FRTN10 Multivariable Control, Lecture 12

Anton Cervin

Automatic Control LTH, Lund University

Course Outline

L1-L5 Specifications, models and loop-shaping by hand

L6-L8 Limitations on achievable performance

L9-L11 Controller optimization: Analytic approach

L12-L14 Controller optimization: Numerical approach

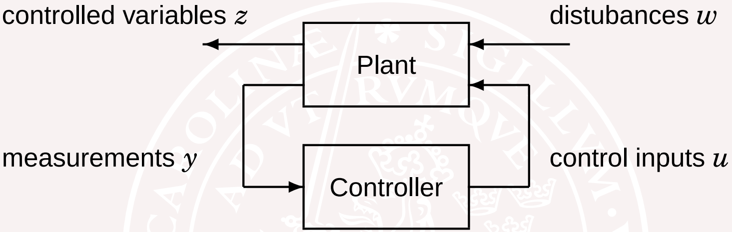
12. **Youla parameterization, Internal Model Control**
13. Synthesis by convex optimization
14. Controller simplification

Lecture 12

- The Youla Parameterization
- Internal Model Control
- Dead Time Compensation

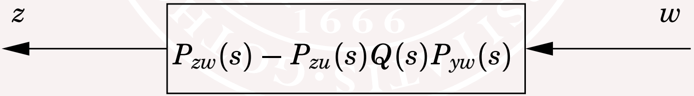
Section 8.4 in Glad/Ljung.

The Youla parameterization (Q parameterization)



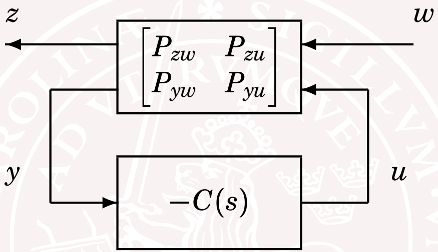
Idea for lectures 12–14:

The choice of controller corresponds to designing a transfer matrix $Q(s)$, to get desirable properties of the following map from w to z:



Once $Q(s)$ is determined, the corresponding controller can be found.

The Youla Parameterization



The closed loop transfer matrix from w to z is

$$G_{zw}(s) = P_{zw}(s) - P_{zu}(s)Q(s)P_{yw}(s)$$

where

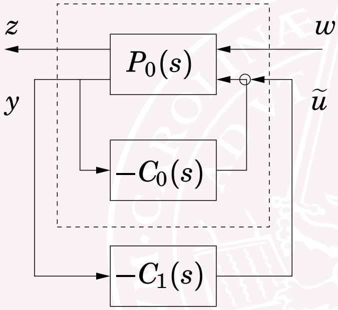
$$Q(s) = C(s)\left[I + P_{yu}(s)C(s)\right]^{-1}$$
$$C(s) = Q(s) + Q(s)P_{yu}(s)C(s)$$
$$C(s) = \left[I - Q(s)P_{yu}(s)\right]^{-1}Q(s)$$

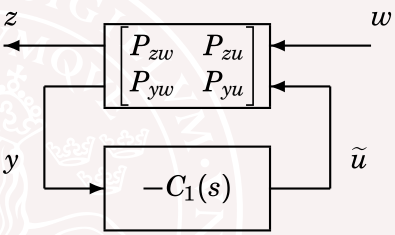
Closed loop maps for stable plants

Suppose the original plant P is stable. Then

- Stabilty of $Q(s)$ implies stability of $P_{zw}(s) - P_{zu}(s)Q(s)P_{yw}(s)$
- If $Q = C[I + P_{yu}C]^{-1}$ is unstable, then the closed loop is unstable.

Closed loop maps for unstable plants





If $P_0(s)$ is unstable, let $C_0(s)$ be some stabilizing controller. Then the previous argument can be applied with P_{zw}, P_{zu}, P_{yw}, and P_{yu} representing the stabilized closed-loop system.

Next lecture: Synthesis by convex optimization

A general control synthesis problem can be stated as a convex optimization problem in the variable $Q(s)$. The problem could have a quadratic objective, with linear/quadratic constraints, e.g.:

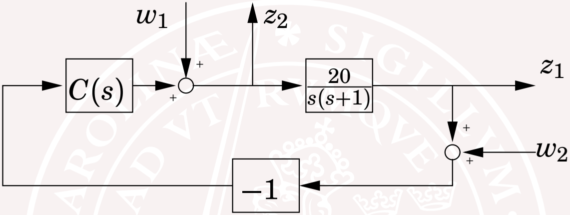
$$\text{Minimize} \quad \int_{-\infty}^{\infty} \Big|P_{zw}(i\omega) + P_{zu}(i\omega) \overbrace{\sum_k Q_k \phi_k(i\omega)}^{Q(i\omega)} P_{yw}(i\omega)\Big|^2 d\omega \quad \Big\} \text{ quadratic objective}$$

subject to
- step response $w_i \to z_j$ is smaller than f_{ijk} at time t_k
- step response $w_i \to z_j$ is bigger than g_{ijk} at time t_k

  } linear constraints
- Bode magnitude $w_i \to z_j$ is smaller than h_{ijk} at ω_k } quadratic constraints

Here $Q(s) = \sum_k Q_k \phi_k(s)$, where $\phi_1, \dots, \phi_m$ are some fixed “basis functions”, and $Q_0, \dots, Q_m$ are optimization variables. Once $Q(s)$ has been determined, the controller is obtained as
$C(s) = \left[I - Q(s)P_{yu}(s)\right]^{-1} Q(s)$

Example — DC-motor



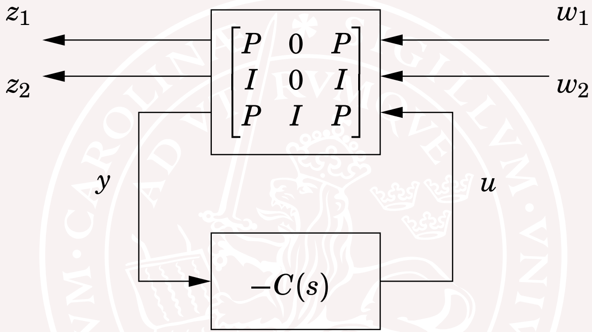
The transfer matrix from (w_1, w_2) to (z_1, z_2) is

$$G_{zw}(s) = \begin{bmatrix} \frac{P}{1+PC} & \frac{-PC}{1+PC} \\ \frac{1}{1+PC} & \frac{-C}{1+PC} \end{bmatrix}$$

where $P(s) = \frac{20}{s(s+1)}$. How to obtain stable P_{zw}, P_{zu}, P_{yw}, P_{yu} to get

$$G_{zw}(s) = P_{zw}(s) - P_{zu}(s)Q(s)P_{yw}(s) \quad ?$$

Stabilizing nominal feedback for DC-motor

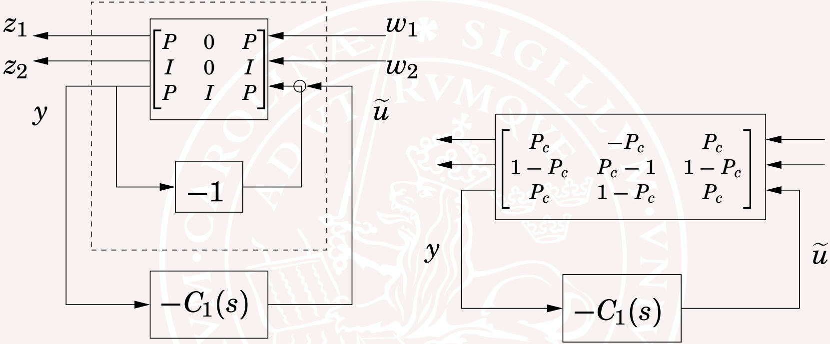


The plant $P(s) = \frac{20}{s(s+1)}$ is not stable, so write

$$C(s) = C_0(s) + C_1(s)$$

where $C_0(s) \equiv 1$ is a stabilizing controller.

Redraw diagram for DC motor example



$$G_{zw}(s) = \begin{bmatrix} P_c & -P_c \\ 1-P_c & P_c-1 \end{bmatrix} + \begin{bmatrix} P_c \\ 1-P_c \end{bmatrix} Q \begin{bmatrix} P_c & 1-P_c \end{bmatrix}$$

where $P_c(s) = (1+P(s))^{-1}P(s) = \frac{20}{s^2+s+20}$ is stable.

DC motor example – final controller

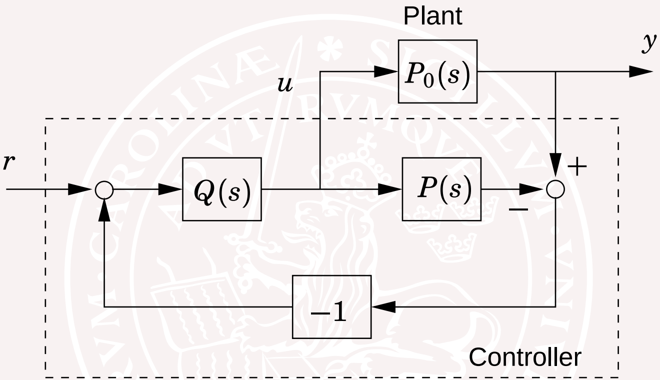
Once $Q(s)$ has been designed, the controller is obtained as

$$C_1 = (I - QP_c)^{-1}Q$$
$$C = C_0 + C_1$$

Outline

- Youla Parameterization
- **Internal Model Control**
- Dead Time Compensation

Internal Model Control

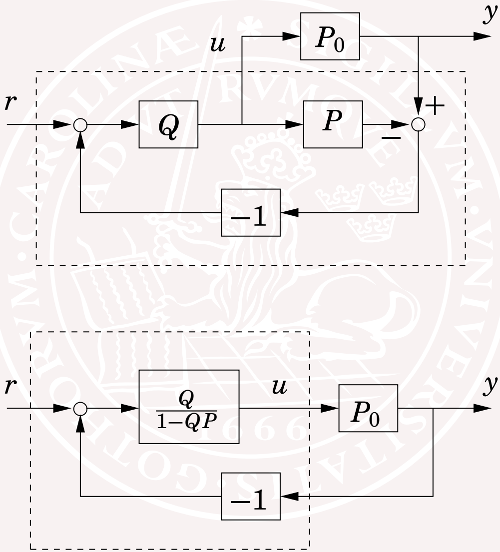


Feedback is used only as the real process deviates from $P(s)$.

The transfer function $Q(s)$ defines how the desired input depends on the reference signal.

When $P = P_0$, the transfer function from r to y is $P(s)Q(s)$.

Two equivalent diagrams



Internal Model Control — Strictly proper plants

When $P = P_0$, the transfer function from r to y is $P(s)Q(s)$.

For perfect reference following, one would like to put $Q(s) = P(s)^{-1}$. For several reasons this is not possible:

- If $P(s)$ is strictly proper, the inverse would have more zeros than poles. Instead, one could choose

$$Q(s) = \frac{1}{(\lambda s + 1)^n} P(s)^{-1}$$

where n is large enough to make Q proper. The parameter λ determines the speed of the closed-loop system.

Internal Model Control — Zeros and delays

Other reasons why $Q(s) = P(s)^{-1}$ is often not possible:

- If $P(s)$ has unstable zeros, the inverse would be unstable. Options:
  - Remove every unstable factor $(-\beta s + 1)$ from the plant numerator before inverting
  - Replace every unstable factor $(-\beta s + 1)$ with $(\beta s + 1)$. With this option, only the phase is modified, not the amplitude function.
- If $P(s)$ includes a time delay, its inverse would have to predict the future. Instead, the time delay is removed before inverting.

Design Example 1 — First order plant model

$$P(s) = \frac{1}{\tau s + 1}$$

$$Q(s) = \frac{1}{\lambda s + 1} P(s)^{-1} = \frac{\tau s + 1}{\lambda s + 1}$$

$$C(s) = \frac{Q(s)}{1 - Q(s)P(s)} = \frac{\frac{\tau s+1}{\lambda s+1}}{1 - \frac{1}{\lambda s+1}} = \underbrace{\frac{\tau}{\lambda}\left(1 + \frac{1}{s\tau}\right)}_{\text{PI controller}}$$

(This way of tuning a PI controller is known as *lambda tuning*)

Design Example 2 — Non-minimum phase plant

$$P(s) = \frac{-\beta s + 1}{\tau s + 1}$$

$$Q(s) = \frac{(-\beta s + 1)}{(\beta s + 1)} P(s)^{-1} = \frac{\tau s + 1}{\beta s + 1}$$

$$C(s) = \frac{Q(s)}{1 - Q(s)P(s)} = \frac{\frac{\tau s+1}{\beta s+1}}{1 - \frac{(-\beta s+1)}{(\beta s+1)}} = \underbrace{\frac{\tau}{2\beta}\left(1 + \frac{1}{s\tau}\right)}_{\text{PI controller}}$$

Outline

- ◦ Youla Parameterization
- ◦ Internal Model Control
- • **Dead Time Compensation**

Dead Time Compensation

Consider the plant model

$$P(s) = P_1(s)e^{-s\tau}$$

Let $C_0 = Q/(1 - QP_1)$ be the controller we would have used without delays. Then $Q = C_0/(1 + C_0 P_1)$.

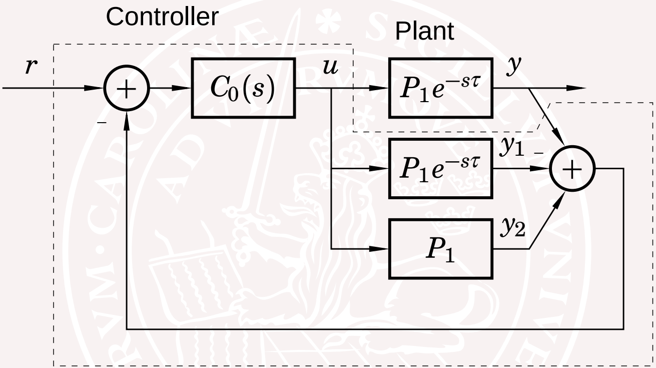
The rule of thumb tell us to use the same Q also for systems with delays. This gives

$$C(s) = \frac{Q(s)}{1 - Q(s)P_1(s)e^{-s\tau}} = \frac{C_0/(1 + C_0 P_1)}{1 - e^{-s\tau} P_1 C_0/(1 + C_0 P_1)}$$

$$C(s) = \frac{C_0(s)}{1 + (1 - e^{-s\tau})C_0(s)P_1(s)}$$

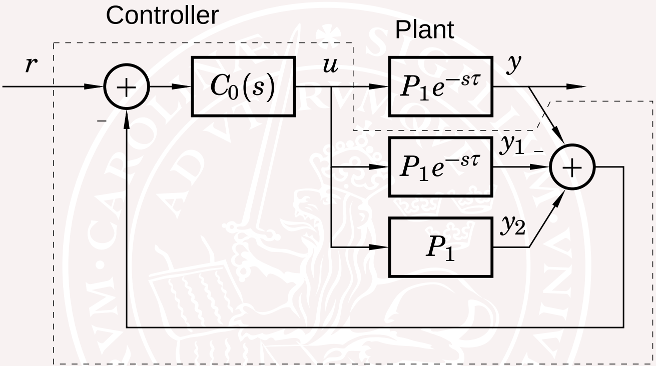
This modification of the $C_0(s)$ to account for time delays is known as a Smith predictor.

Smith Predictor



The Smith predictor uses an internal model of the process (with and without the delay). Ideally Y and Y_1 cancel each other and only feedback from Y_2 “without delay” is used.

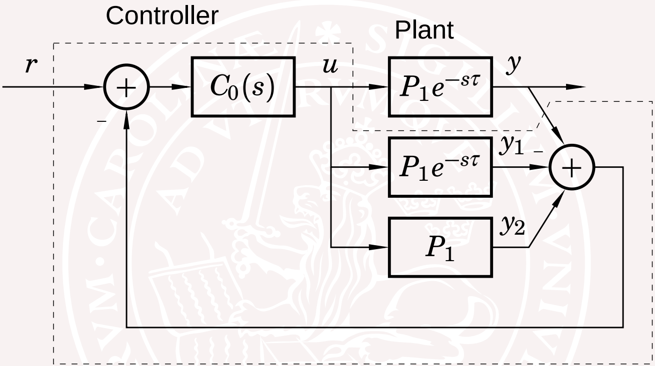
Smith Predictor



$$Y(s) = e^{-s\tau}\frac{C_0(s)P_1(s)}{1 + C_0(s)P_1(s)}R(s)$$

- Delay eliminated from denominator!
- Reference response greatly simplified!

Smith Predictor — A Success Story!

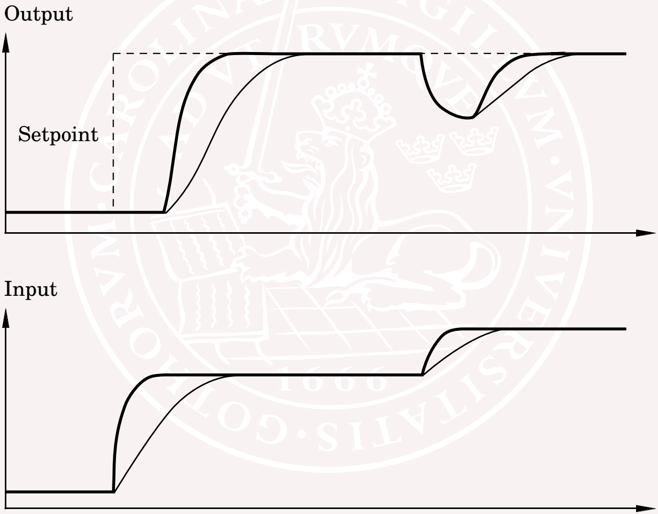


- Numerous modifications
- Many industrial applications

Otto J.M. Smith listed in the ISA "Leaders of the Pack" list (2003) as one of the 50 most influential innovators since 1774.

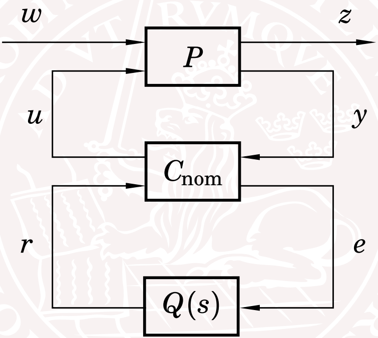
Example: Dead Time Compensation

Smith predictor (thick) and standard PI controller (thin)



Youla parameterization revisited

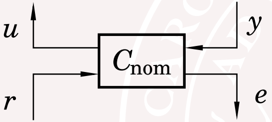
The Youla parameterization:

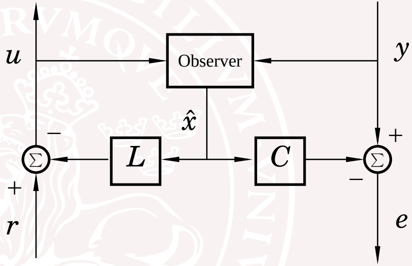


where C_{nom} stabilizes the $[P, C]$-system and $Q(s)$ is any stable transfer function.

Nominal Controller: State Feedback from Observer

Linear system $\dot{x} = Ax + Bu + B_w w$, $y = Cx + D_w w$





with observer

$$\dot{\hat{x}} = A\hat{x} + Bu + Ke$$
$$u = r - L\hat{x}$$
$$e = y - C\hat{x}$$

Summary

- $Q(s)$ can be designed by hand for simple plants
  - Internal Model Control
  - Warning: Cancellation of slow poles gives poor disturbance rejection
- $Q(s)$ can be found via convex optimization, also for multivariable plants (see Lecture 13)

Course Outline

- L1-L5 Specifications, models and loop-shaping by hand
- L6-L8 Limitations on achievable performance
- L9-L11 Controller optimization: Analytic approach
- L12-L14 Controller optimization: Numerical approach
  12. Youla parameterization, Internal Model Control
  13. Synthesis by convex optimization
  14. Controller simplification