

Process Dynamics and Control

6. Multi-loop Systems

Kee-Youn Yoo

Department of Chemical & Biomolecular Engineering, SeoulTech

What are we up to?

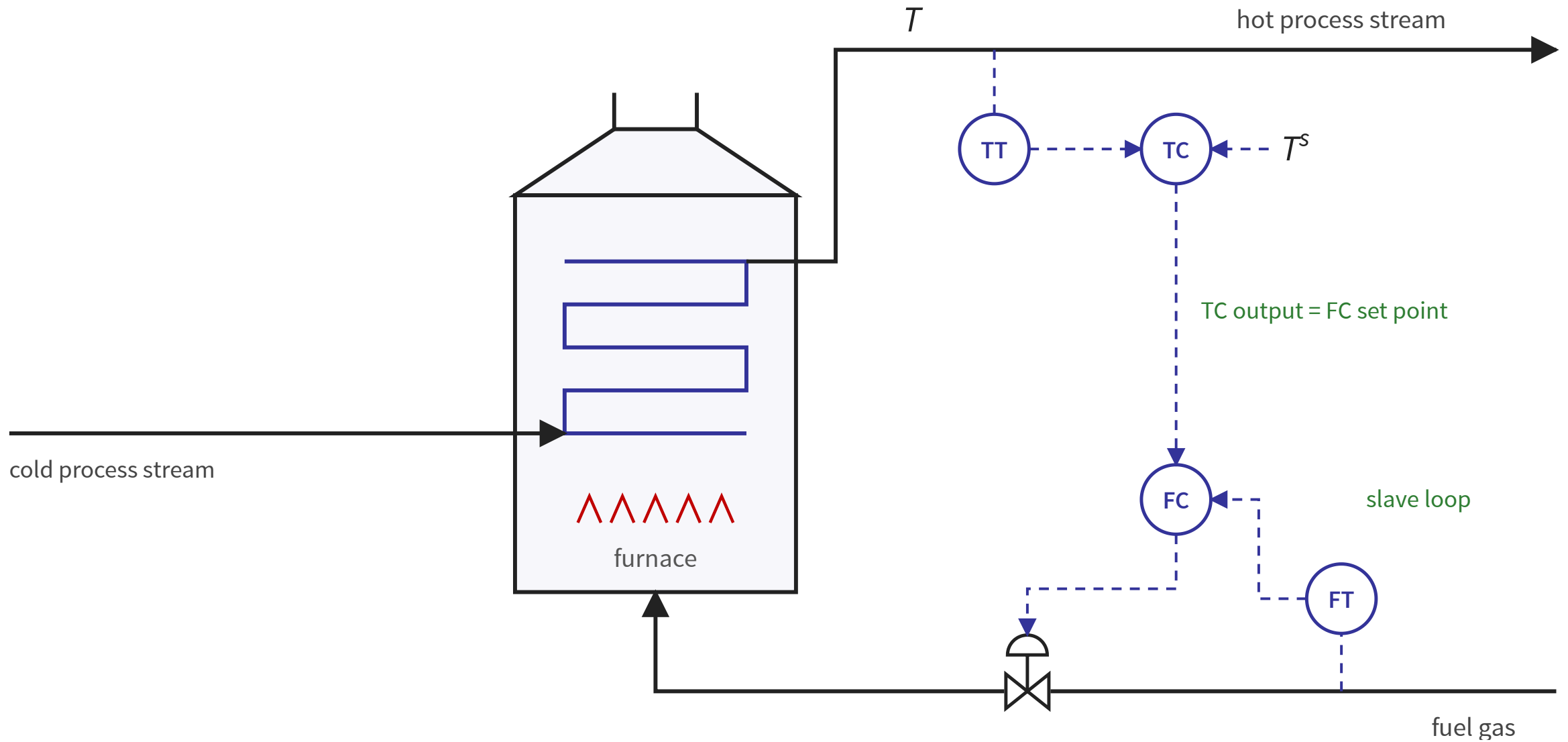
Single-loop feedback (Ch 4–5) reacts only **after** the controlled variable has moved. This chapter adds structure around the loop.

Strategy	What it handles	Key idea
Cascade (6.1)	disturbances in the manipulated variable	a fast inner loop fixes them before they reach the process
Feedforward (6.2–6.3)	measured disturbances	act on the disturbance itself, with a model
Ratio (6.4)	streams that must stay in proportion	one flow follows another
Smith predictor (6.5)	large dead time	feed back a <i>prediction</i> of the output
MIMO, RGA, decoupling (6.6–6.7)	several interacting loops	choose the pairing, then cancel the interaction

Running example for 6.1–6.4: the **fired heater** — outlet temperature controlled with the fuel-gas flow.

6.1 Cascade Control

The fired-heater problem



The temperature controller (TC) does **not** move the valve directly: its output is the **set point of a flow controller** (FC), which holds the fuel-gas flow against upstream pressure changes.

Master and slave loops

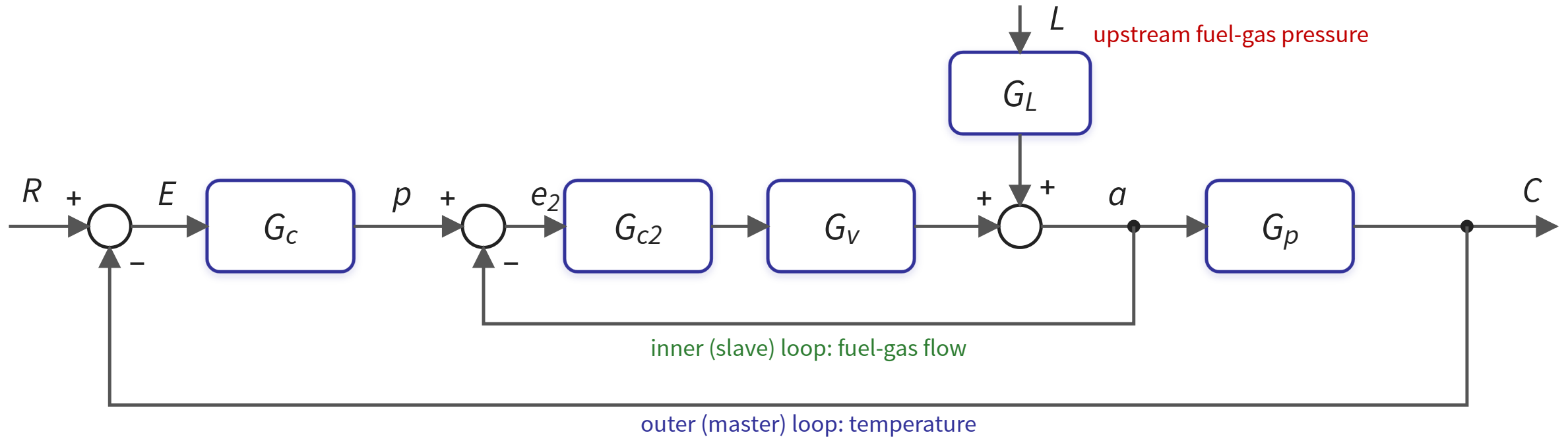
- **Master (primary) controller TC:** compares T with T^s and asks for a fuel-gas flow.
- **Slave (secondary) loop FC + FT + valve:** delivers that flow, correcting supply-pressure upsets **before** they change the temperature.

To be effective, the **slave loop must be much faster** (smaller time constant) than the outer loop.

Design rules

1. Tune the slave loop first, with the master in manual. Use as high a gain as feasible; P-only is usually enough.
2. Then tune the master loop on the closed slave loop. Integral action belongs here, because it removes the offset.

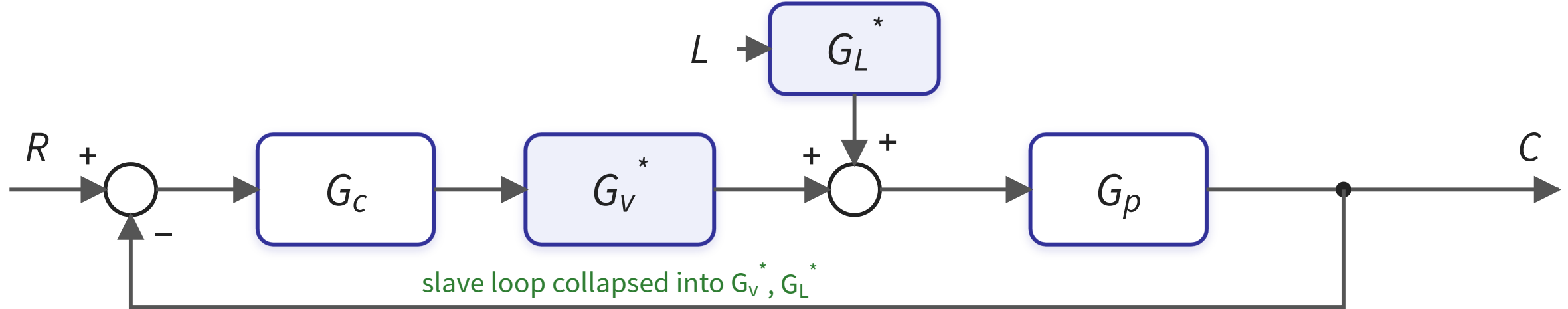
Block diagram



$$e_2 = p - a, \quad a = G_{c2}G_v e_2 + G_L L$$

$$a = \underbrace{\frac{G_{c2}G_v}{1 + G_{c2}G_v}}_{G_v^*} p + \underbrace{\frac{G_L}{1 + G_{c2}G_v}}_{G_L^*} L$$

Reduced block diagram



$$C = \frac{G_c G_v^* G_p}{1 + G_c G_v^* G_p} R + \frac{G_L^* G_p}{1 + G_c G_v^* G_p} L$$

The slave loop turns the valve G_v into a faster, more accurate actuator G_v^* , and it shrinks the load path to G_L^* .

Example 6.1 Choosing the slave gain K_{c2}

PI master, P slave: $G_p = \frac{0.8}{2s+1}$, $G_v = \frac{0.5}{s+1}$, $G_L = \frac{0.75}{s+1}$

$$G_v^* = \frac{K_{c2}K_v}{\tau_v s + 1 + K_{c2}K_v} = \frac{K_v^*}{\tau_v^* s + 1}$$

$$\tau_v^* = \frac{\tau_v}{1 + K_{c2}K_v}, \quad K_v^* = \frac{K_{c2}K_v}{1 + K_{c2}K_v}$$

$$K_L^* = \frac{K_L}{1 + K_{c2}K_v}$$

K_{c2}	τ_v^*	K_v^*	K_L^*
0	1	—	0.75
2	0.50	0.50	0.375
5	0.29	0.71	0.21
18	0.10	0.90	0.075
50	0.04	0.96	0.029

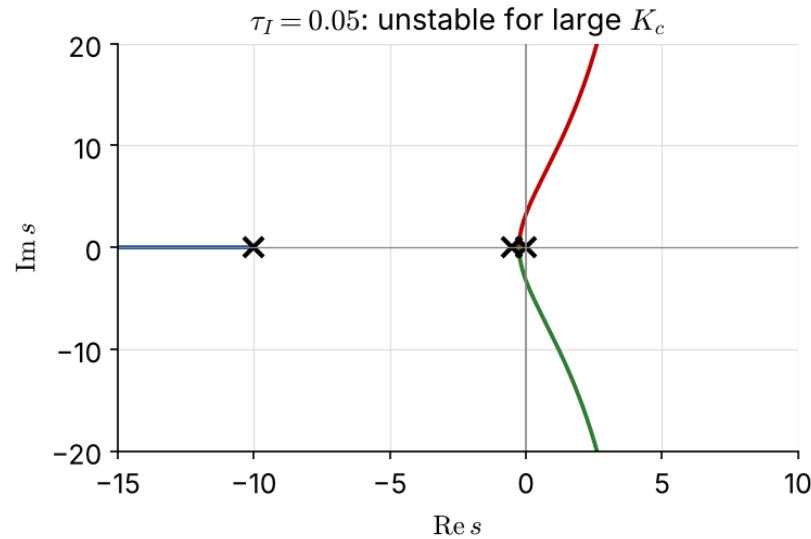
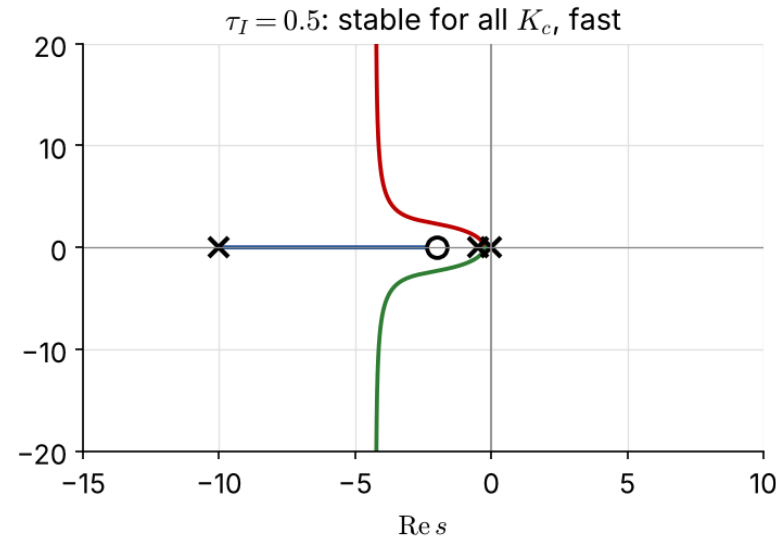
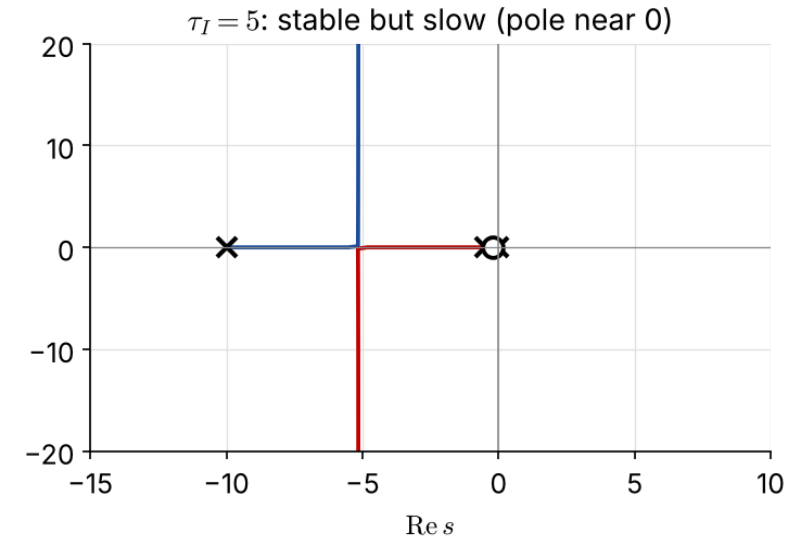
As K_{c2} grows: $K_v^* \rightarrow 1$ (the valve does what it is told),
 $K_L^* \rightarrow 0$ (load rejected), $\tau_v^* \rightarrow 0$ (faster).

(b) For $\tau_v^* = 0.1\tau_v$: $1 + 0.5K_{c2} = 10 \Rightarrow K_{c2} = 18$,
 giving $K_v^* = 0.9$ (10 % offset in the slave loop).

(c) Why P-only in the slave? The master PI removes the steady-state error in C ; the slave offset is harmless.

Example 6.1 (d): choosing the master τ_I

$$1 + K_c \frac{\tau_I s + 1}{\tau_I s} \frac{0.9}{0.1s + 1} \frac{0.8}{2s + 1} = 0$$



Poles $0, -0.5, -10$; PI zero at $-1/\tau_I$; $n - m = 2$, $\gamma = \frac{-10.5 + 1/\tau_I}{2}$

- $\tau_I = 0.05$: $\gamma = +4.75 \rightarrow$ unstable for $K_c > 1.54$
- $\tau_I = 0.5$: $\gamma = -4.25$, stable for all K_c , dominant poles far left
- $\tau_I = 5$: stable, but a closed-loop pole is trapped near the zero at $-0.2 \rightarrow$ slow

Choose $\tau_I = 0.5$: never unstable, and the fastest response.

Example 6.1 (e): without the cascade

$\tau_I = 0.5$, single loop:

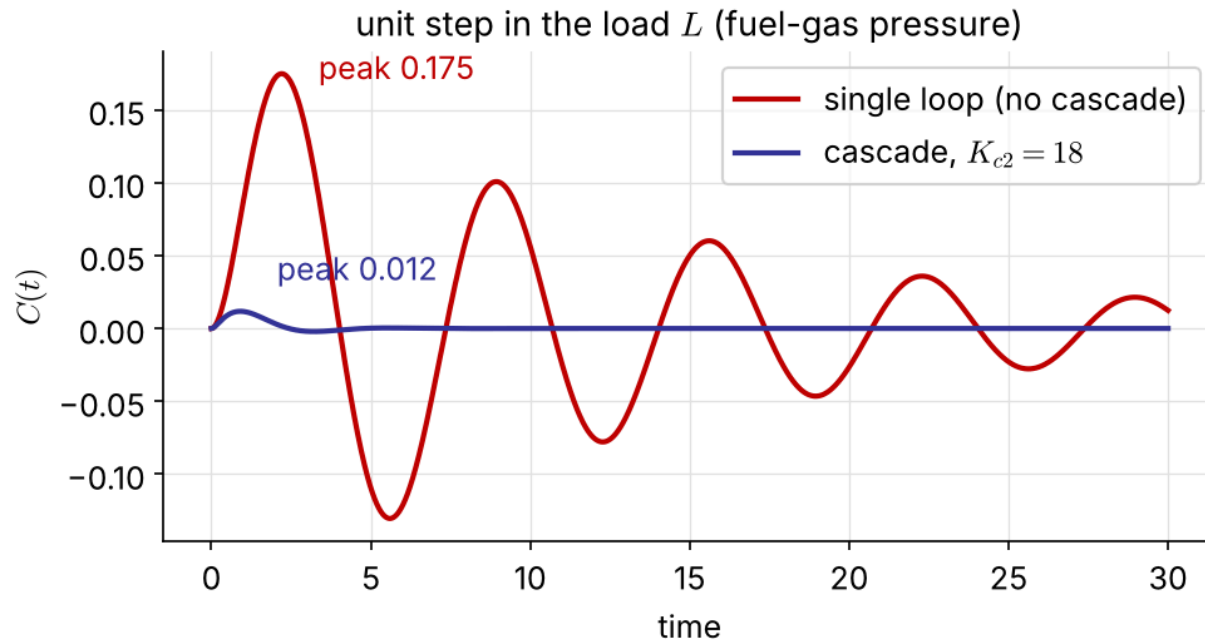
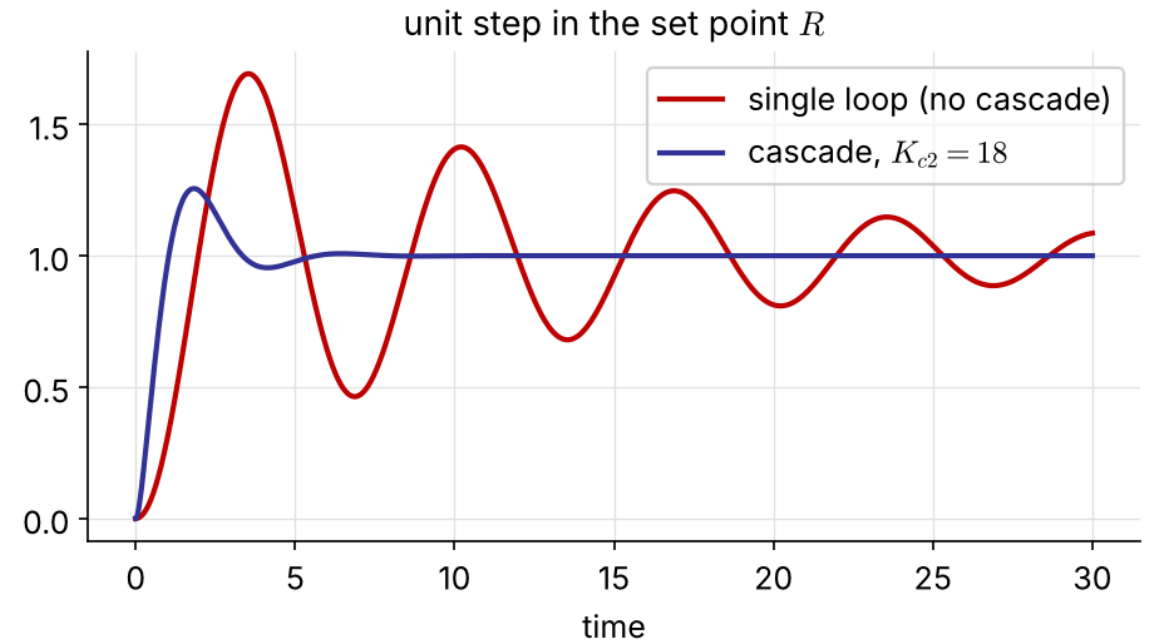
$$1 + K_c \frac{0.5s + 1}{0.5s} \frac{0.5}{s + 1} \frac{0.8}{2s + 1} = 0 \Rightarrow s^3 + 1.5s^2 + (0.5 + 0.2K_c)s + 0.4K_c = 0$$

Routh: $1.5(0.5 + 0.2K_c) > 0.4K_c \Rightarrow 0 < K_c < 7.5$

With the cascade the same master loop is stable for **every** $K_c > 0$. The slave loop removed the slow valve pole at -1 and replaced it with a fast one at -10 .

Example 6.1: responses (replaces the Simulink runs)

Ex 6.1: primary PI $K_c = 3$, $\tau_I = 0.5$



With the same master tuning ($K_c = 3$, $\tau_I = 0.5$), a fuel-gas pressure upset moves the temperature **15× less** with the cascade (peak 0.175 $\rightarrow$ 0.012). The set-point response is also much better damped.

Example 6.1 in code

Python

```

1 import numpy as np, control as ct
2
3 Gp, Gv, GL = ct.tf(0.8, [2, 1]), ct.tf(0.5, [1, 1]), ct.tf(0.75, [1, 1])
4 Kc2 = 18
5 Gv_s = ct.feedback(Kc2*Gv, 1)          # G_v* = 9/(s+10)
6 GL_s = ct.minreal(GL/(1 + Kc2*Gv))     # G_L* = 0.75/(s+10)
7
8 Kc, tI = 3, 0.5
9 Gc = ct.tf([Kc*tI, Kc], [tI, 0])
10 load_single = ct.minreal(GL*Gp/(1 + Gc*Gv*Gp))
11 load_cascade = ct.minreal(GL_s*Gp/(1 + Gc*Gv_s*Gp))
12 t = np.linspace(0, 30, 3001)
13 for T in (load_single, load_cascade):
14     print(ct.step_response(T, t).outputs.max()) # 0.175, 0.012

```

MATLAB

```

1 Gp = tf(0.8,[2 1]); Gv = tf(0.5,[1 1]); GL = tf(0.75,[1 1]);
2 Kc2 = 18;
3 Gv_s = feedback(Kc2*Gv, 1);          % G_v*
4 GL_s = minreal(GL/(1 + Kc2*Gv));     % G_L*
5
6 Kc = 3; tI = 0.5; Gc = Kc*tf([tI 1],[tI 0]);
7 load_single = minreal(GL*Gp/(1 + Gc*Gv*Gp));
8 load_cascade = minreal(GL_s*Gp/(1 + Gc*Gv_s*Gp));
9 step(load_single, load_cascade, 30), legend('single loop','cascade')

```

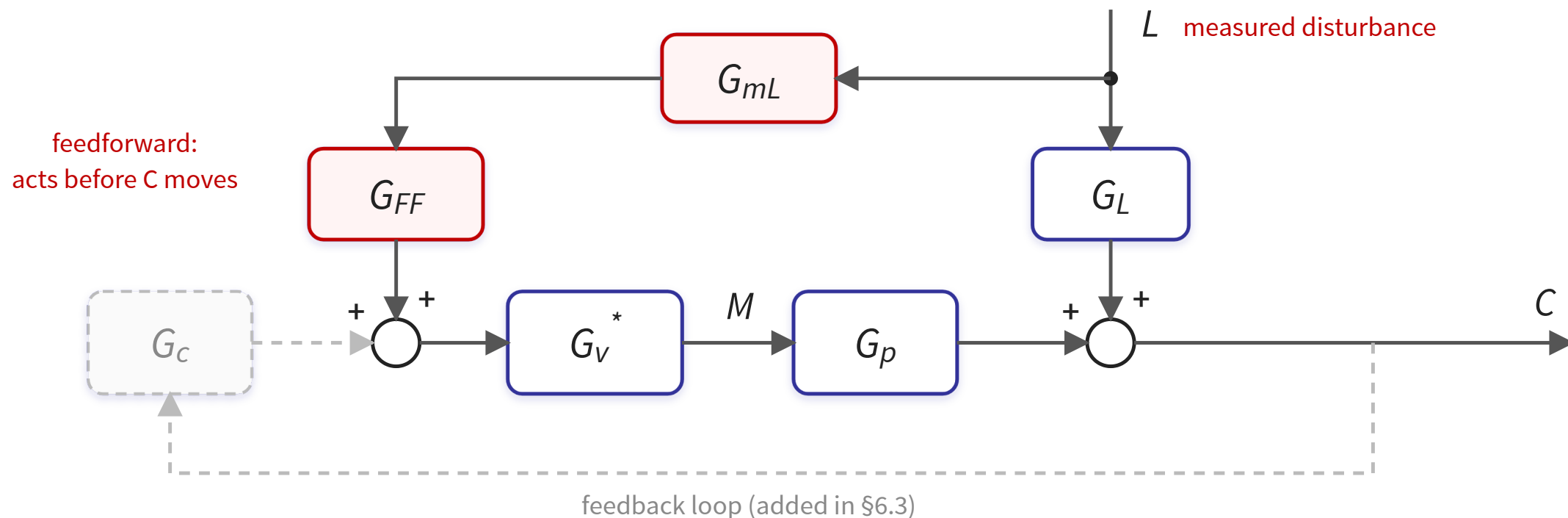
6.2 Feedforward Control

Act on the disturbance, not on the error

Measure the disturbance L and use a **process model** to move M before C is affected:

$$C = G_L L + G_p M \xrightarrow{C=R} M = \frac{1}{G_p} R - \frac{G_L}{G_p} L$$

- Set-point changes are normally left to the feedback loop; only the second term is implemented.
- The **feedforward controller** is the model-based rule $-G_L/G_p$.



The feedforward controller

$$C = [G_L + G_{mL}G_{FF}G_v^*G_p]L \xrightarrow{C=0} G_{FF} = -\frac{G_L}{G_{mL}G_v^*G_p}$$

(the original wrote F_{FF} for G_{FF})

Pitfalls of the ideal G_{FF} :

- often **not realizable**: the numerator order exceeds the denominator order
- a dead time in G_p larger than in G_L gives $e^{+\theta s}$, which would need the future
- it is only as good as the model

Practical form: lead-lag × gain

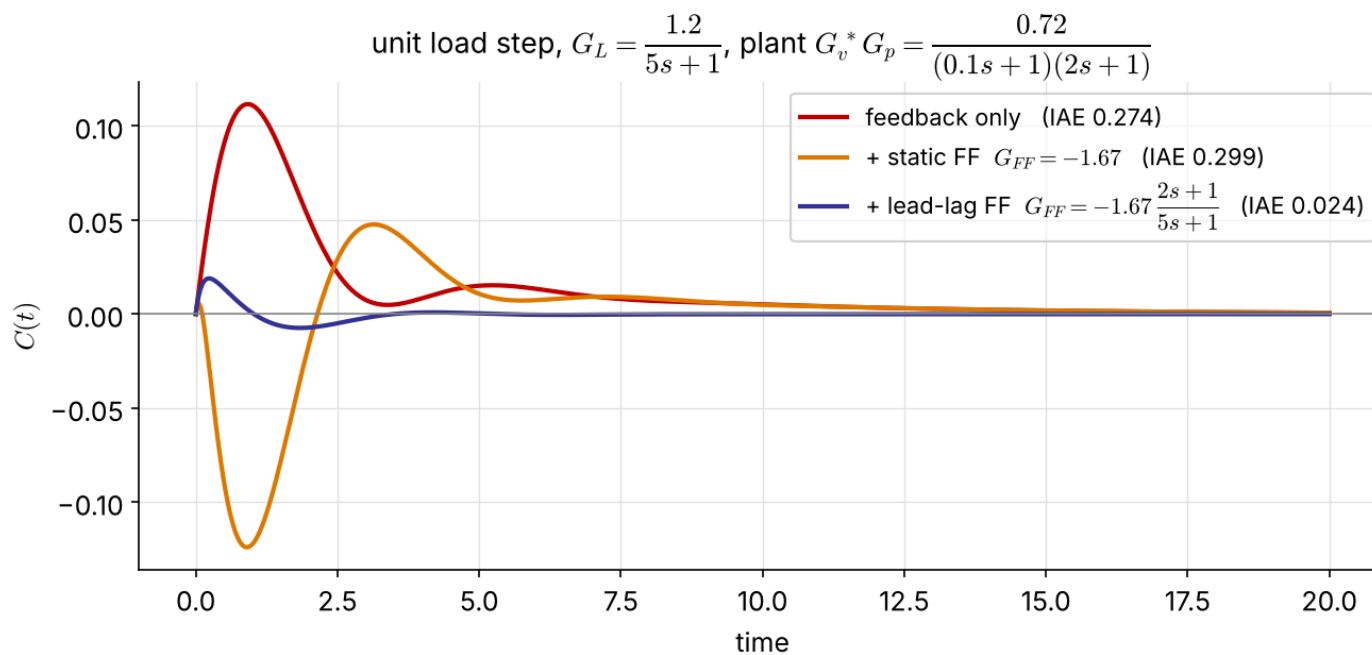
$$G_{FF} = K_{FF} \frac{\tau_{ld}s + 1}{\tau_{lg}s + 1}$$

$$K_{FF} = -\frac{K_L}{K_{mL}K_v^*K_p}$$

- static part K_{FF} : steady-state compensator
- lead-lag: dynamic compensator

If the measurement is fast and τ_v^* is small (a cascade slave loop), the static gain alone may be enough — **but check** (Ex 6.7).

Example 6.7 Static or dynamic feedforward?



Cascade plant of Ex 6.1 with a load at the output:

$$G_L = \frac{1.2}{5s+1}, G_{mL} = 1, \text{PI}$$

$$K_c = 3, \tau_I = 0.5$$

$$G_{FF}^{\text{ideal}} = -1.67 \frac{(0.1s+1)(2s+1)}{5s+1}$$

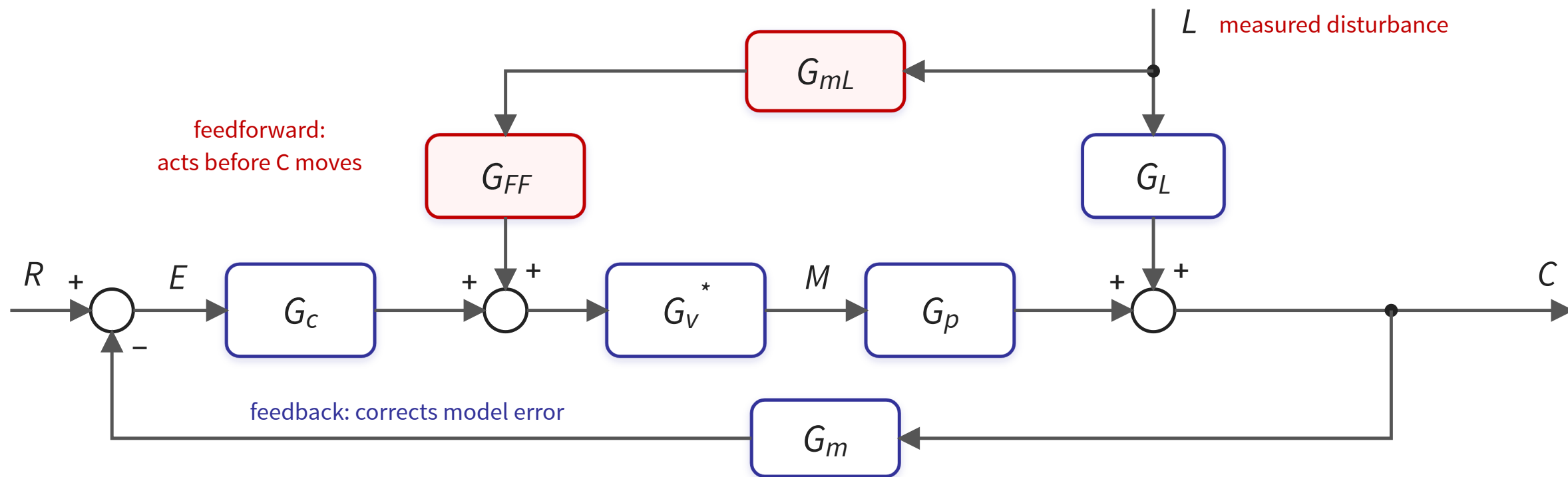
improper $\rightarrow$ drop the fast $(0.1s+1)$:

$$G_{FF} = -1.67 \frac{2s+1}{5s+1}$$

The **static** G_{FF} acts too early (the load is slower than the process) and makes things **worse** than feedback alone. The **lead-lag** cuts IAE **11×**.

6.3 Feedforward–Feedback Control

Feedforward never works alone

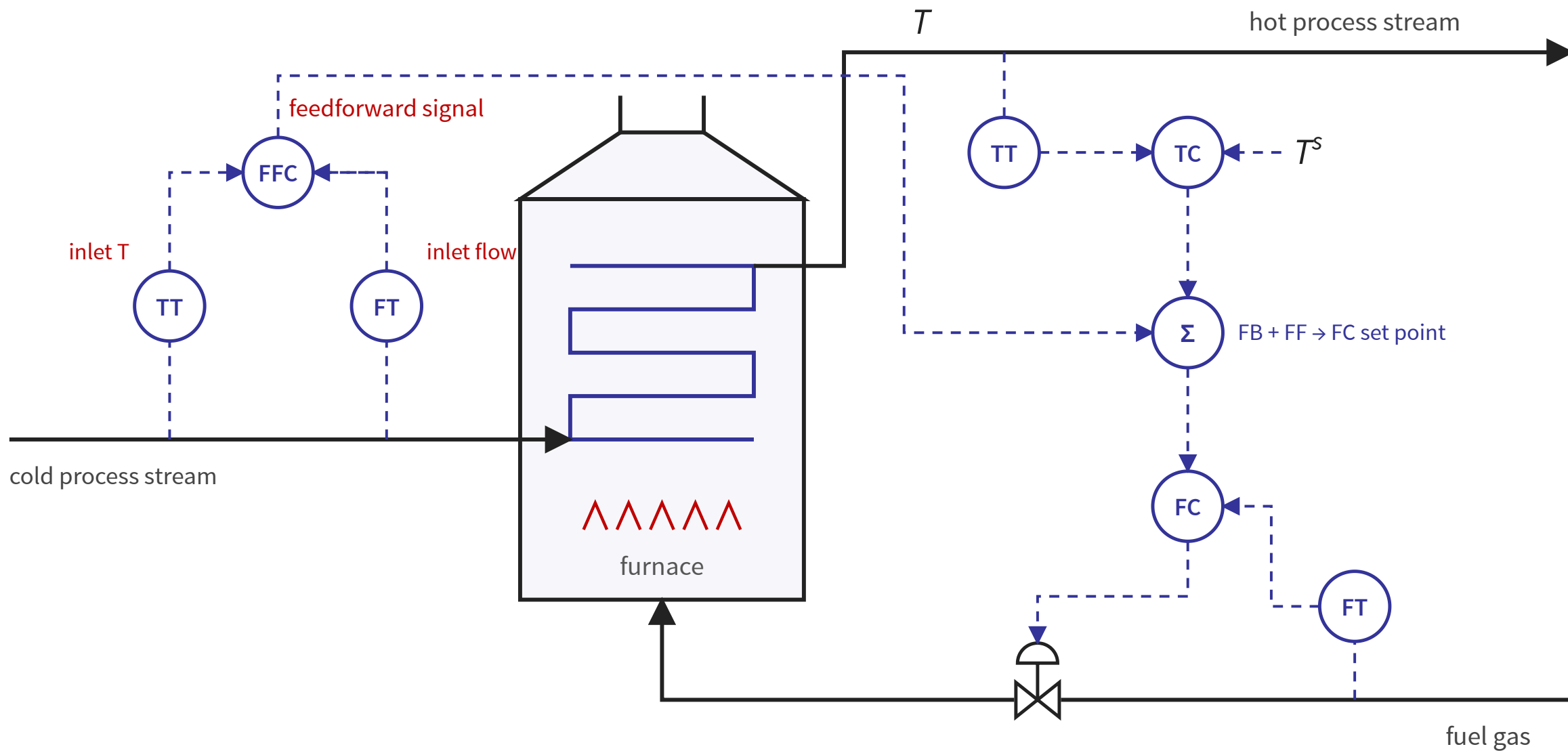


$$C = \frac{G_L + G_{mL}G_{FF}G_v^*G_p}{1 + G_mG_cG_v^*G_p} L + \frac{G_cG_v^*G_p}{1 + G_mG_cG_v^*G_p} R$$

Characteristic equation
 $1 + G_mG_cG_v^*G_p = 0$: G_{FF} does not affect stability.

Feedforward does most of the work; feedback removes what the imperfect model leaves behind.

Example 6.2 Furnace with feedforward, feedback and cascade

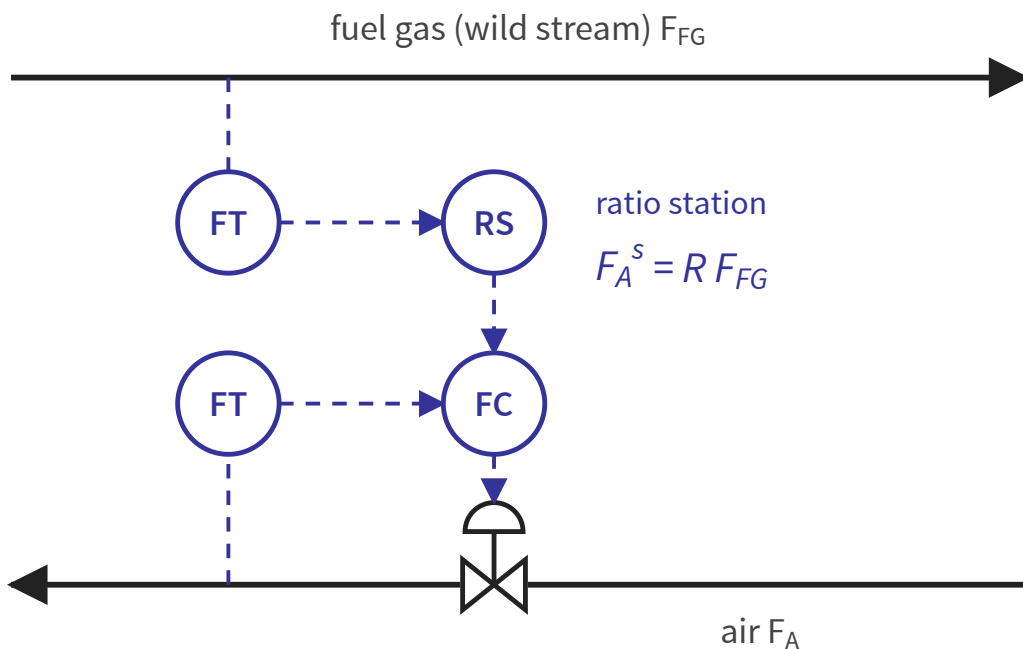


Inlet temperature and flow of the process stream (the main loads) feed the **FFC**; its signal is **added** (Σ) to the TC output. The sum is the set point of the fuel-gas **flow** controller, so cascade handles the fuel-supply upsets.

6.4 Ratio Control

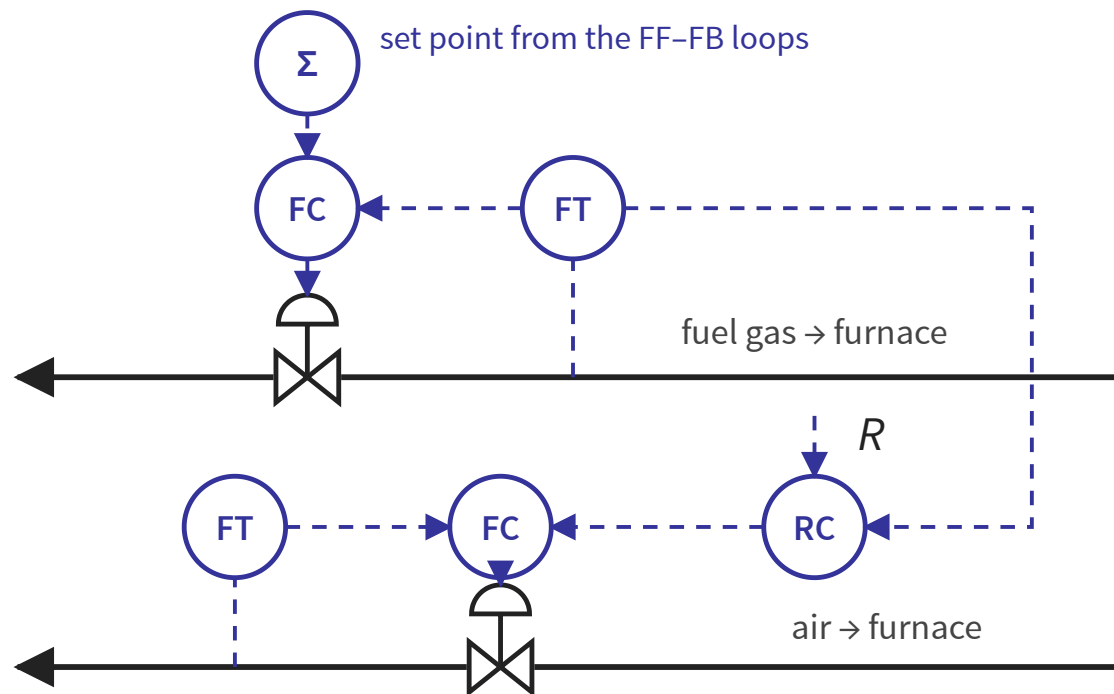
Keeping air in proportion to fuel

(a) simple ratio control



(a) **Simple ratio:** the fuel flow is “wild” (set elsewhere). The ratio station RS multiplies its measurement by $R = F_A / F_{FG}$ to give the air-flow set point.

(b) full-metering ratio control



(b) **Full metering:** both flows are controlled. The fuel set point comes from the FF–FB loops, and the ratio controller RC sets the air flow from the **measured** fuel flow.

Sufficient air means efficient combustion and lower emissions of CO and unburnt hydrocarbons.

6.5 The Smith Predictor

Dead time in the loop

$$1 + G_c(s)G(s)e^{-\theta s} = 0$$

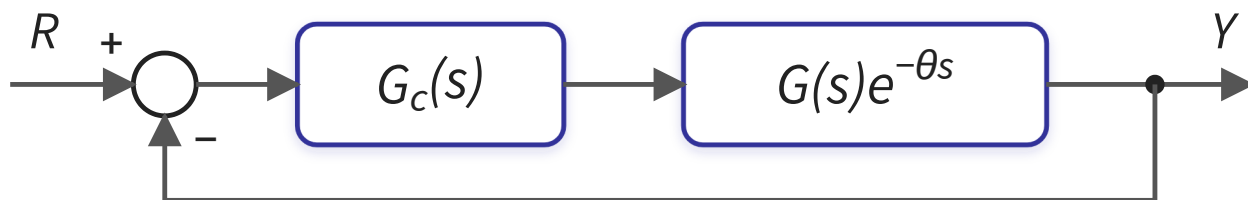
- Dead time adds phase lag $-\omega\theta$, lowers the gain margin, and forces a small K_c (Ch 5).
- The root cause is that the measured output is **outdated** information.

If $G(s)$ and θ are known, **predict** the undelayed output and feed that back instead.

Notation: the original slides use t_d ; here θ , as in Ch 4–5.

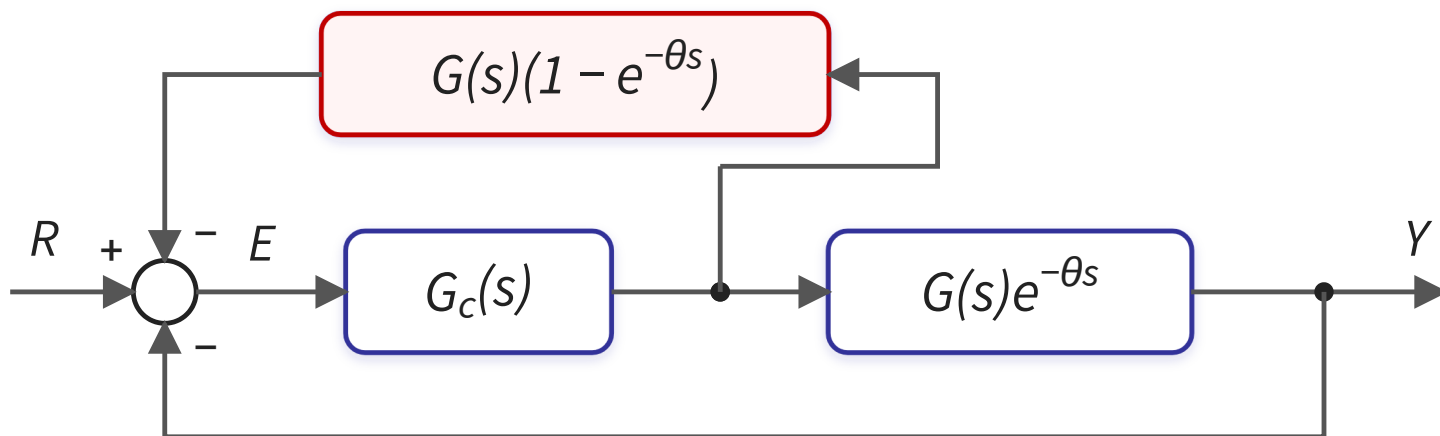
The Smith predictor

(a) feedback with dead time

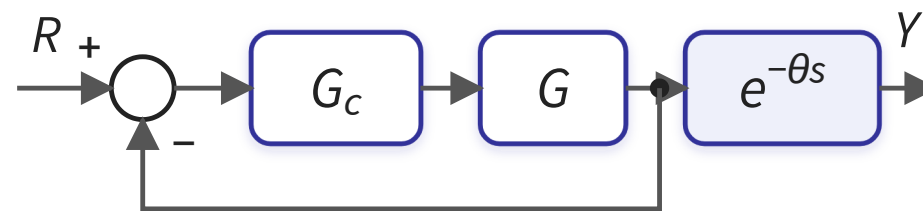


$$1 + G_c G e^{-\theta s} = 0: \text{delay in the loop}$$

(b) Smith predictor (dead-time compensator)



(c) equivalent loop



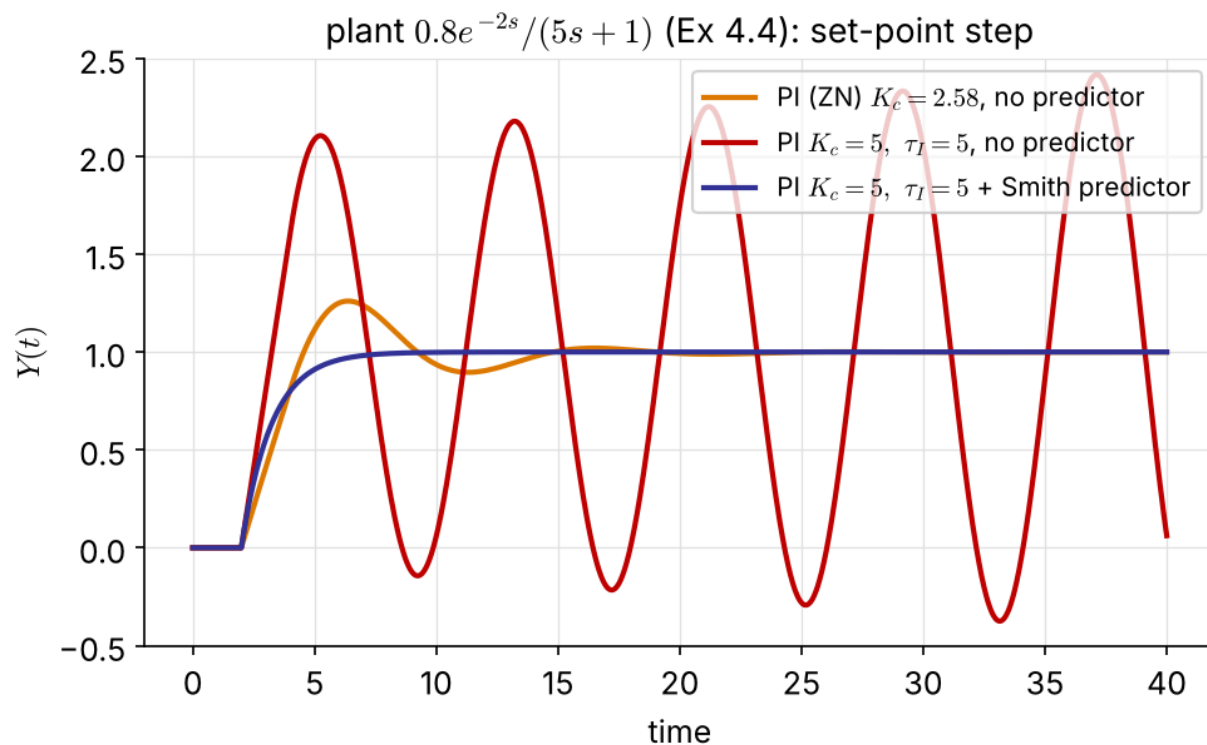
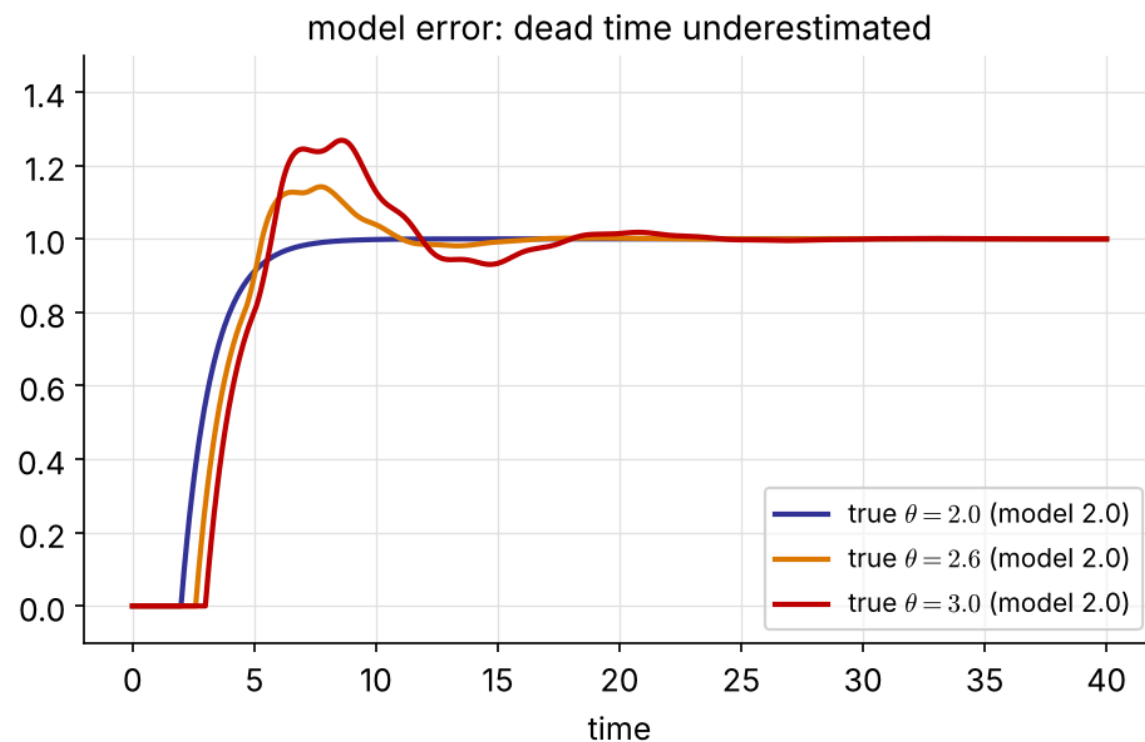
delay moved outside the loop
char. eq. $1 + G_c G = 0$

$$E = R - Y - G_c G (1 - e^{-\theta s}) E, \quad Y = G_c G e^{-\theta s} E$$

$$\Rightarrow E = R - G_c G E, \quad \frac{Y}{R} = \frac{G_c G}{1 + G_c G} e^{-\theta s}$$

The characteristic equation $1 + G_c G = 0$ has **no dead time**. Tune G_c as if the delay were absent; the response is the same, only shifted by θ .

Example 6.8 Smith predictor on the Ex 4.4 plant



$G = \frac{0.8e^{-2s}}{5s+1}$, $K_{cu} = 5.73$ (Ch 4-5). PI with $\tau_I = 5$ cancels the lag, and $K_c = 5$ gives a first-order loop with $\tau_{cl} = 5/(0.8K_c) = 1.25$.

- without the predictor, $K_c = 5$ is **unstable**
- with it: no overshoot, $Y(t)$ is the ideal response delayed by 2

Caveats: it needs an accurate model; a 50 % dead-time error brings back overshoot. For **load** disturbances the delay is not removed; the predictor only speeds up the correction.

Smith predictor in code

Python

```

1 from blocks import FOPDT, Delay, PI, DT      # exact dead time (no Pade); blocks.py in ch6/
2 import numpy as np
3
4 def smith(Kc, tI, theta_true=2.0, theta_model=2.0, tf=40):
5     G = FOPDT(0.8, 5, theta_true)           # plant 0.8 e^{-theta s}/(5s+1)
6     M0 = FOPDT(0.8, 5, 0.0)                 # model without delay
7     Md = Delay(theta_model)                 # model delay
8     c = PI(Kc, tI)
9     y = ym0 = ymd = 0.0; out = []
10    for k in range(int(tf/DT) + 1):
11        u = c(1.0 - y - (ym0 - ymd))         # feed back y + G(1 - e^{-theta s}) u
12        y, ym0 = G(u), M0(u); ymd = Md(ym0)
13        out.append(y)
14    return np.arange(len(out))*DT, np.array(out)
15
16 t, y = smith(5, 5)                         # Kc = 5 > Kcu: still stable

```

MATLAB

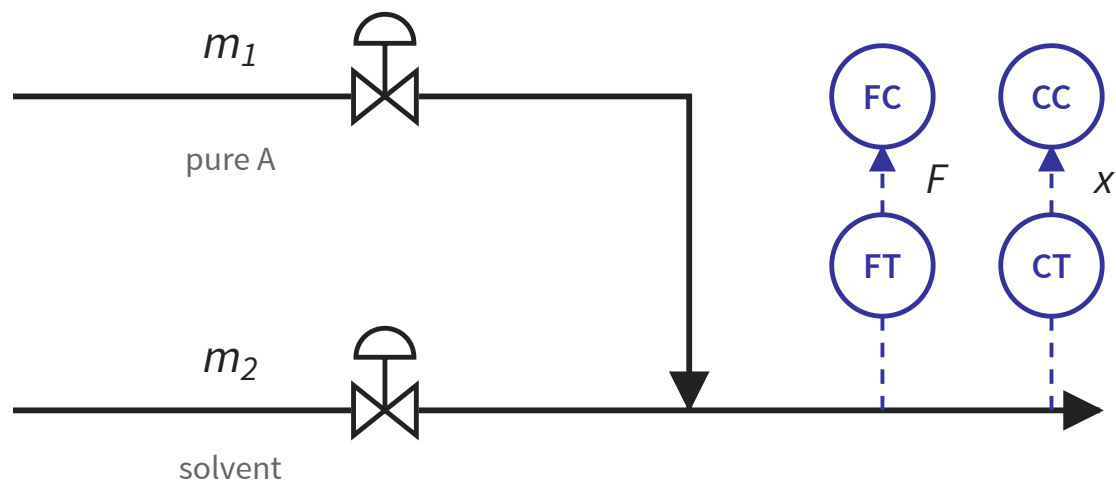
```

1 s = tf('s');
2 G0 = 0.8/(5*s + 1);                        % model without delay
3 Gd = ss(G0*exp(-2*s));                     % plant with delay (ss: internal delays)
4 Gc = 5*(1 + 1/(5*s));                      % PI, Kc = 5, tauI = 5
5
6 C_smith = feedback(Gc, G0 - Gd);            % controller + dead-time compensator
7 T_smith = feedback(C_smith*Gd, 1);          % = Gc*G0/(1+Gc*G0) * exp(-2s)
8 T_plain = feedback(Gc*Gd, 1);              % unstable
9 step(T_smith, 40)

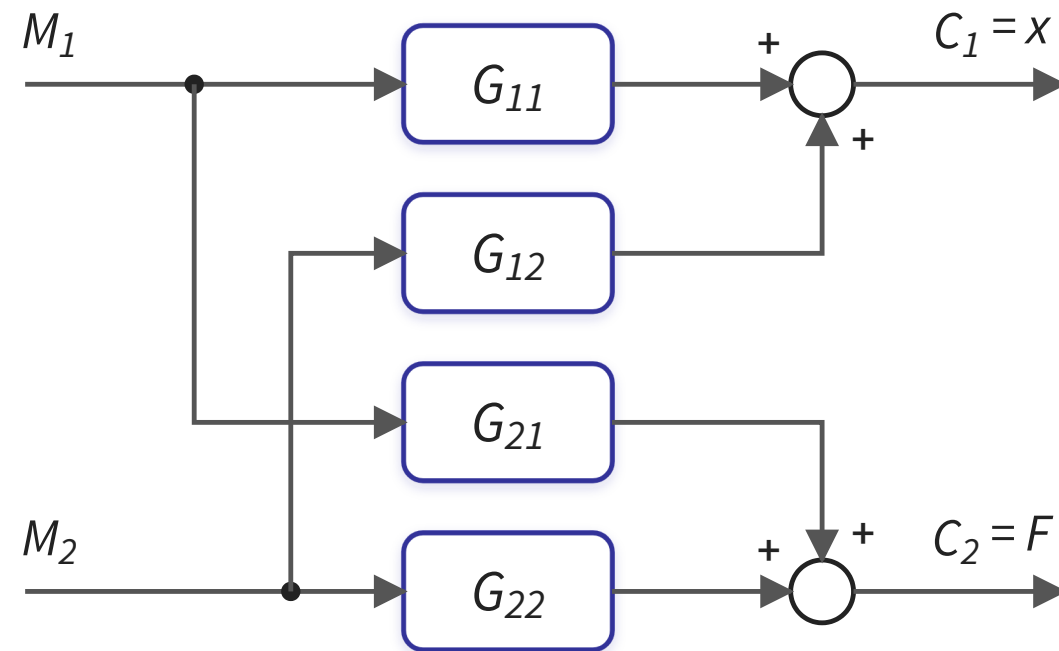
```

6.6 Multiple-Input Multiple-Output Control

A blending process



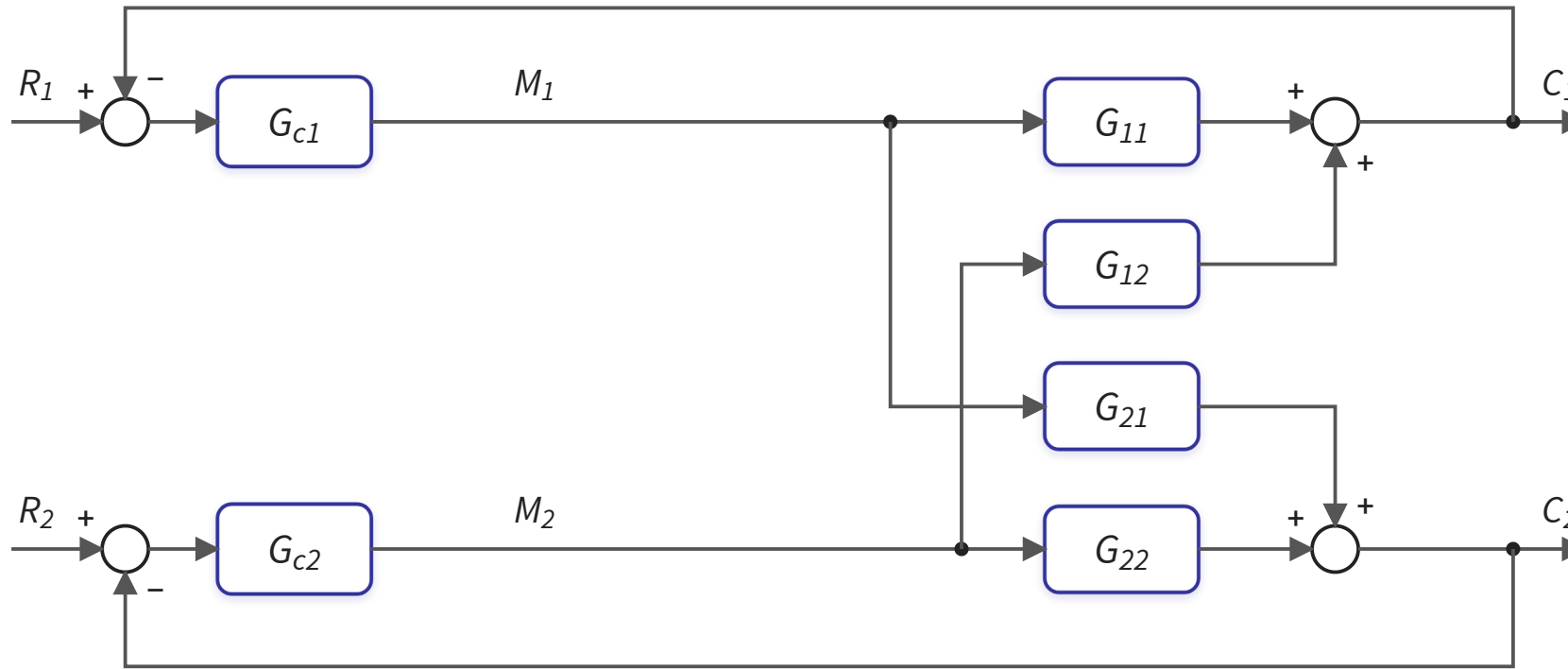
control total flow F and composition x
with the two valve flows m_1, m_2



$$C_1 = G_{11}M_1 + G_{12}M_2, \quad C_2 = G_{21}M_1 + G_{22}M_2$$

Every manipulated variable affects **both** outputs. The questions are: **which** M should control **which** C , and how strongly do the loops fight?

MIMO closed loop



With $M_i = G_{ci}(R_i - C_i)$, the closed loop is

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \frac{1}{p(s)} \begin{bmatrix} G_{11}G_{c1} + g(s) & G_{12}G_{c2} \\ G_{21}G_{c1} & G_{22}G_{c2} + g(s) \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}, \quad g = (G_{11}G_{22} - G_{12}G_{21})G_{c1}G_{c2}$$

$p(s) = (1 + G_{11}G_{c1})(1 + G_{22}G_{c2}) - G_{12}G_{21}G_{c1}G_{c2} = 0$: tuning one loop changes the stability of **both**.

Process gain matrix

Mass balances (m_1 pure A, m_2 solvent): $F = m_1 + m_2$, $x = \frac{m_1}{m_1 + m_2}$

$$x \approx x_s + \underbrace{\left[\frac{m_2}{(m_1 + m_2)^2} \right]_s}_{K_{11}} (m_1 - m_{1,s}) - \underbrace{\left[\frac{m_1}{(m_1 + m_2)^2} \right]_s}_{K_{12}} (m_2 - m_{2,s})$$

$$\begin{bmatrix} X \\ F \end{bmatrix} = \underbrace{\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}, \quad K_{21} = K_{22} = 1$$

Example 6.3 $m_1 = 0.1$, $m_2 = 10$ g/s:

$$\mathbf{K} = \begin{bmatrix} 9.8 \times 10^{-2} & -9.8 \times 10^{-4} \\ 1 & 1 \end{bmatrix}$$

$|K_{11}| \gg |K_{12}|$: m_2 barely affects x , a nearly **one-way** interaction.

Relative gain array (RGA)

$$\lambda_{ij} = \frac{\left. \frac{\partial c_i}{\partial m_j} \right|_{m_k, k \neq j}}{\left. \frac{\partial c_i}{\partial m_j} \right|_{c_k, k \neq i}} = \frac{\text{gain with all other loops open}}{\text{gain with all other loops closed (perfect control)}}$$

For a 2×2 system, with $\kappa = \frac{K_{12}K_{21}}{K_{11}K_{22}}$:

$$\lambda_{11} = \frac{1}{1 - \kappa}, \quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_{11} & 1 - \lambda_{11} \\ 1 - \lambda_{11} & \lambda_{11} \end{bmatrix}$$

Properties

- each row and column sums to 1
- independent of the units and scaling of the variables
- any size: $\mathbf{\Lambda} = \mathbf{K} \circ (\mathbf{K}^{-1})^T$ (element-wise product)

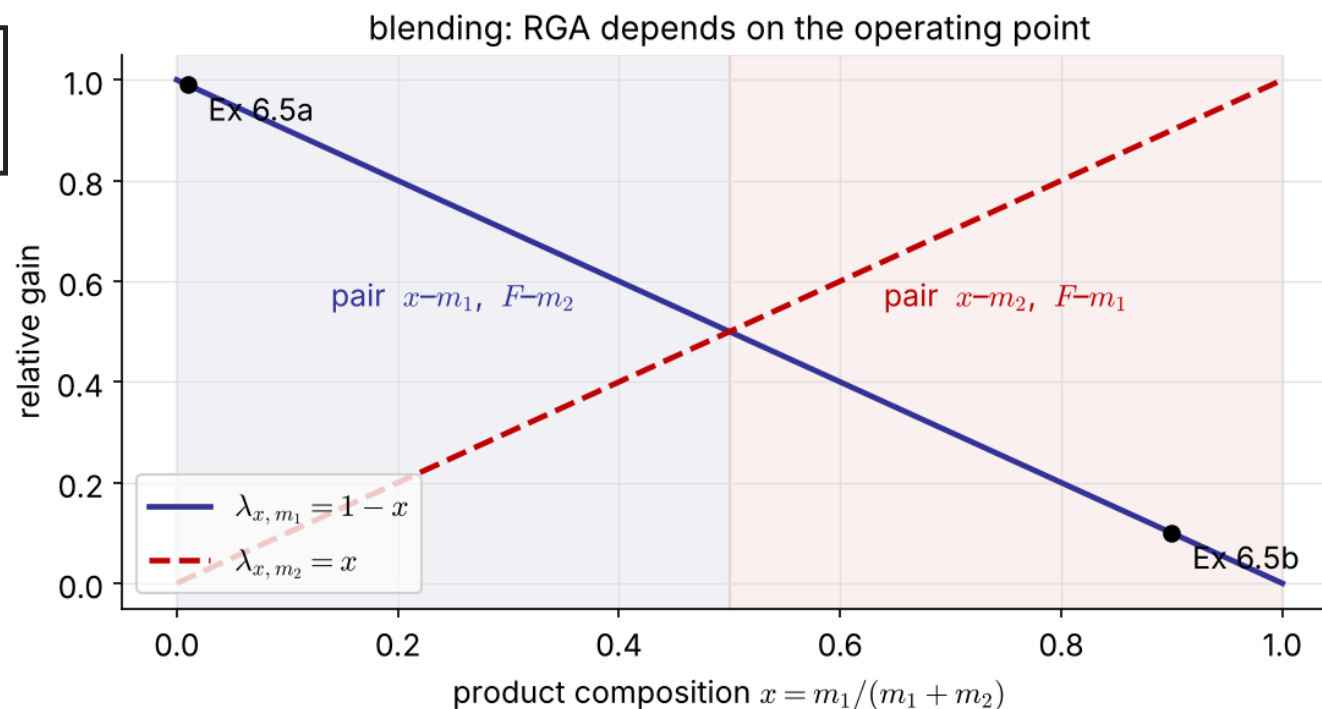
Example 6.4 RGA of the blending process

$$\left. \frac{\partial x}{\partial m_1} \right|_{m_2} = \frac{m_2}{(m_1 + m_2)^2}, \quad \left. \frac{\partial x}{\partial m_1} \right|_F = \frac{1}{F} = \frac{1}{m_1 + m_2} \Rightarrow \lambda_{x,m_1} = \frac{m_2}{m_1 + m_2} = 1 - x$$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_{x,m_1} & \lambda_{x,m_2} \\ \lambda_{F,m_1} & \lambda_{F,m_2} \end{bmatrix} = \begin{bmatrix} 1 - x & x \\ x & 1 - x \end{bmatrix}$$

Same result from the gains:

$$\kappa = K_{12}K_{21}/(K_{11}K_{22}) = -m_1/m_2$$



Reading the RGA: pairing rules

λ_{ij}	κ (2×2)	meaning	pair m_j-c_i ?
1	0	no interaction from the other loop	yes, ideal
0.5–1	< 0	closing the other loop raises this loop's gain	yes
> 1	0–1	closing the other loop lowers the gain	yes; avoid $\lambda \gg 1$
0.5	–1	the other loop doubles the gain; both pairings bad	—
0	—	m_j has no effect on c_i	no
< 0	> 1	the gain changes sign when the other loop closes	never

Rule: pair on relative gains that are **positive and closest to 1**. The **sign of κ** decides the case, not the signs of the gains (Ex 6.6: mixed signs, $\lambda = 1.91$).

Example 6.5 Pairing depends on the operating point

(a) $m_1 = 0.1$, $m_2 = 10$ g/s ($x = 0.0099$)

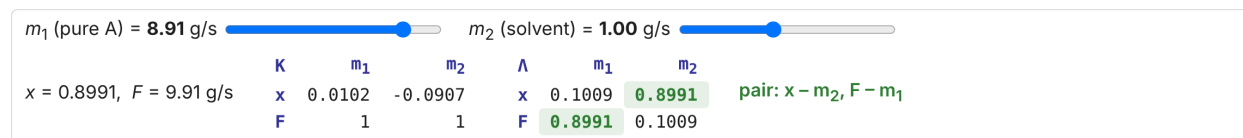
$$\Lambda = \begin{bmatrix} \mathbf{0.9901} & 0.0099 \\ 0.0099 & \mathbf{0.9901} \end{bmatrix}$$

pair x - m_1 , F - m_2

(b) $m_1 = 9$, $m_2 = 1$ g/s ($x = 0.9$)

$$\Lambda = \begin{bmatrix} 0.1 & \mathbf{0.9} \\ \mathbf{0.9} & 0.1 \end{bmatrix}$$

pair x - m_2 , F - m_1



Control the composition with the **smaller** stream: it changes x more and F less.

RGAs in code

Python

```

1 import numpy as np
2
3 def rga(K):
4     K = np.asarray(K, float)
5     return K * np.linalg.inv(K).T           # Lambda = K o (K^-1)^T
6
7 m1, m2 = 0.1, 10.0; F = m1 + m2
8 K = [[m2/F**2, -m1/F**2],
9       [1.0,      1.0    ]]
10 print(rga(K))                             # [[0.9901 0.0099] [0.0099 0.9901]]
11
12 K66 = [[0.07, -0.05], [0.1, -0.15]]
13 print(rga(K66))                           # distillation, Ex 6.6
                                           # [[ 1.91 -0.91] [-0.91  1.91]]

```

MATLAB

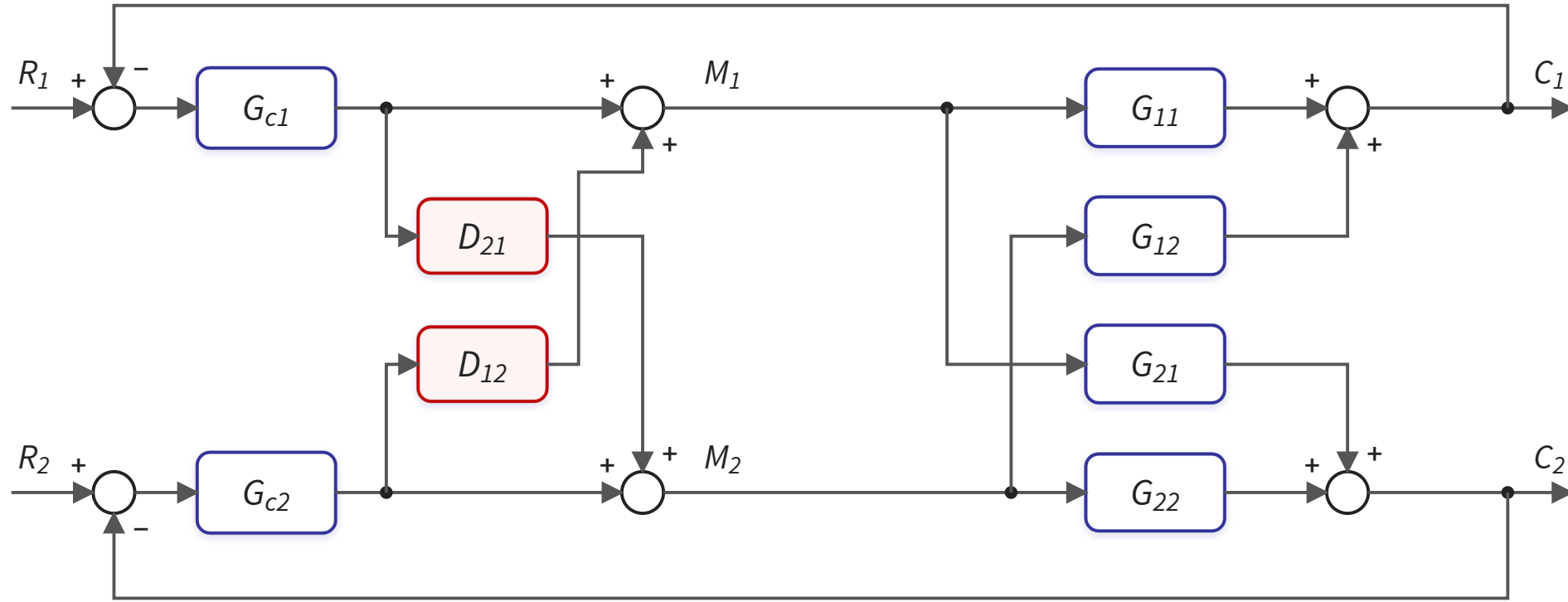
```

1 rga = @(K) K .* inv(K).';                 % Lambda = K o (K^-1)^T
2
3 m1 = 0.1; m2 = 10; F = m1 + m2;
4 K = [m2/F^2, -m1/F^2;
5       1,      1    ];
6 rga(K)                                     % [0.9901 0.0099; 0.0099 0.9901]
7
8 K66 = [0.07 -0.05; 0.1 -0.15];
9 rga(K66)                                  % distillation, Ex 6.6
                                           % [1.91 -0.91; -0.91 1.91]

```

6.7 Decoupling of Interacting Systems

Feedforward-like decouplers



Cancel the effect of controller 2 on C_1 :

$$G_{12} + G_{11}D_{12} = 0 \Rightarrow D_{12} = -\frac{G_{12}}{G_{11}}$$

Cancel the effect of controller 1 on C_2 :

$$G_{21} + G_{22}D_{21} = 0 \Rightarrow D_{21} = -\frac{G_{21}}{G_{22}}$$

Implement them like feedforward controllers: lead-lag plus gain, dropping any $e^{+\theta s}$ term that would need the future.

Example 6.6 Distillation column

$$\begin{bmatrix} X_D \\ X_B \end{bmatrix} = \begin{bmatrix} \frac{0.07e^{-3s}}{12s+1} & \frac{-0.05e^{-s}}{15s+1} \\ \frac{0.1e^{-4s}}{11s+1} & \frac{-0.15e^{-2s}}{10s+1} \end{bmatrix} \begin{bmatrix} L \\ V \end{bmatrix} \quad L : \text{reflux}, V : \text{boil-up}$$

$$\mathbf{K} = \begin{bmatrix} 0.07 & -0.05 \\ 0.1 & -0.15 \end{bmatrix}$$

$$\mathbf{\Lambda} = \begin{bmatrix} 1.91 & -0.91 \\ -0.91 & 1.91 \end{bmatrix}$$

Decouplers

$$D_{12} = -\frac{G_{12}}{G_{11}} = 0.714 \frac{12s+1}{15s+1} e^{+2s} \approx 0.71$$

$$D_{21} = -\frac{G_{21}}{G_{22}} = 0.667 \frac{10s+1}{11s+1} e^{-2s} \approx 0.67$$

Pair x_D - L , x_B - V : $\lambda = 1.91$ is not 1 but it is positive, and the reflux enters next to the distillate.

D_{12} would need a prediction (e^{+2s}) $\rightarrow$ drop it; D_{21} is realizable as it stands.

Example 6.6: PI tuning (ITAE)

ITAE set-point correlations for FOPDT loops, applied to the diagonal elements:

$$K_c = \frac{0.586}{K} \left(\frac{\theta}{\tau} \right)^{-0.916}, \quad \tau_I = \frac{\tau}{1.03 - 0.165 \theta / \tau}$$

$$K_{c1} = \frac{0.586}{0.07} \left(\frac{12}{3} \right)^{0.916} = 29.8$$

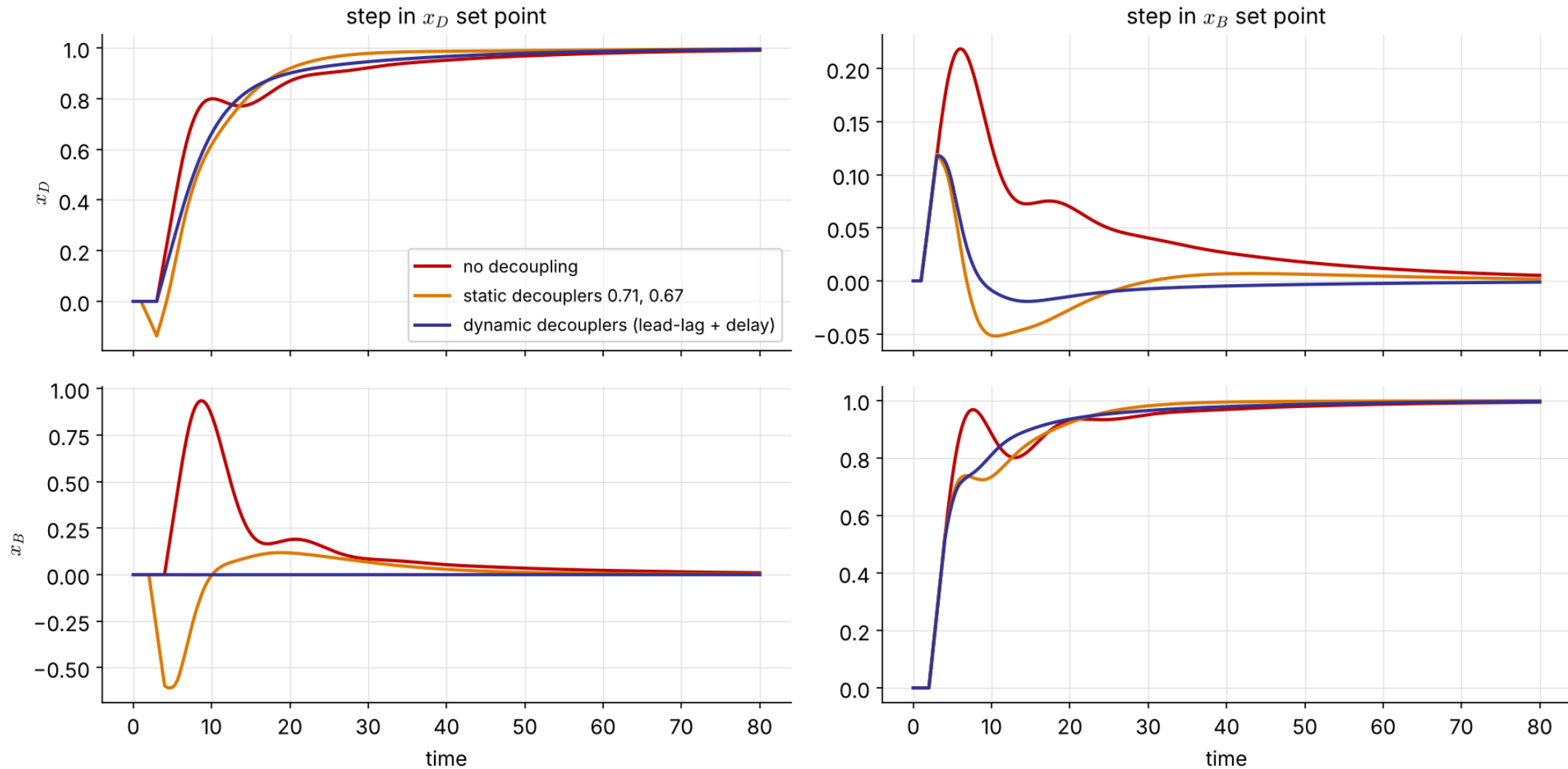
$$\tau_{I1} = \frac{12}{1.03 - 0.165(3/12)} = 12.1$$

$$K_{c2} = -\frac{0.586}{0.15} \left(\frac{10}{2} \right)^{0.916} = -17.1$$

$$\tau_{I2} = \frac{10}{1.03 - 0.165(2/10)} = 10.0$$

$K_{c2} < 0$ because $K_{22} < 0$: more boil-up lowers x_B .

Example 6.6: does decoupling help? (replaces the Simulink runs)



- **No decoupling:** an x_D set-point step drives x_B up to 0.94.
- **Static decouplers:** x_B now dips to -0.61 . D_{21} acts at $t = 2$ through G_{22} , before G_{21} arrives at $t = 4$.
- **Dynamic D_{21}** (with its e^{-2s}): x_B stays at 0, exact decoupling.

A decoupler that ignores the **timing** can trade one interaction for another. Keep realizable delays and lags.

Decoupling in code

Python

```

1 from blocks import simulate_2x2          # exact dead time; blocks.py in ch6/
2
3 G = [[(0.07, 12, 3), (-0.05, 15, 1)],    # (K, tau, theta) of each G_ij
4       [(0.10, 11, 4), (-0.15, 10, 2)]]
5 Kc = (29.8, -17.1); tI = (12.1, 10.0)
6
7 cases = {'none': (None, None),
8          'static': (0.714, 0.667),
9          'dynamic': ((0.714, 12, 15, 0), (0.667, 10, 11, 2))} # (K, tau_lead, tau_lag, theta)
10 for name, (D12, D21) in cases.items():
11     t, xD, xB, L, V = simulate_2x2(G, Kc, tI, D12, D21, r1=1, r2=0, tf=80)
12     print(name, 'max |x_B| =', abs(xB).max().round(2))      # 0.93, 0.61, 0.00

```

MATLAB

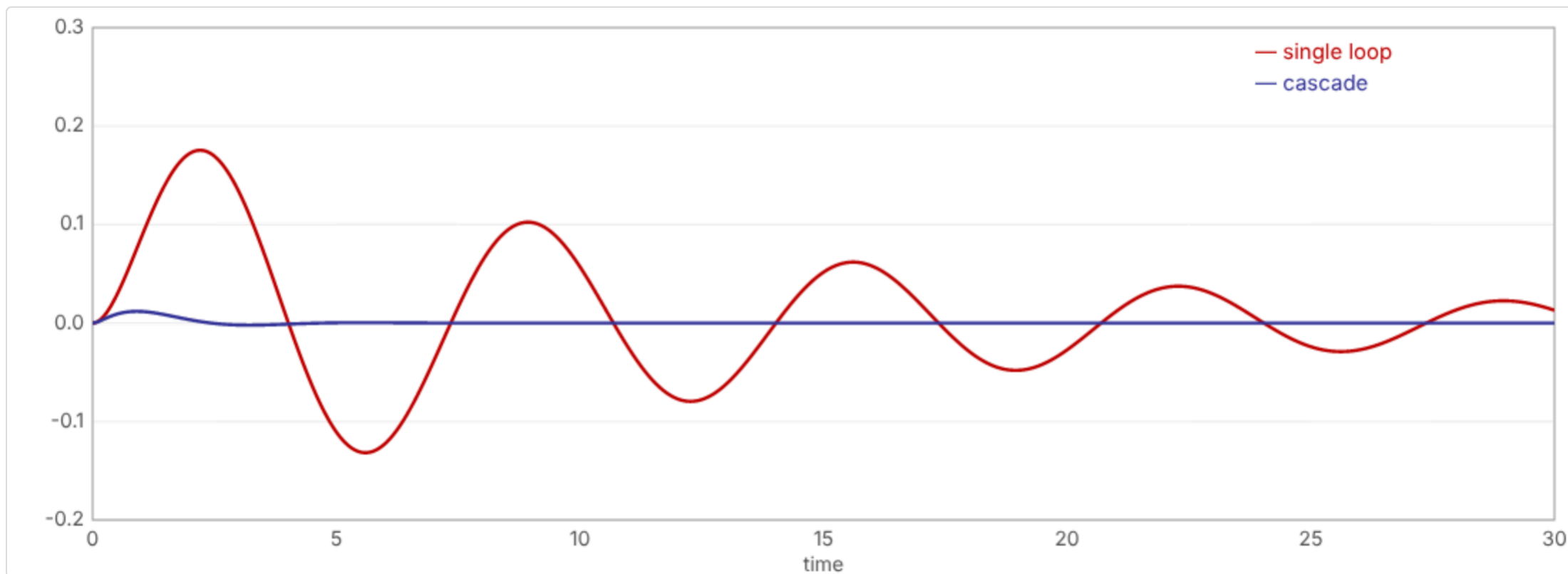
```

1 s = tf('s');
2 G = [0.07*exp(-3*s)/(12*s+1), -0.05*exp(-s)/(15*s+1);
3       0.10*exp(-4*s)/(11*s+1), -0.15*exp(-2*s)/(10*s+1)];
4 C = blkdiag(29.8*(1 + 1/(12.1*s)), -17.1*(1 + 1/(10.0*s)));
5
6 D_static = [1, 0.714; 0.667, 1];          % M = D * u
7 D_dynamic = [1, 0.714*(12*s+1)/(15*s+1);
8              0.667*(10*s+1)/(11*s+1)*exp(-2*s), 1];
9 for D = {eye(2), D_static, D_dynamic}
10     T = feedback(ss(G)*D{1}*C, eye(2));    % ss: mixed delays
11     step(T(:,1), 80), hold on              % step in r1 (x_D)
12 end

```

Interactive: cascade vs. single loop

primary $K_c = 3.0$ τ_I τ_I slave $K_{c2} = 18$ ☒ load step ☐ set-point step
 single loop: stable · peak 0.175 cascade: stable · peak 0.012 · slave $\tau_v^* = 0.100$, $K_v^* = 0.900$



Interactive version: seoultechpse.github.io/ProCon/ch6/ch6_multiloop.html — change K_c , τ_I and K_{c2} , and switch between load and set-point steps.

Summary: which structure for which problem?

Symptom	Structure	Needs
upsets enter through the manipulated stream	cascade	a secondary measurement much faster than the primary
a large disturbance can be measured	feedforward + feedback	a model of G_L/G_p (lead-lag is usually enough)
two streams must stay proportional	ratio	flow measurements
dead time dominates	Smith predictor	an accurate model, especially of θ
several loops interact	RGA pairing, then decoupling	gain matrix (pairing); dynamic model (decoupling)

None of these replaces feedback: each one is built **around** a feedback loop.

Problem #1 (homework)

$$\mathbf{G}(s) = \begin{bmatrix} \frac{2e^{-s}}{10s + 1} & \frac{1.5e^{-s}}{15s + 1} \\ \frac{1.5e^{-s}}{15s + 1} & \frac{2e^{-s}}{10s + 1} \end{bmatrix}$$

- Compute the RGA. Which pairing do you choose, and how strong is the interaction?
- Design the two decouplers. Are they realizable? Compare the static and dynamic versions.
- Tune two PI controllers with the ITAE correlations and simulate a step in R_1 with and without decoupling (use `simulate_2x2` from `blocks.py`).

Hint: $\kappa = \frac{K_{12}K_{21}}{K_{11}K_{22}}$.

Resources

- Slides (interactive HTML): seoultechpse.github.io/ProCon/ch6/ch6_multiloop.html
- PDF: [slides](#) · [handout with note space](#)
- Python notebook: [Open in Colab](#) — every calculation and figure of this chapter
- Simulator with exact dead time: [blocks.py](#)

Further reading:

- D. E. Seborg, T. F. Edgar, D. A. Mellichamp, *Process Dynamics and Control*
- P. C. Chau, *Process Control: A First Course with MATLAB*
- B. W. Bequette, *Process Control: Modeling, Design, and Simulation*
- C. D. Johnson, *Process Control Instrumentation Technology*
- S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*

A final word — Neal R. Amundson (University of Minnesota, circa 1970)

1. In an advanced course ... when has any substance there will be **no text which contains all of the material presented.**
2. Teachers who know their subject well will always present material not in the prescribed text when one is prescribed.
3. If there were a text which presented all of the materials there would be no need for the lecturer.
4. In a course of any substance it will not be possible for the student to understand or follow all of the arguments instantaneously all of the time.
5. Assuming that 4 is true it is essential that **good notes be taken.** Usually this is not possible. Normally the notes will be very sketchy and if they are to be of any use later, they must be expanded upon by the student as soon after the lecture as possible.
6. Because of 5 notes in real courses must be recopied, filled out, corrected for teacher aberrations, misprints, errors, etc., so that **the student ends up with his own text.**
7. Notes which have not been so treated are normally almost useless for later reference.
8. In core subjects one must learn and memorize and in this way make deposits in the bank from which he can withdraw later. If nothing is ever put in the bank then one is intellectually bankrupt. All knowledge is what one knows. **If one cannot verbalize what he knows then he properly does not understand what he thinks he knows.**
9. First-year graduate students, in general, must change their ways of studying from their undergraduate habits since in respectable graduate programs the emphasis is on understanding rather than getting through the course. This necessary shift is why many first-year graduate students go into shock after about six weeks of graduate work.
10. If a student aims toward a Ph.D. degree he is expected not only to understand but be able to improve by generalization or better by innovations. Good graduate programs require the latter.
11. **10 is what research is all about.**