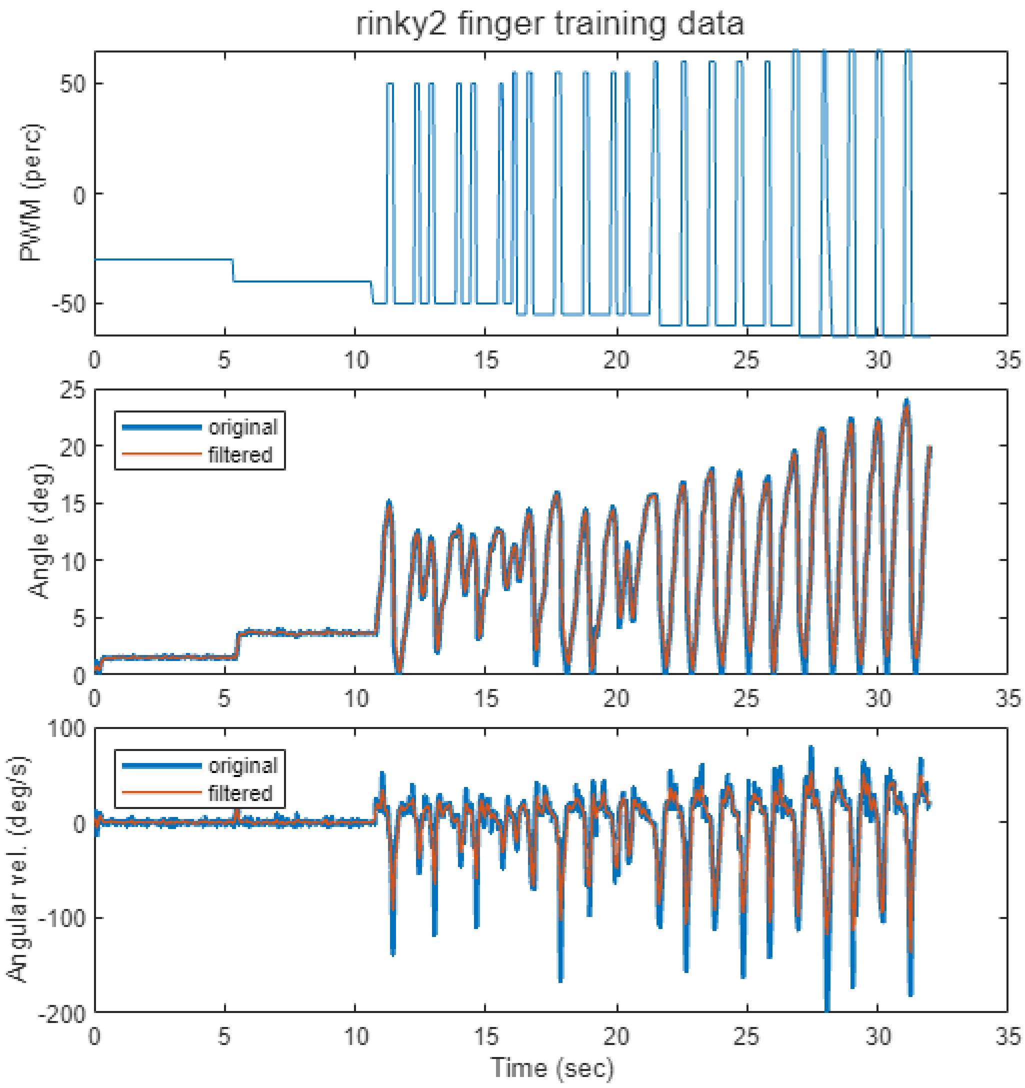
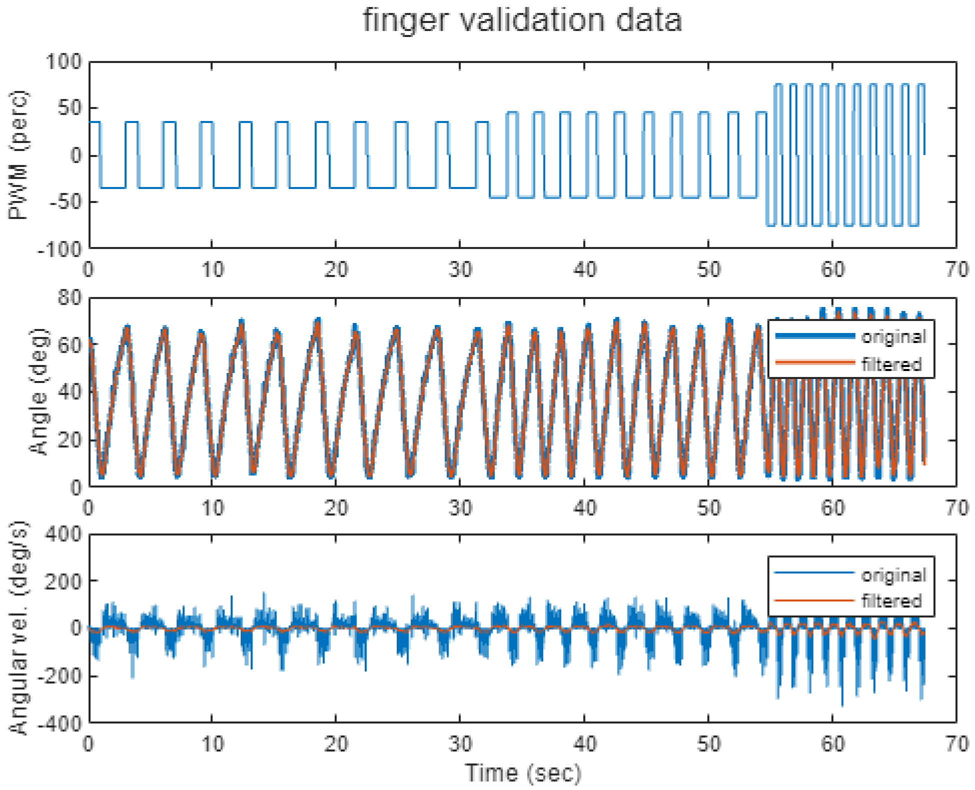
# ergoCub Finger PID tuning

## Read and plot datasets





## Run identification as process model

We target a continuous LTI model in the form

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

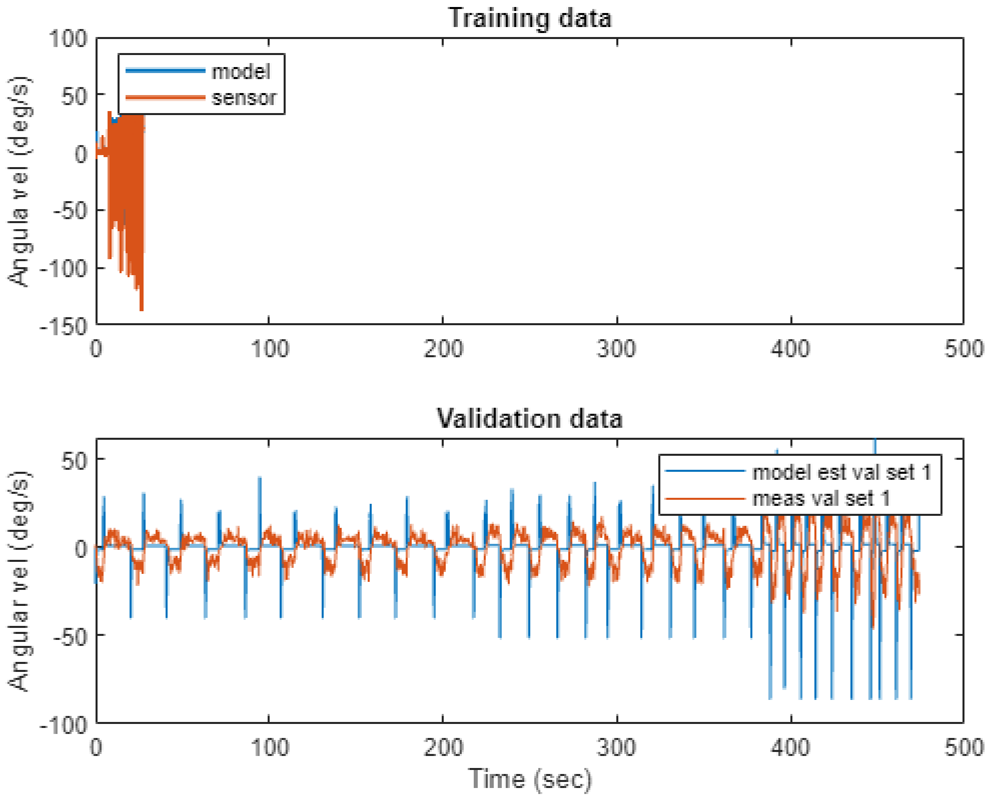
The poles can be real or complex conjugate.

```
ans =

  From input "u1" to output "y1":
   -8.5371 (s+0.09388)
  ----------------------
  (s^2 + 10.59s + 31.68)

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

Plot results.



```
 name: "Validation set 1 result"
value: 15.3583
```

## 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, dcerror:0.005, peakerror:1.1, Overshoot:20
```

Define the goals.

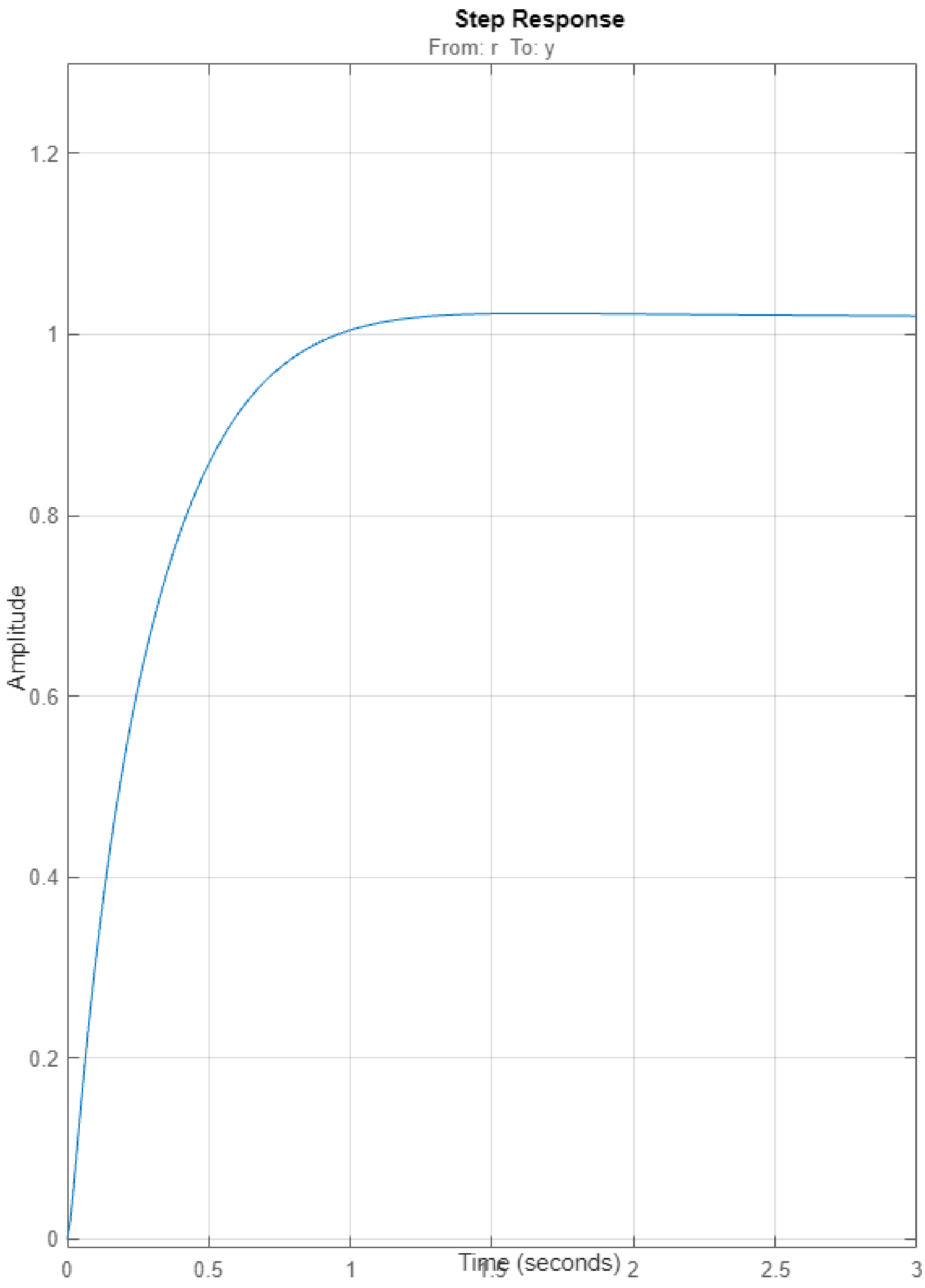
```
hard_goals: tracking
```

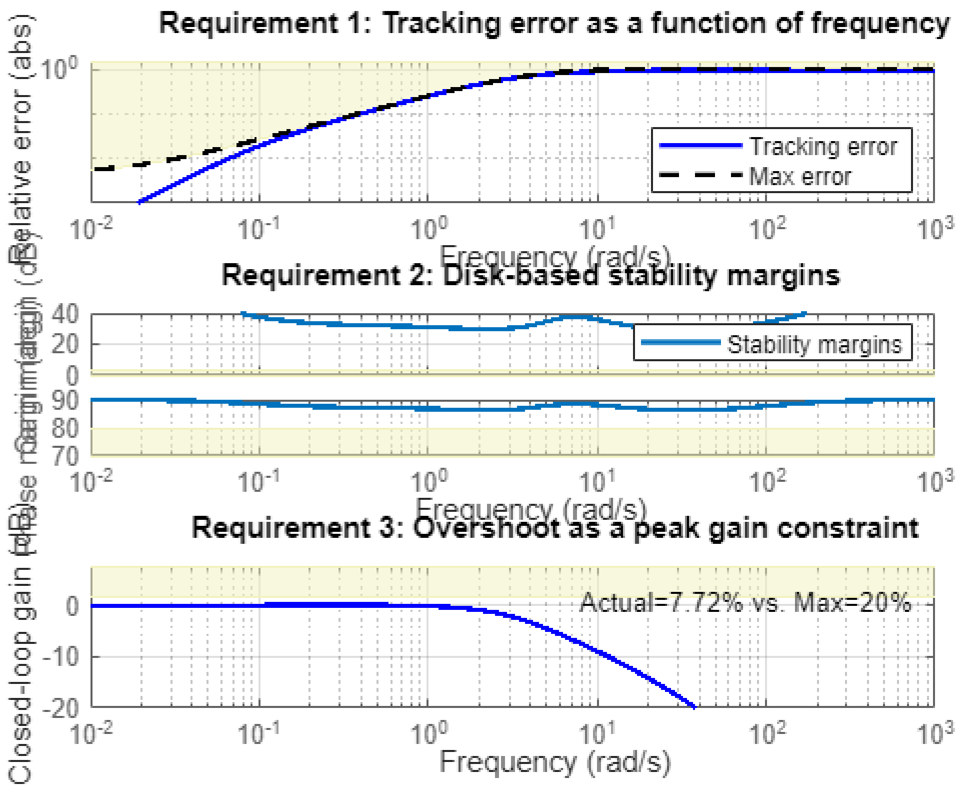
```
soft_goals: margin, overshoot
```

### Tune up the controller

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

Plot tuning results.




Requirement 1: Tracking error as a function of frequency

Requirement 2: Disk-based stability margins

Requirement 3: Overshoot as a peak gain constraint

Get the discretized controller.

```
Cz =
 
             Ts*(z+1)                  1
  Kp + Ki * -------- + Kd * -------------------
             2*(z-1)         Tf+Ts/2*(z+1)/(z-1)

  with Kp = -4.38, Ki = -14, Kd = -0.428, Tf = 0.01, Ts = 0.05
 
Sample time: 0.05 seconds
Discrete-time PIDF controller in parallel form.
Model Properties
```