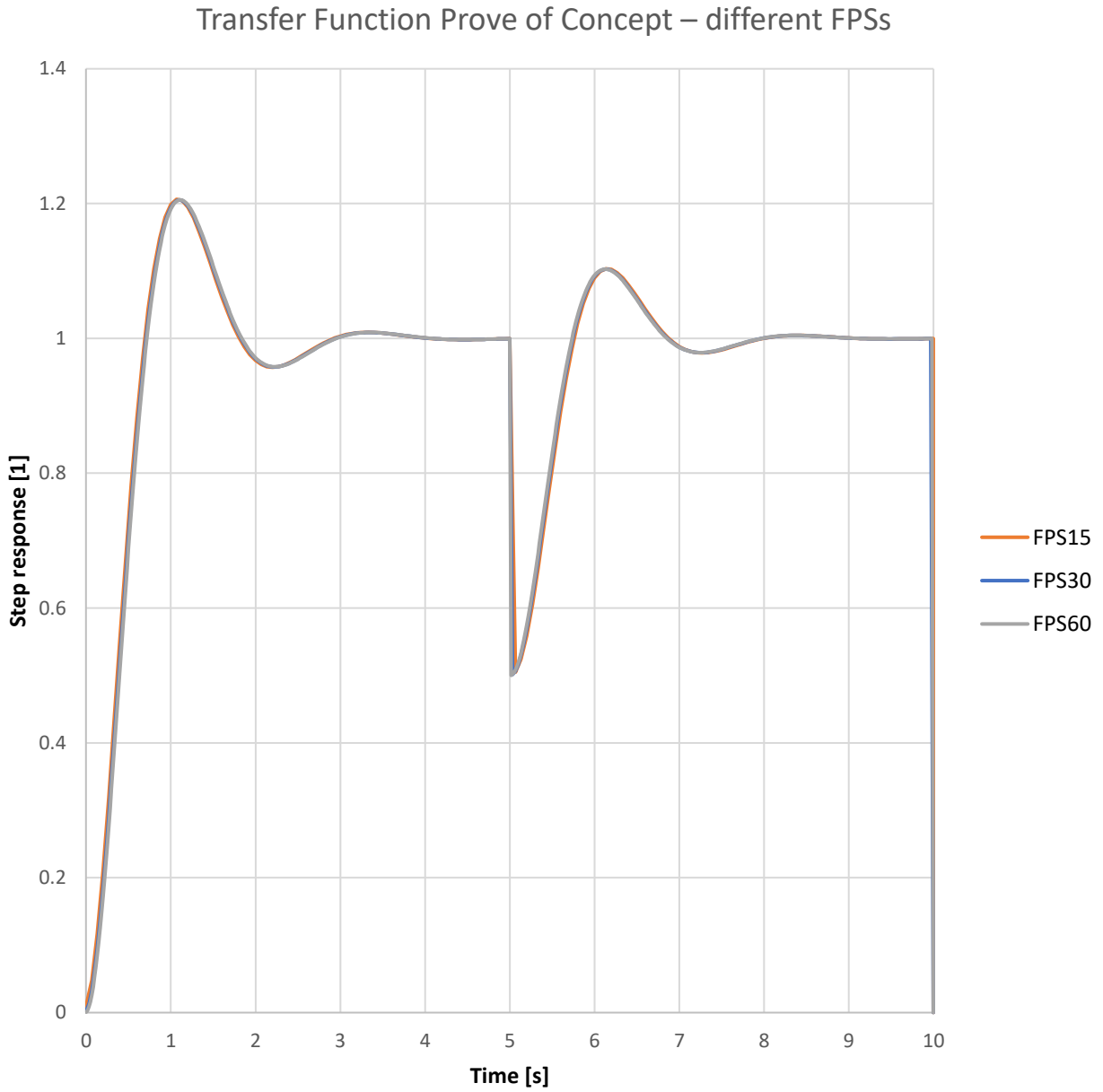
- Well defined step-response
  - Cut-off frequency defines the „speed“ of response
  - Damping factor defines how much the oscillations are damped





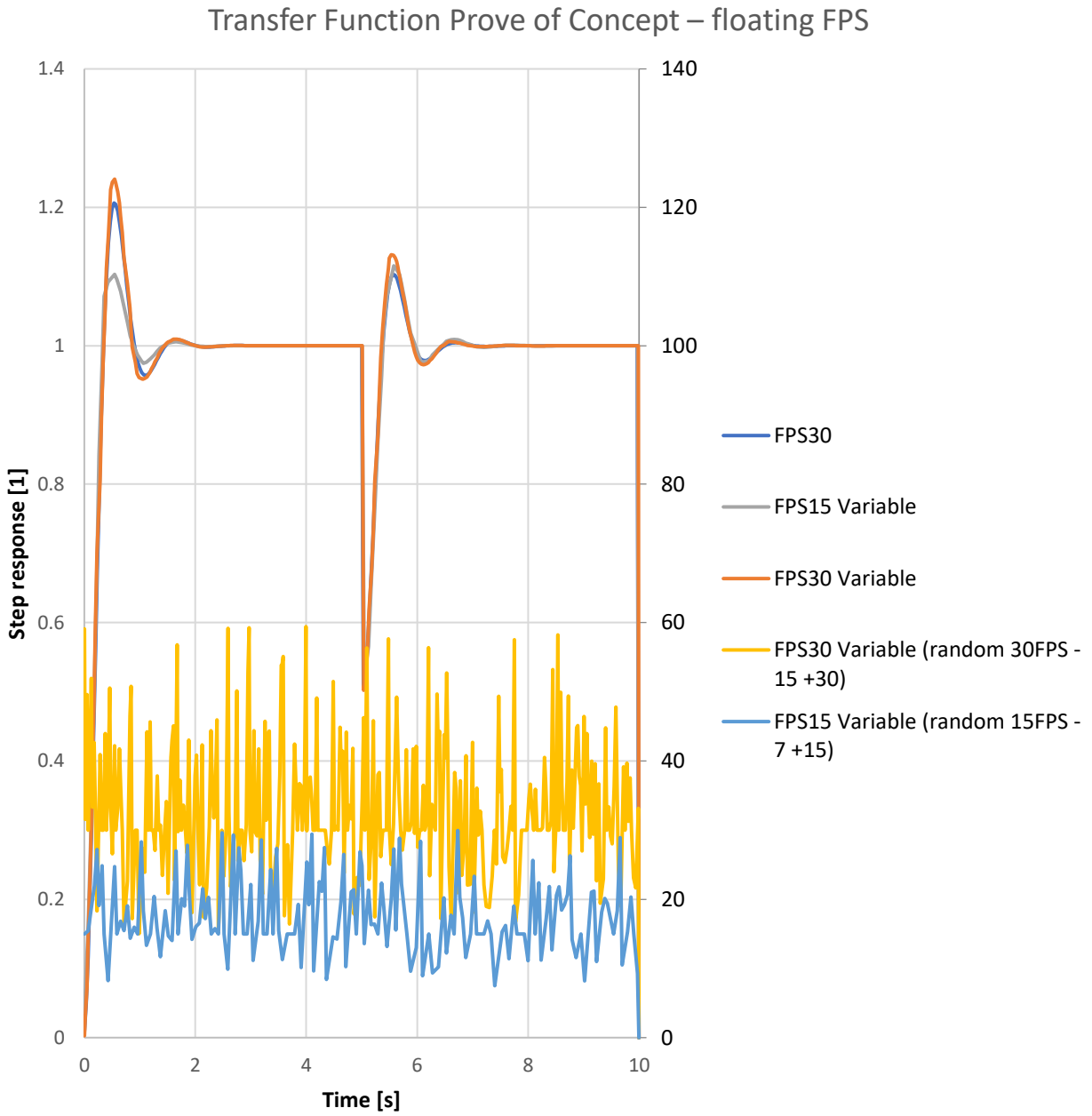
User performance

- Consistent performance at different constant FPS



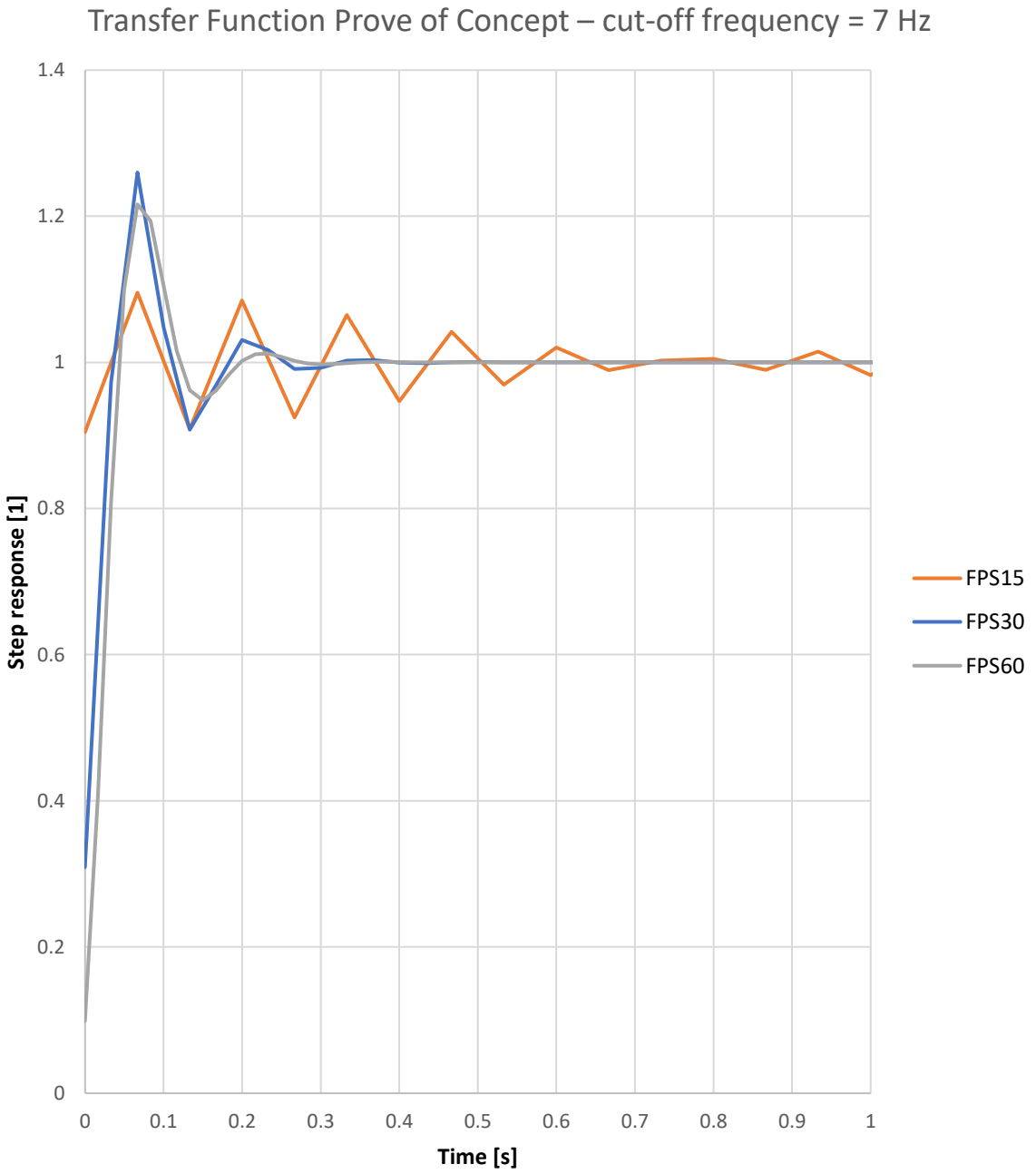
User performance

- Good stability at variable FPS (randomly generated from 0.5x to 2x of base FPS)
- Depends on the settings of the transfer function vs. Base FPS



User performance

- Fast systems at low FPS tend to be unstable
- Common criteria applies
  - FPS > 3x $f_{cut\text{-}off}$
  - Sets the limitation of use to 5Hz and less cut-off frequency systems



Benefits

- Well-defined model of 2nd order systems
- Works with constant FPS and with variable FPS with limited performance
- Parameters (damping factor, cut-off frequency) can be changed dynamically
- Multiple systems can be linked together to create higher-order systems