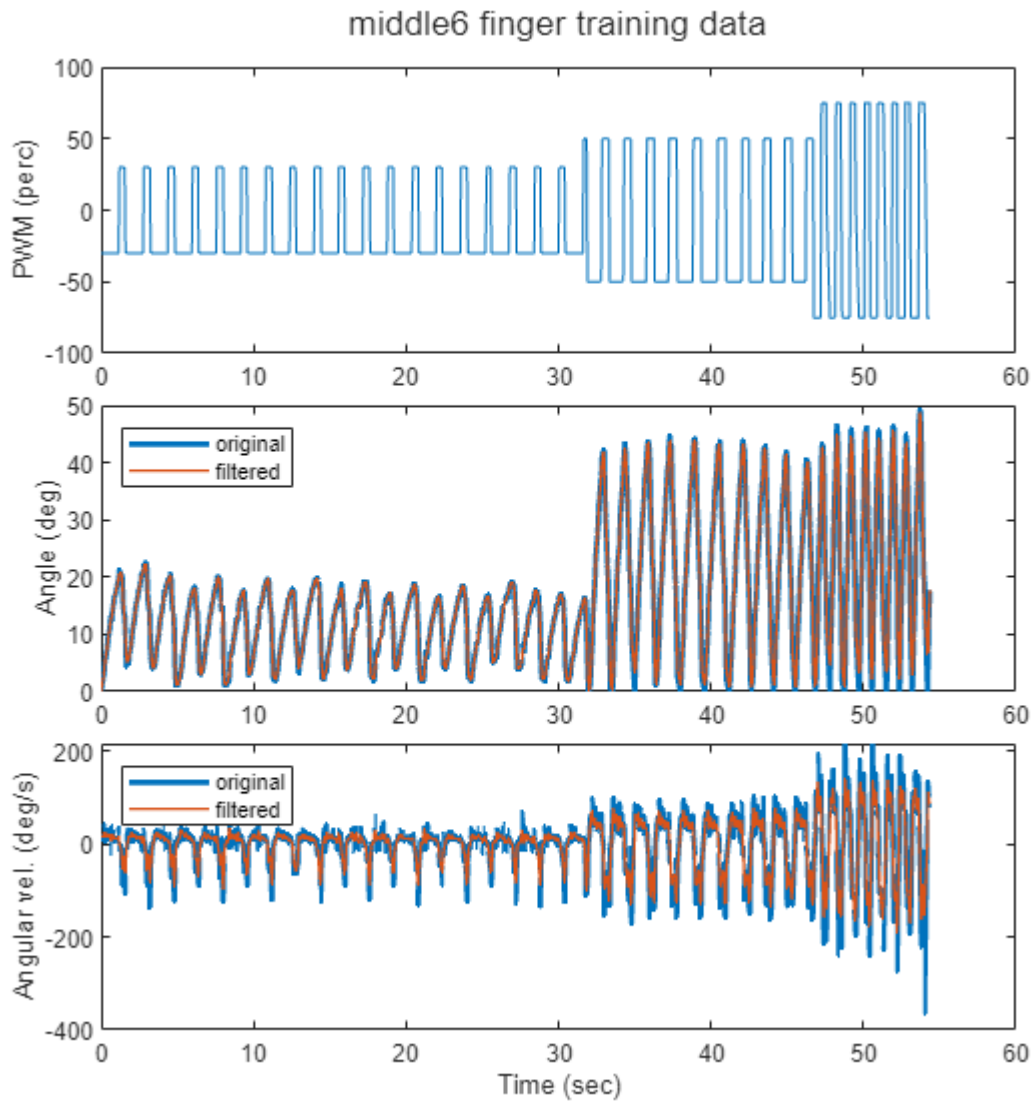


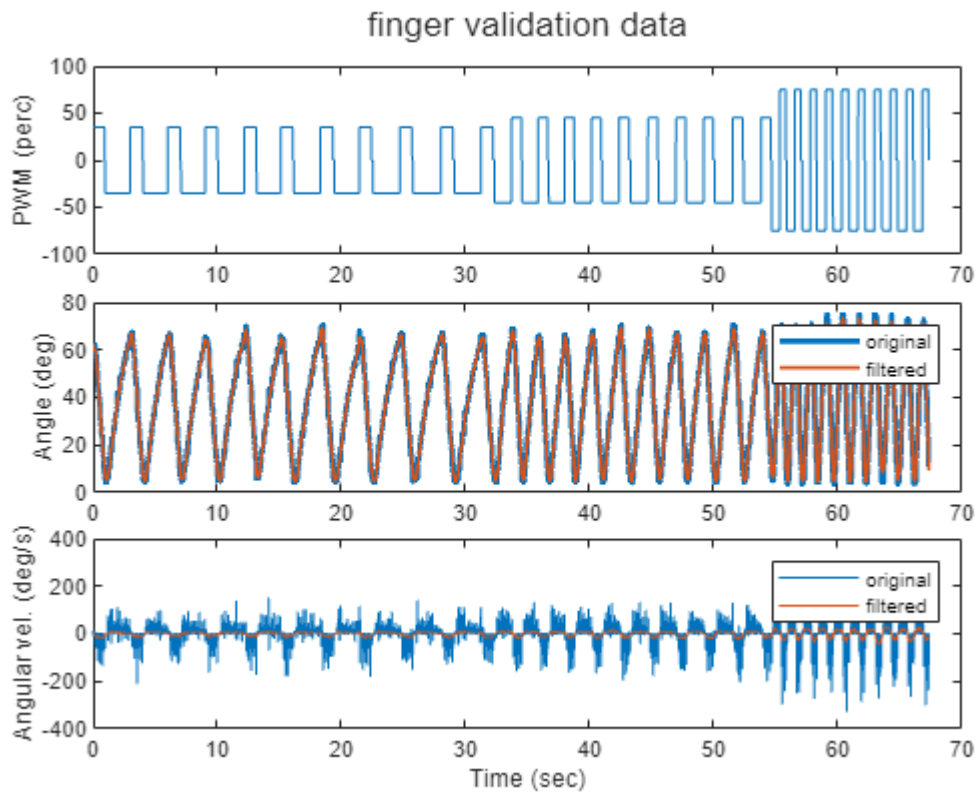
ergoCub Finger PID tuning

Copyright (C) 2023 Fondazione Istituto Italiano di Tecnologia (IIT)

All Rights Reserved.

Read and plot datasets





Run identification as process model

We target a continuous LTI model in the form

$$\frac{y}{u} = \frac{k(s - z_1)}{(s - p_1)(s - p_2)}$$

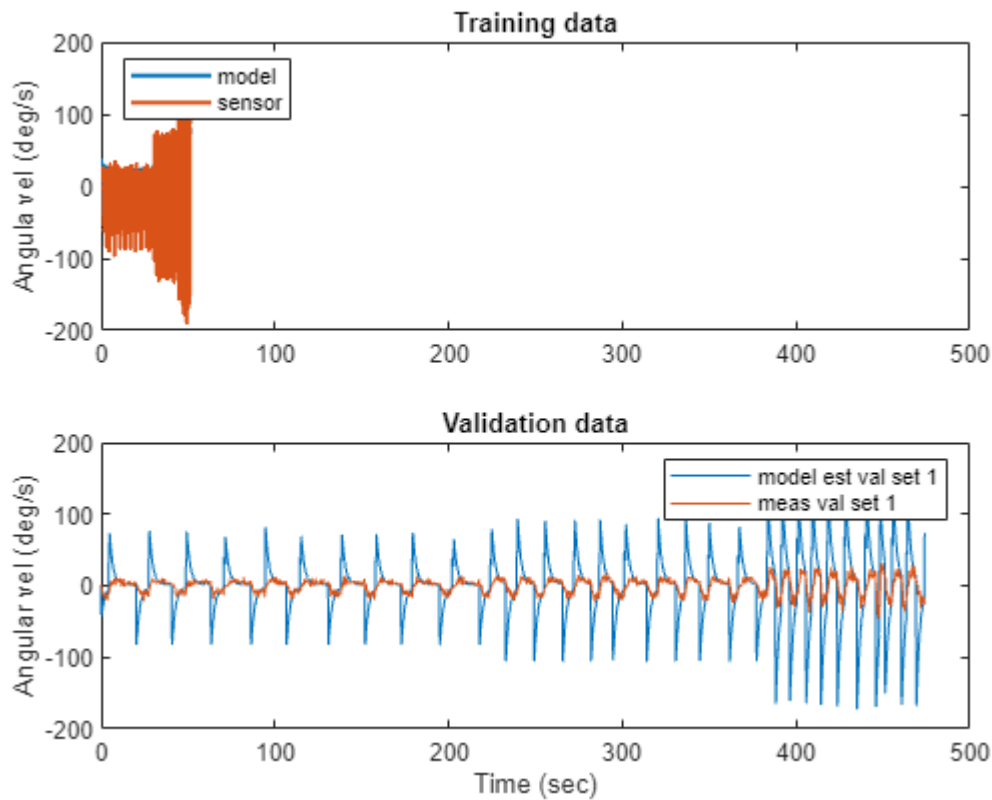
The poles can be real or complex conjugate.

ans =

```
From input "u1" to output "y1":
-22.728 (s+0.01721)
-----
(s+16.95) (s+0.5272)
```

```
Continuous-time zero/pole/gain model.
Model Properties
```

Plot results.



```
name: "Validation set 1 result"
value: 40.9043
```

Add integrator for position control

We change the IO relationship from $\frac{\dot{\theta}}{u}$ to $\frac{\theta}{u}$ since we want to control the finger position.

Run autotuning

Here we define the specifications to tune the position controller in a robust way, and run the tuning algorithm.

Define tuning goals

Define the desired specifications.

```
option 3 - custom
Ts:0.05, responsetime:0.5, derror:0.005, peakerror:1.1, Overshoot:20
```

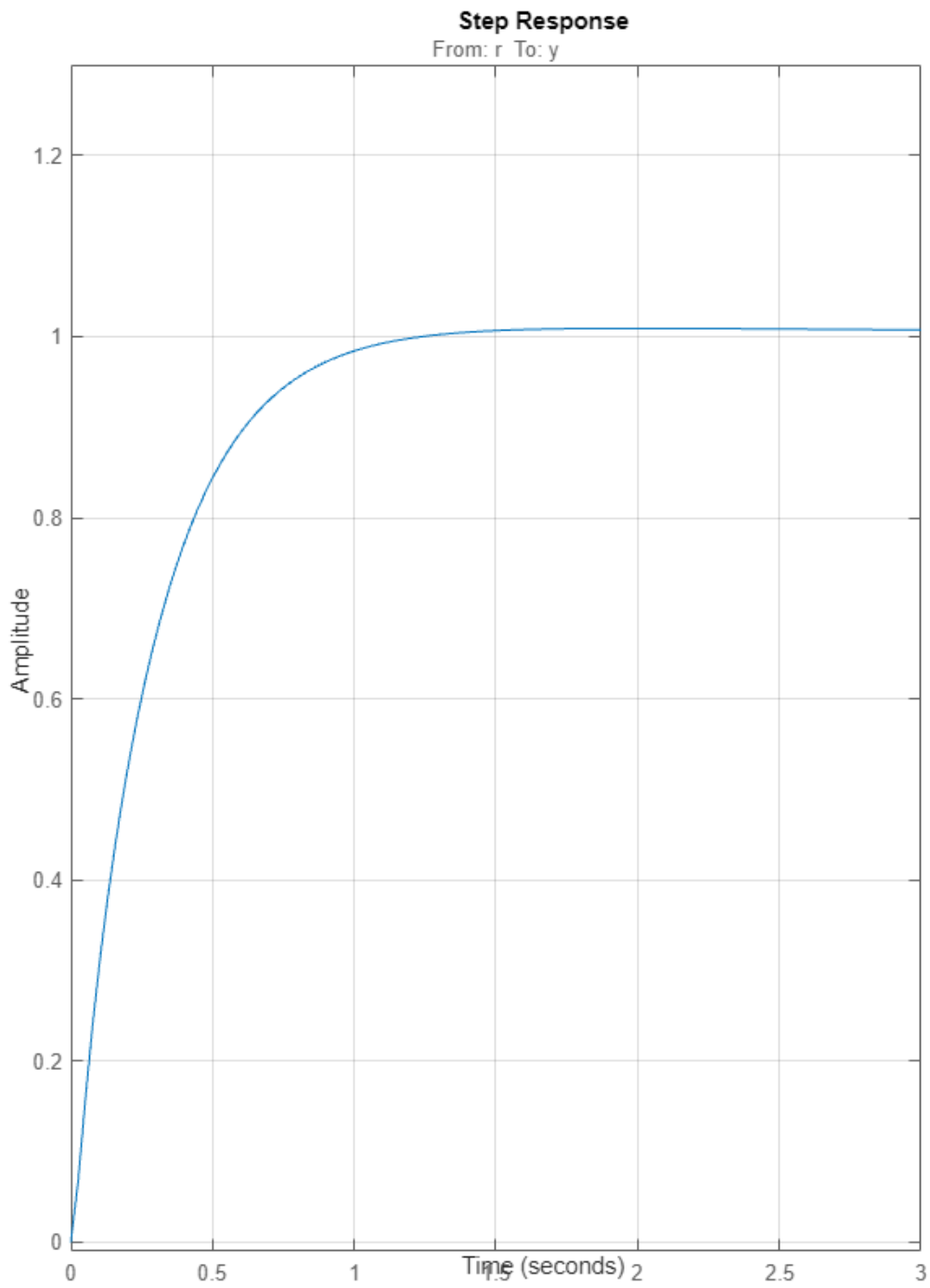
Define the goals.

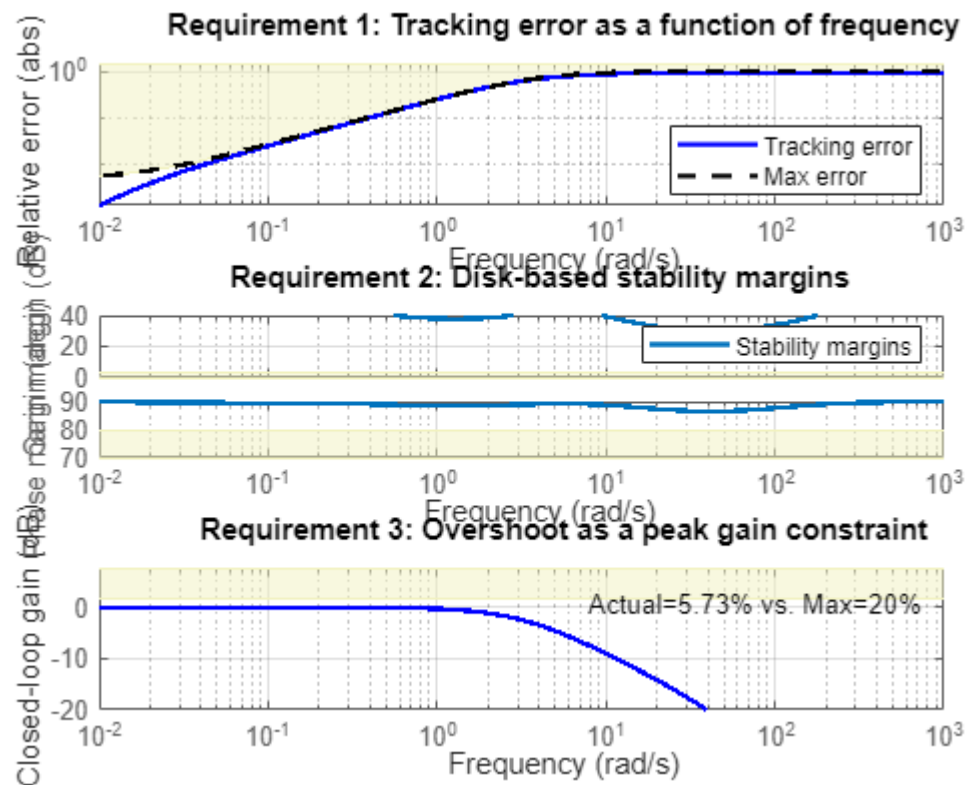
```
hard_goals: tracking
soft_goals: margin, overshoot
```

Tune up the controller

Tune a PI controller. The D is zero since the system is highly damped due to friction.

Plot tuning results.





Get the discretized controller.

Cz =

$$K_p + K_i * \frac{T_s * (z+1)}{2 * (z-1)} + K_d * \frac{1}{T_f + T_s / 2 * (z+1) / (z-1)}$$

with $K_p = -2.78$, $K_i = -1.53$, $K_d = -0.158$, $T_f = 0.01$, $T_s = 0.05$

Sample time: 0.05 seconds

Discrete-time PIDF controller in parallel form.

Model Properties