Feedback Control: Selected Topics

Heuristic Tuning of PID loops

- Assuming a reasonably simple and stable plant, rule of thumb is:
  – Turn on the proportional gain and the derivative gain together until the system transient response is acceptable
  – Turn on the integral gain slowly so as to eliminate the steady-state error
- Why does it work?
  – Proportional gain is like a spring, the derivative gain is like damping. They are like *physical dissipative devices* and unlikely to destabilize your system (until you take the spring and damping too high)
  – Integral gain IS destabilizing → proceed cautiously!

1. Zeigler-Nichols Methods for Tuning of PID Controllers

- Ultimate cycle method
  - Increase proportional gain only until the system has sustained oscillations at a period T_u; this gain is K_u. (If no oscillations occur, don't use this method!)
  - For proportional-only control, use
    - $K_p = K_u / 2$
  - For proportional-integral control use
    - $K_p = 0.45 K_u$ and $K_i = 0.54K_u / T_u$
  - For full PID, use
    - $K_p = 0.6K_u$, $K_i = 1.2K_u / T_u$ and $K_d = 4.8K_u / T_u$

Assume the plant is of the form $P = k / (s^2 + 2\zeta\omega_n s + \omega_n^2)$

(no zeros, undamped natural frequency ω_n, damping ratio ζ)

With proportional-only control at K_u, the CL characteristic equation is

$$s^2 + 2\zeta\omega_n s + \omega_n^2 + kK_u = 0$$

Because system has oscillations at frequency $2\pi/T_u$, we know that

$$\omega_n^2 + kK_u \sim [\, 2\pi/T_u \,]^2 \qquad \text{OR} \qquad kK_u = [\, 2\pi/T_u \,]^2 - \omega_n^2 = \mathbf{Q}$$

At this condition, the damping is not enough to counter the unmodelled dynamics that are causing the oscillation, so it is *ignored*.

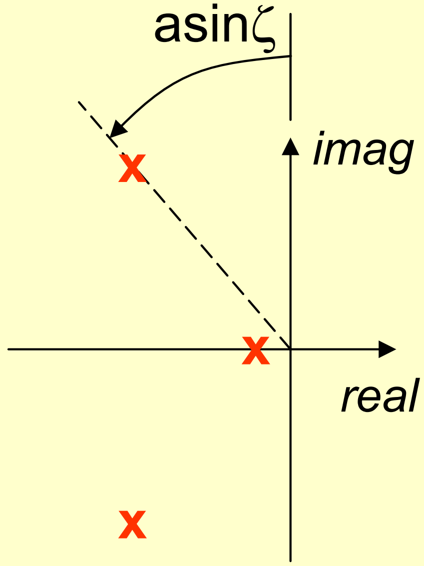
The characteristic equation with the Z-N PID gains becomes:

$$s^2 + 0 + \omega_n^2 + k * [\text{ PID controller }] = 0$$

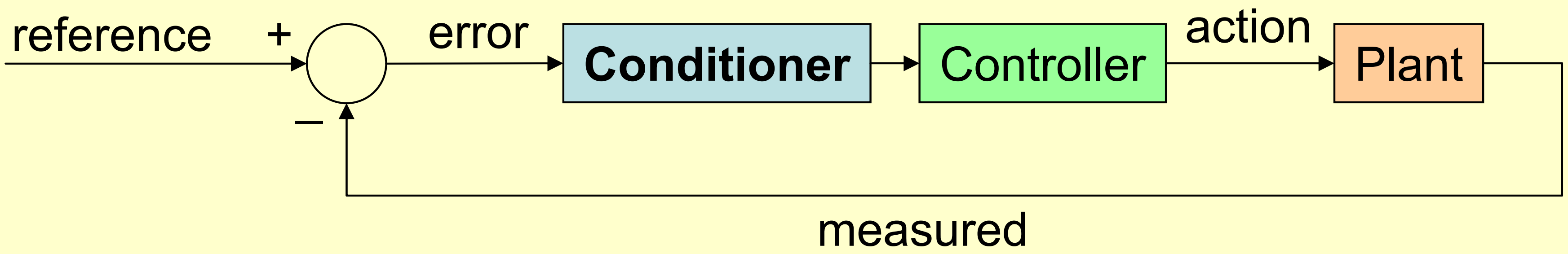
$$s^2 + 0 + \omega_n^2 + \mathbf{Q}\,[\, 0.6 + 1.2 / T_u / s + 4.8\, s / T_u \,] = 0$$

$$s^3 + [\, 4.8\,\mathbf{Q} / T_u \,]\, s^2 + [\, 4\,\pi^2 / T_u^2 - \mathbf{Q} + 0.6\,\mathbf{Q} \,]\, s + 1.2\,\mathbf{Q} / T_u = 0$$

For a wide range of **Q** and T_u, this will give ~20% overshoot (ζ~0.7) because the poles look like this:



2. The 2π Discontinuity in Heading Control



Objective of Conditioner is to make sure:

Controller never gets an error signal that is discontinuous because of this effect

Controller will always go for the shortest path – i.e., will turn 90 degrees left instead of 270 degrees right!

Simple logic:

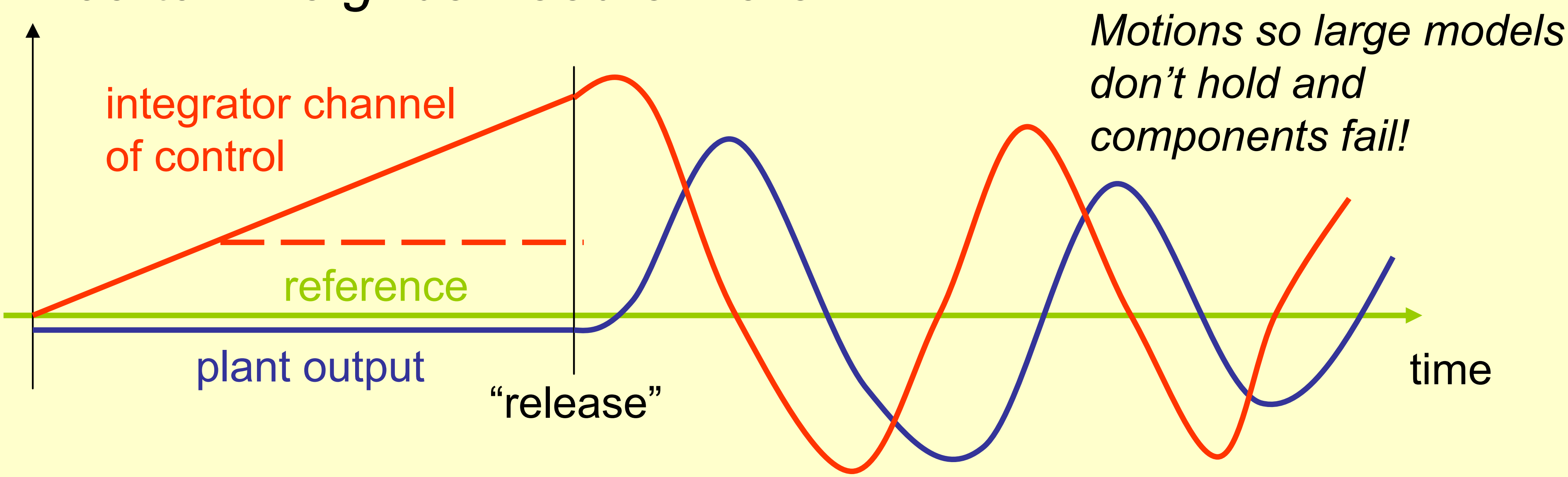
Subtract or add 2π to error to bring it into the range $[-\pi, \pi]$.

3. Integrator Windup

- A purely linear effect that has broken many systems and caused damage and injury!
- Basic issue: The integrator in the controller builds up a large control signal over time if the system is prevented from responding.

PID: K_p*error + K_d*d(error)/dt + $K_i \int$ error dt

Solution: constrain this part of the control to be within a certain neighborhood of zero.



4. Sensor Noise & Outliers

- Most common model for sensor noise is Broadband, Gaussian:
  - Broadband means no particular frequency is favored – spectrum is flat; white noise.
  - Gaussian means samples fit the probability distribution function:

  $N(0,1) = 1 / \mathrm{sqrt}(2\pi) * \exp [- x^2 / 2]$

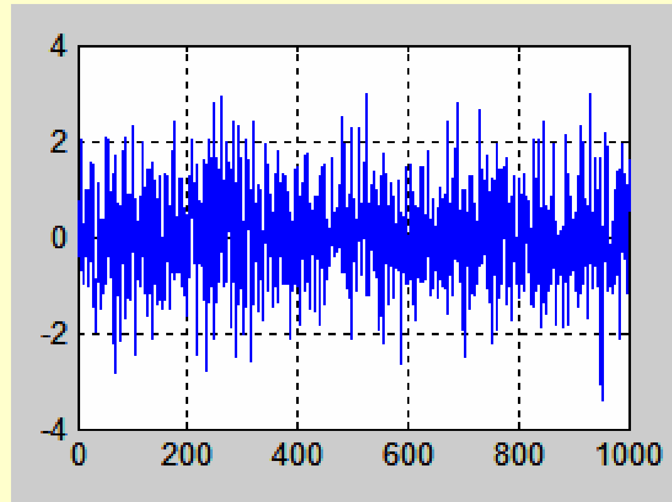
  Such processes are defined completely by variance μ and mean value x_o:

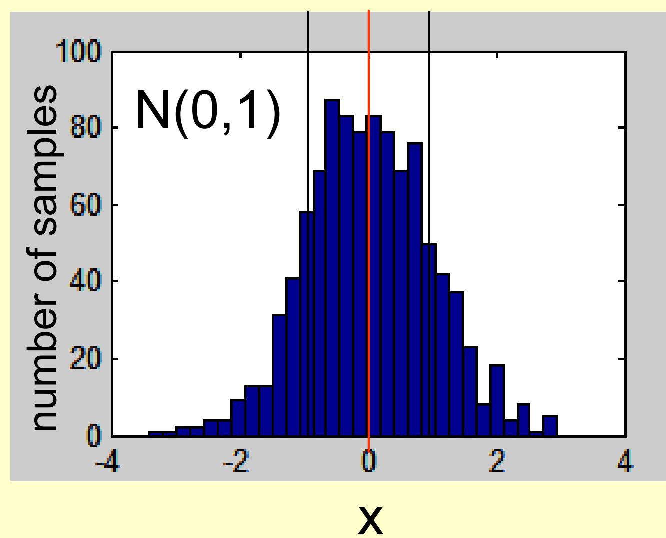
  $N(x_o,\mu) = x_o + \mathrm{sqrt}(\mu)\, N(0,1)$

Computing the variance from n samples:

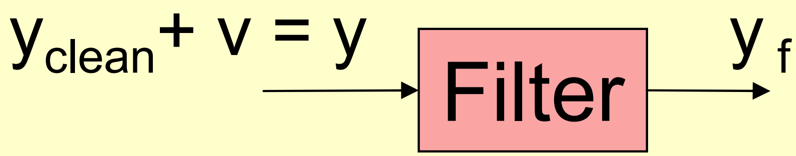
$\mu = [(x_1-x_o)^2 + (x_2-x_o)^2 + \ldots + (x_n-x_o)^2] / (n-1)$

1000 samples of a zero-mean, unit variance normal variable





Linear Filtering



Use good judgment!

filtering brings out trends, reduces noise BUT

filtering obscures dynamic response

Causal filtering: $y_f(t)$ depends only on *past measurements* – appropriate for real-time implementation

Example: $y_f(t) = (1-\varepsilon)\, y_f(t-\Delta t) + \varepsilon\, y(t)$ *("first-order lag")*

Acausal filtering: $y_f(t)$ depends on *all measurements*

– appropriate for post-processing

Example: $y_f(t) = [\, y(t+\Delta t) + y(t) + y(t-\Delta t)\,] / 3$ *("moving window")*

Convolution implies that the filter transfer function *F(s)* times the LaPlace transform of the input signal will give the LaPlace transform of the filter output:

$$Y_f(s) = F(s)\,[\, Y_{clean}(s) + V(s)\,]$$

Since a white noise process has uniform spectrum, the quantity $|F(j\omega)|$ determines what frequencies will get through → idea is to eliminate enough of the noise frequency band that the system dynamics can be seen.