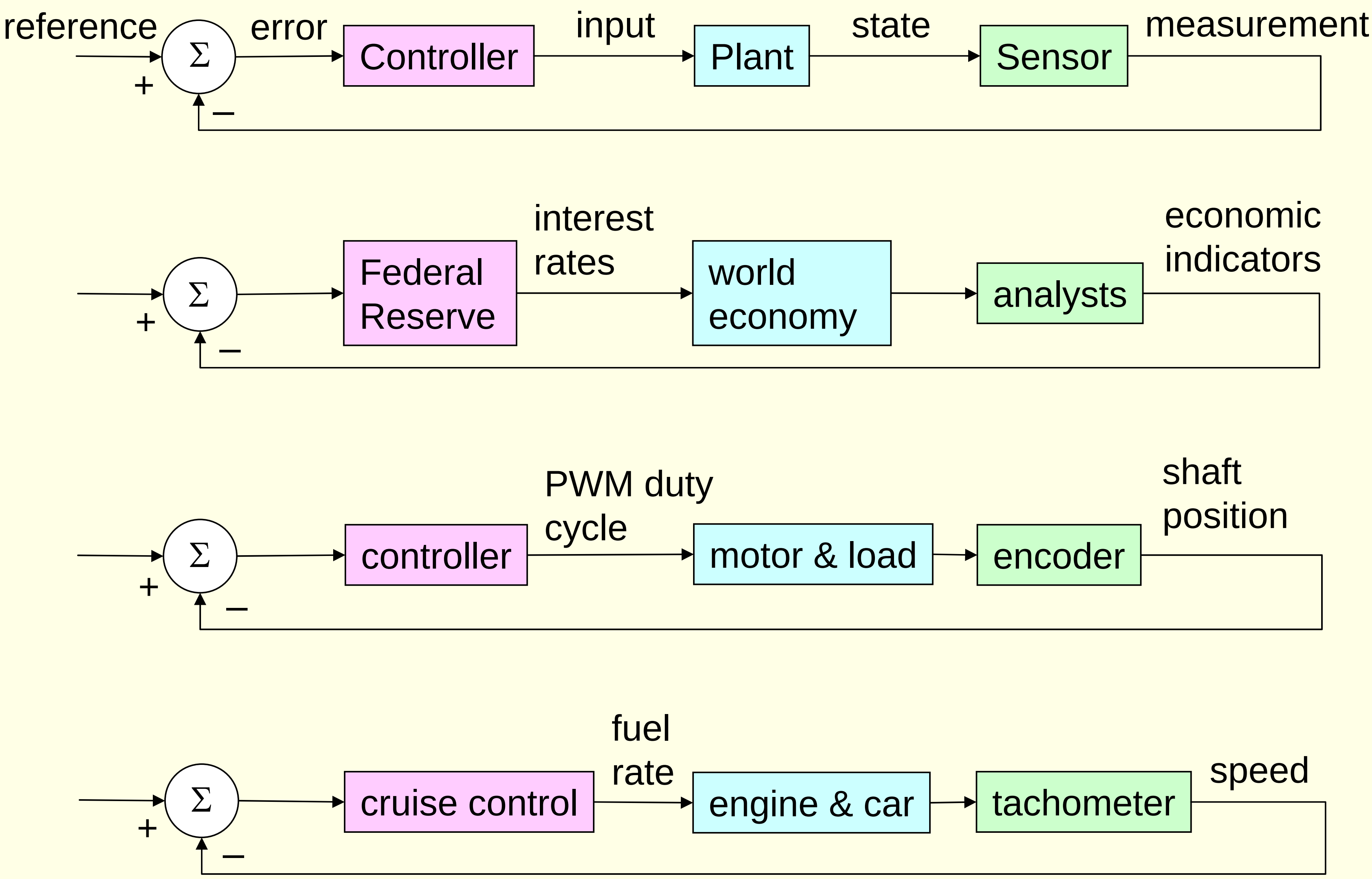
Feedback Control

Components of Engineered Feedback Systems

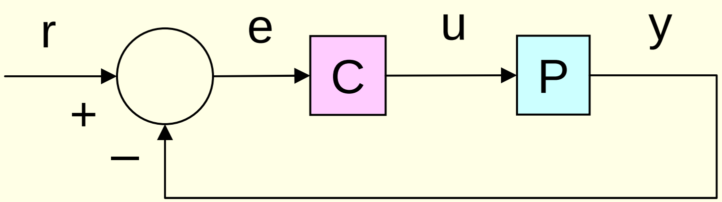
- **<u>Plant</u>**: the system whose behavior is to be controlled.
Examples: vehicle attitude, temperature, chemical process, business accounting, team and personal relationships, global climate
- **<u>Actuator</u>**: systems which alter the behavior of the plant.
Examples: motor, heater, valve, law enforcement (!), pump, FET, hydraulic ram, generator, US Mint
- **<u>Sensor</u>**: system which measures certain states of the plant.
Examples: thermometer, voltmeter, Geiger counter, opinion poll, balance sheet, financial analyst
- **<u>Controller</u>**: translates sensor output into actuator input.
Examples: computer, analog device, human interface, committee

- Extreme variability in time scales:
  - active noise cancellation requires ~100 *kiloHertz* sensing and actuation
  - Social Security is assessed and corrected at ~3 *nanoHertz* (10 years)

Feedback fundamentally creates a new dynamics!



Basics in the Frequency Domain



e = r – y

u = Ce = C(r-y)

y = Pu = PCe = PC(r-y) → (PC + 1)y = PCr ➔ **y / r = PC / (PC + 1)**

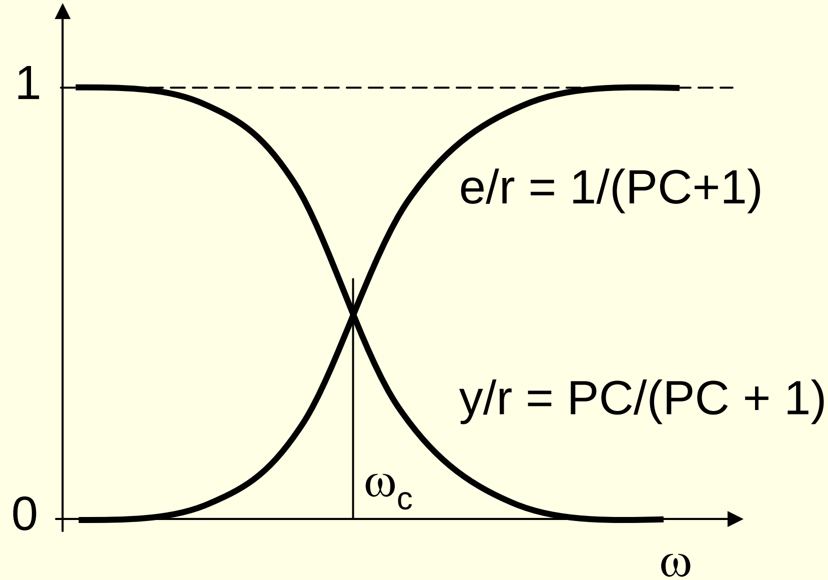
Similarly, e = r – y = r – PCe → (PC+1)e = r ➔ **e / r = 1 / (PC + 1)**

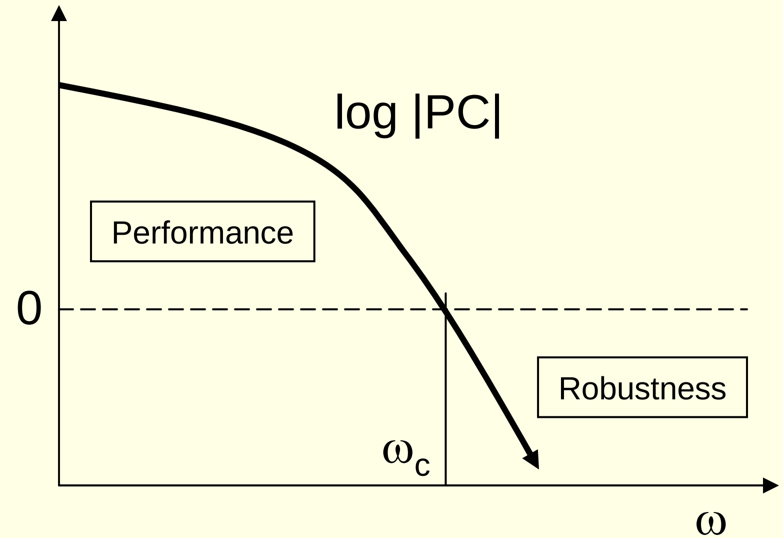
u = C(r-Pu) → (PC + 1)u = Cr ➔ **u / r = C / (PC + 1)**

Why can we do this? Convolution in time domain = Multiplication in freq. domain!

P <u>must</u> roll off at high frequencies – because no physical plant can respond to input at arbitrarily high frequency.

- Ideal case: e is a small fraction of r: e/r << 1, equivalent to y/r ~ 1
- This implies mag (PC + 1) >> 1 or mag (PC) >> 1.
- If plant P is given, then C has to be *designed* to make PC big.
- But mag (u / r) ~ mag(1 / P): HUGE when P gets small at high frequencies → excessive control action which will saturate or break actuators, excite unmodelled plant behavior, etc.. ← issues of *robustness*





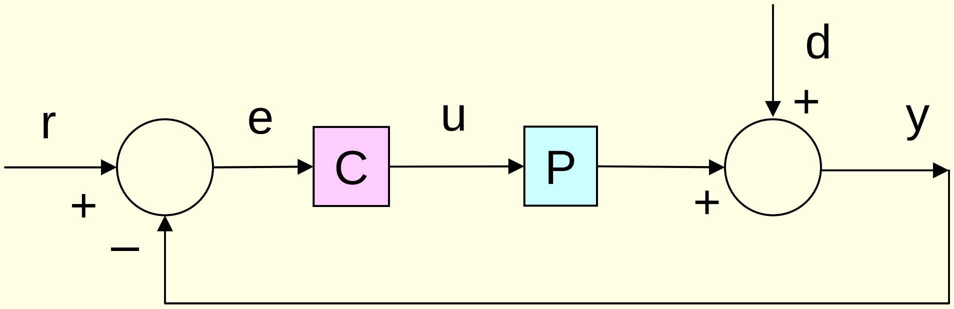
Good tracking only possible at low frequencies → leads to a “formula” for design:

Make |PC| *large at low frequencies*, e/r ~ 0, y/r ~ 1;
Good regulation and tracking at low frequencies

Make |PC| *small at high frequencies*, e/r ~ 1, y/r ~ 0, u/r ~ C
Poor tracking at high frequencies, but reasonable control action

The frequency where |PC| = 1 is the <u>crossover frequency</u> ω_c ;
Above this point, closed loop t.f. y/r = PC/(PC+1) drops off to zero.
So ω_c is about the *bandwidth* of the closed-loop t.f.

Random Physical Disturbances



e = r – y and u = Ce = C(r-y)
y = Pu + d = PCe + d = PC(r-y) + d →

With r = 0, (PC + 1)y = d ➔ **y / d = 1 / (PC + 1) (= - e / d also)**

competing!

u = C(r- Pu - d) →

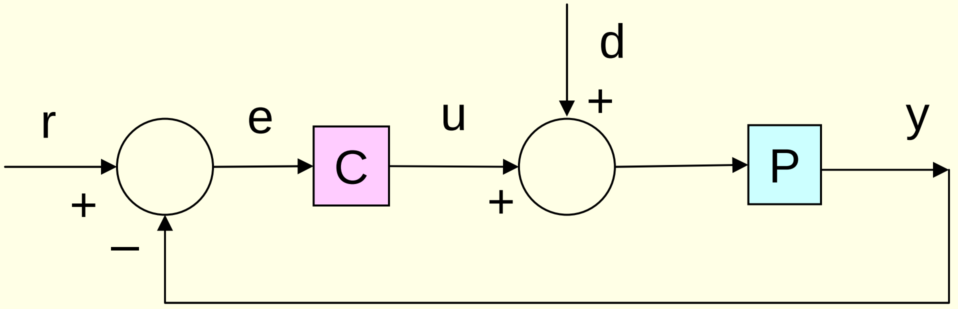
With r = 0, (PC + 1)u = -Cd ➔ **u / d = - C / (PC + 1)**

Because PC+1 is large at low frequencies, y/d will be small at low frequencies; the closed-loop system rejects low-frequency disturbances

- d is a random input, sometimes white or with frequency content, e.g., ocean waves!
- *Spectrum* of y when system is driven by random noise as in previous analysis:

$$S_y = [y/d]^* \; [y/d] \; S_d$$

- d can enter either at the plant output (as above), or at the plant input, i.e., it has the same units as control u. (Equations are different.)



LaPlace vs. Fourier XFM

Fourier Transform integrates **$x(t)\,e^{-j\omega t}$** over the time range from
<u>negative infinity</u> to positive infinity

Laplace Transform integrates **$x(t)\,e^{-st}$** over the time range from
<u>zero</u> to positive infinity

Result: $X(j\omega)$ can describe *acausal* systems, $X(s)$ describes only *causal* ones!

Many important results of Fourier Transform carry over to LaPlace Transform:

$\mathcal{L}(x(t)) = X(s)$ (notation)

$\mathcal{L}(ax(t)) = a\,X(s)$ (linearity)

$\mathcal{L}(x(t) * y(t)) = X(s)Y(s)$ (convolution)

$\mathcal{L}(x_t(t)) \leftrightarrow sX(s)$ (first time derivative)

$\mathcal{L}(x_{tt}(t)) \leftrightarrow s^2X(s)$ (second and higher time derivatives)

$\mathcal{L}(\int x(t)dt) \leftrightarrow X(s) / s$ (time integral)

$\mathcal{L}(\delta(t)) = 1$ (unit impulse)

$\mathcal{L}(1(t)) = 1/s$ (unit step)

LaPlace Transform and Stability

- For linear systems, stability of a system refers to whether the impulse response has *exponentially growing components*.
- *No pre-determined input can stabilize an unstable system; no pre-determined input can destabilize a stable system.*
- Some examples you can work out:

$$\mathcal{L}(e^{-\alpha t}) = 1 / (s + \alpha)$$

$$\mathcal{L}(t\, e^{-\alpha t}) = 1 / (s + \alpha)^2$$

$$\mathcal{L}[\, e^{-\alpha t} \sin(\omega t) \,] = \omega / (s^2 + 2\alpha s + \alpha^2 + \omega^2)$$

$$\mathcal{L}[\, \omega_d\, e^{-\zeta\omega n t} \sin(\omega_d t) / (1-\zeta^2) \,] = \omega_n^2 / (s^2 + 2\zeta\omega_n s + \omega_n^2)$$

Major observation: stable signal $\Leftrightarrow$ roots of $\mathcal{L}$ denominator have negative real parts: EQUALITY IS TRUE FOR ALL FIRST- AND SECOND-ORDER SYSTEMS

Decoding the transfer function

Numerator polynomials are a snap:

$(s + 2)/(s^2+s+5) = s/(s^2 + s + 5) + 2/(s^2+s+5)$

"input derivative plus two times the input, divided by the denominator"

For higher-order polynomials in the denominator: use partial fractions, e.g.,

$(s+1)/(s+2)(s+3)(s+4) = -0.5/(s+2) + 2/(s+3) -1.5/(s+4)$ *(all real poles)*

$(s+1)/s(s^2+s+1) = -s/(s^2+s+1) + 1/s$ *(some complex poles)*

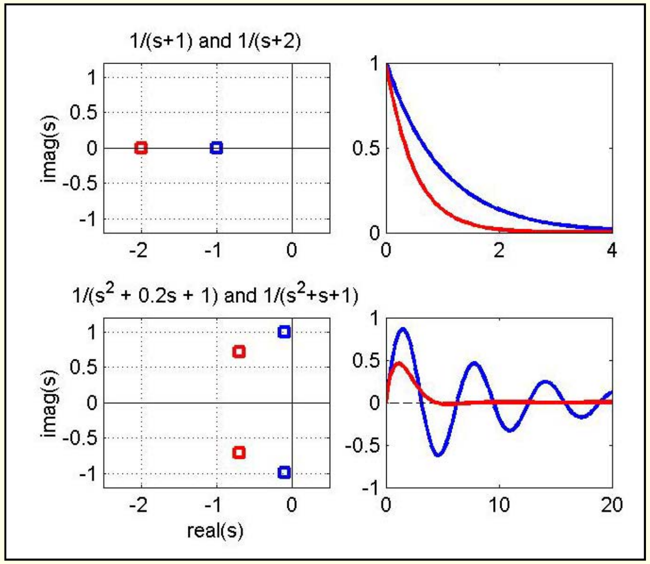
Any high-order transfer function can <u>always</u> be broken down into a sum of transfer functions with factored first- and second-order polynomials in the denominator.

stability ←→ the roots of the characteristic equation have negative real part.

More details:

real negative root $-\alpha$: the mode decays with time constant $1/\alpha$

complex roots at $-\omega_n\zeta$ +/- $j\omega_d$: the mode decays with frequency ω_d and exponential envelope having time constant $1/\zeta\omega_n$



Example with a double integrator: e.g., a motor or dynamic positioning

System is $mx_{tt}(t) = u(t)$ where:

- m is mass
- $x_{tt}(t)$ is double time derivative of position
- $u(t)$ is control action; thrust

Let a *Control law* be: **$u = - k_p x$ (Proportional Control: P)**

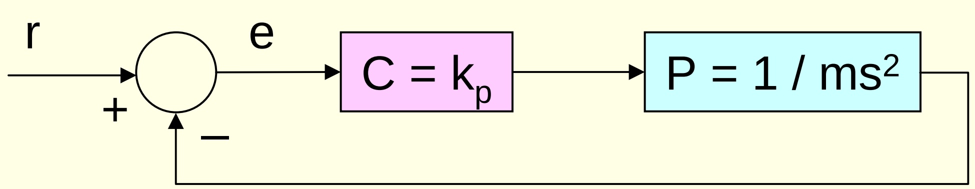
Closed-loop system dynamics become $mx_{tt} + k_p x = 0$

Response to an initial condition is undamped oscillations at frequency $\omega_n = \text{sqrt}(k_p/m)$

$P = 1/ms^2$

$C = k_p$

$PC = k_p/ms^2 \rightarrow$



$$e/r = 1/(PC + 1) = ms^2 / (ms^2 + k_p)$$

Tracking error is small when s is small; large when s is large, as desired.

BUT characteristic equation $ms^2 + k_p = 0$ has two imaginary poles – undamped!

Try the control law $u = -k_p x - k_d x_t$ **(Proportional + Derivative: PD)**

Closed-loop system dynamics become $m x_{tt} + k_d x_t + k_p x = 0$

Recall for a second-order underdamped oscillator:

$0 < k_d < 2\ \mathrm{sqrt}(k_p/m)$

$\omega_n = \mathrm{sqrt}(k_p/m)$ (undamped natural frequency)

$\zeta = k_d / 2\ \mathrm{sqrt}(k_p m)$ (damping ratio)

$\omega_d = \omega_n\ \mathrm{sqrt}(1-\zeta^2)$ (damped natural frequency)

Response to an initial condition is either:

- *Damped oscillations at frequency* $\omega_d = \mathrm{sqrt}(1-\zeta^2)\omega_n$, *inside an exponential envelope with time constant* $1/\zeta\omega_n$ OR
- *Sum of two decaying exponentials* (overdamped case)

Consider a <u>constant disturbance</u>: $mx_{tt} + k_d x_t + k_p x = F$;

System will settle at $x = F/k_p$; this is a steady-state error!

But k_p cannot be increased arbitrarily – natural frequency will be too high and too much control action

Try the control law $u = -k_p x - k_d x_t - k_i \int x \, dt$

(Proportional + Derivative + Integral: PID)

Closed-loop system dynamics become

$$mx_{tt} + k_d x_t + k_p x + ki \int x \, dt = F$$

If the system is stable ($ms^3 + k_d s^2 + k_p s + k_i = 0$ has roots with negative real part), then differentiate:

$mx_{ttt} + k_d x_{tt} + k_p x_t + k_i x = 0 \rightarrow$ settles to $x = 0$!

The PID

$C = k_p + k_d s + k_i/s$
$= (k_p s + k_d s^2 + k_i) / s$

High-frequency response is $\sim k_d s$; increases with frequency and disobeys the rule of finite power. High frequency errors will lead to very large control action!

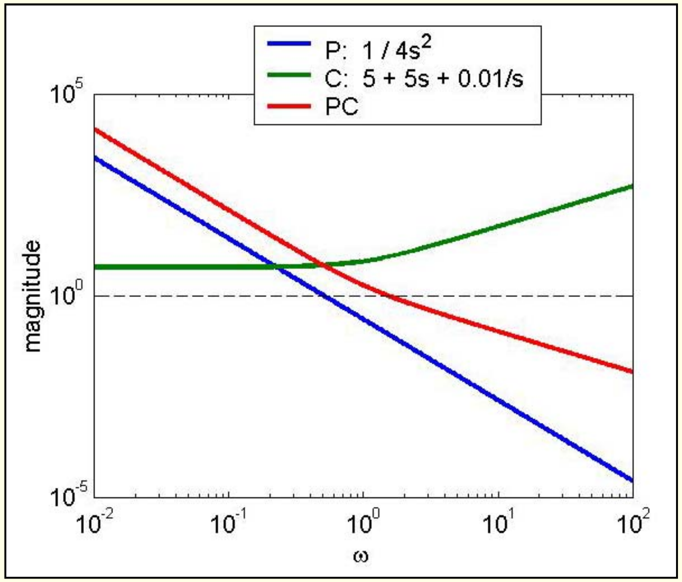
Sensor noise solutions:

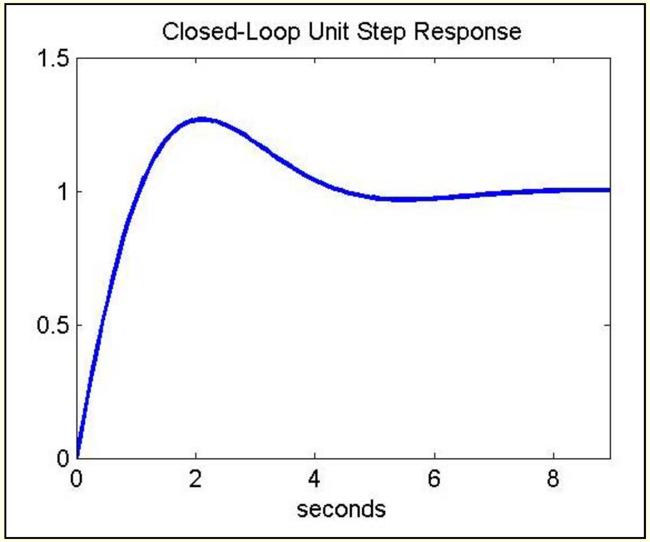
- use a very clean and high-res. sensor for x, which can be easily differentiated numerically, *e.g., motor encoder*
- use a sensor that measures dx/dt directly, *e.g., tachometer*
- filter the measurement. For a low-pass, we would get

$$C_f = [\ (k_p s + k_d s^2 + k_i) / s\ ]\ \ [\ \lambda / (s+\lambda)\ ]$$
$$= \lambda\ (k_p s + k_d s^2 + k_i) / s\ (s+\lambda)$$

But combine with a double integrator plant $\quad P = 1/ms^2$

$PC = m(k_p s + k_d s^2 + k_i) / s^3$, which *does* go to zero at high frequencies, as desired → the system does have a real bandwidth, which can be tuned.





Selected Application Notes

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 $K_d = 4.8K_u / T_u$

Explanation →

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*.

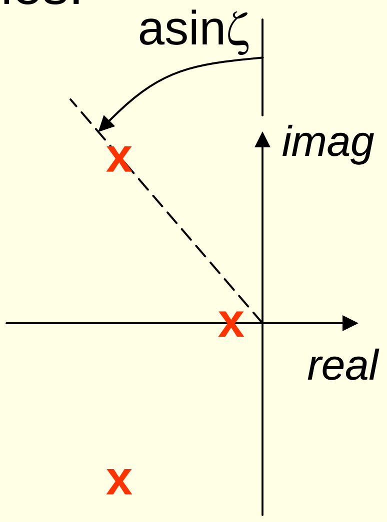
The characteristic equation with the Z-N PID gains becomes:

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

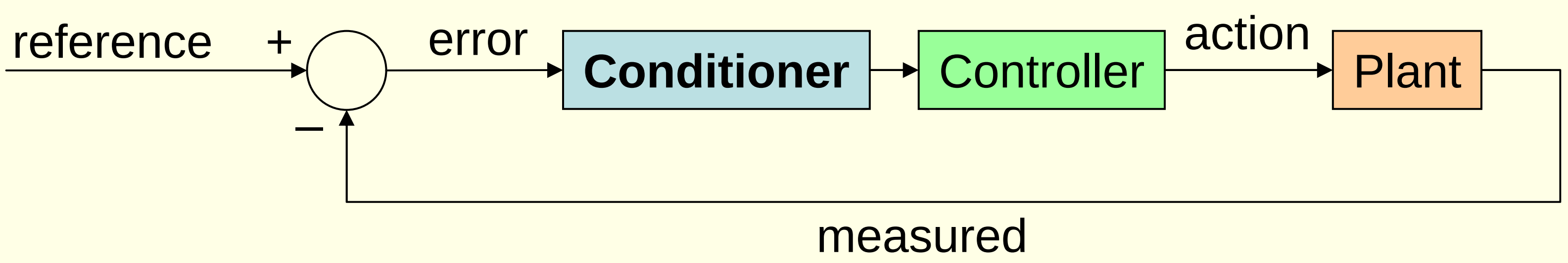
$$s^2 + 0 + \omega_n^2 + \mathbf{Q}\, [\, 0.6 + 1.2 / T_u / s + 4.8\, s / T_u \,] = 0$$

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

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:

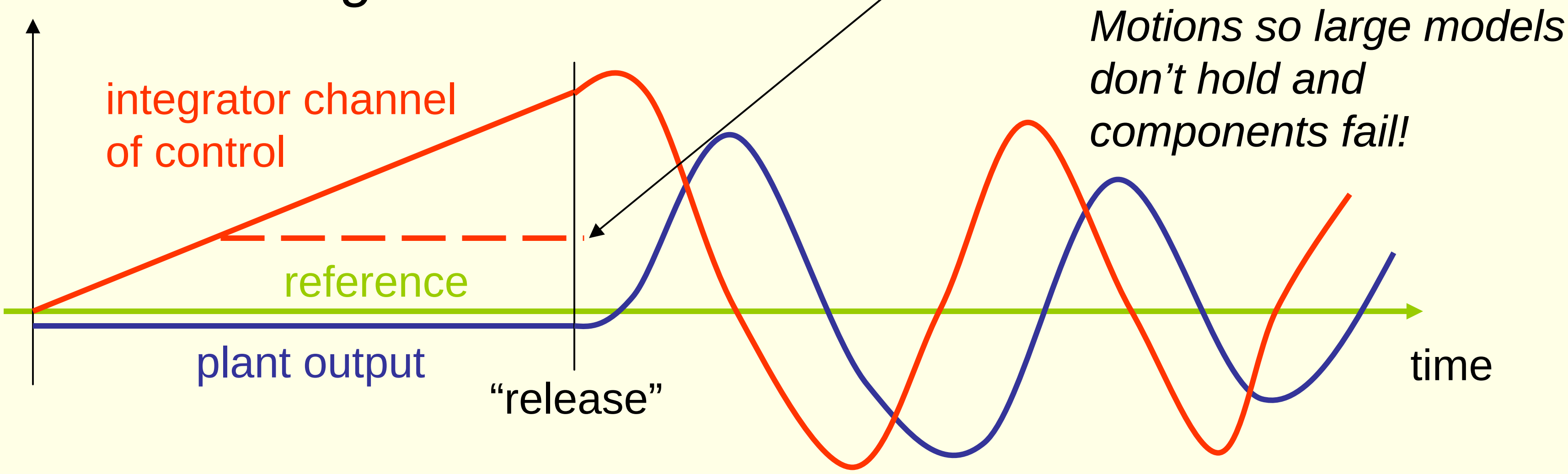
Subtract or add 2π to error to bring it into the range $[-\pi, \pi]$.

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:

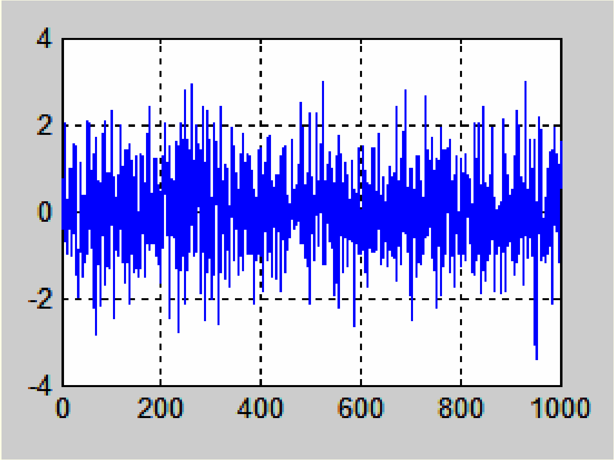
$N(0,1) = 1 / \mathrm{sqrt}(2\pi) * \exp [- x^2 / 2]$

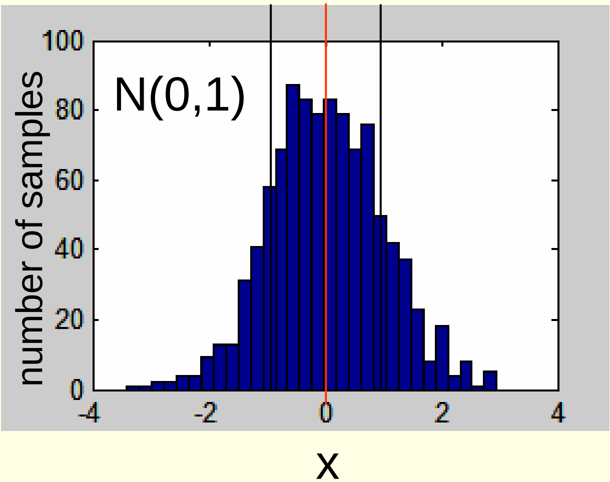
Such processes are defined completely by variance μ and mean value x_o:

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

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

$$y_{clean} + v = y \longrightarrow \boxed{\text{Filter}} \longrightarrow y_f$$

Use good judgment!

filtering brings out trends, reduces noise   BUT

filtering obscures dynamic response

<u>Causal filtering</u>: $y_f(t)$ 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")*

<u>Acausal filtering</u>: $y_f(t)$ 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. IMPACT ON CONTROL LOOP.