FRTN10 Multivariable Control, Lecture 12

Automatic Control LTH, 2017

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

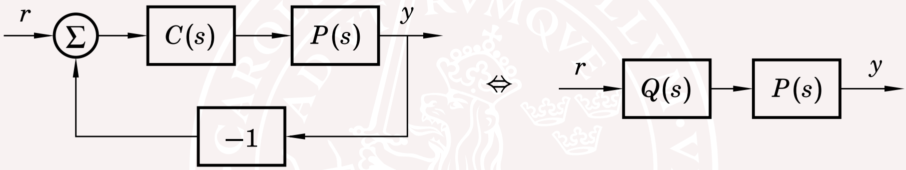
Lecture 12 – Outline

1. The Quola parameterization
2. Internal model control (IMC)

[Glad&Ljung Section 8.4]

Basic idea of Youla and IMC

Assume stable SISO plant P. Model for design:

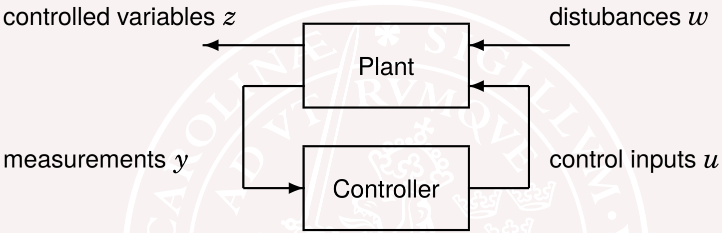


$$\frac{PC}{1+PC} = PQ$$

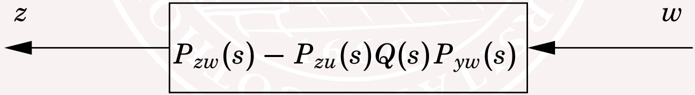
$$Q = \frac{C}{1+PC}$$

Design Q to get desired closed-loop properties. Then $C = \dfrac{Q}{1-QP}$

General idea for Lectures 12–14



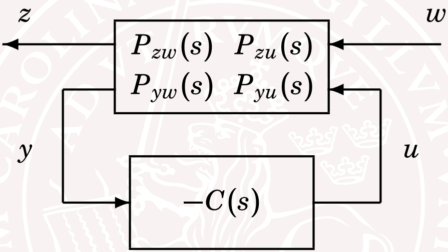
The choice of controller corresponds to designing a transfer matrix $Q(s)$, to get desirable properties of the following map from w to z:



Once $Q(s)$ has been designed, the corresponding controller can be found.

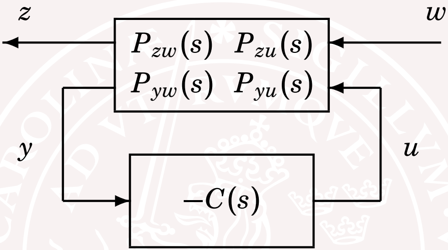
The Youla (Q) parameterization

General closed-loop control system:



$$Z(s) = P_{zw}(s)W(s) + P_{zu}(s)U(s)$$
$$Y(s) = P_{yw}(s)W(s) + P_{yu}(s)U(s)$$
$$U(s) = -C(s)Y(s)$$

The Youla (Q) parameterization



Closed-loop transfer function from w to z:

$$G_{zw}(s) = P_{zw}(s) - P_{zu}(s)\underbrace{C(s)\big[I + P_{yu}(s)C(s)\big]^{-1}}_{=Q(s)} P_{yw}(s)$$

Given $Q(s)$, the controller is $C(s) = \big[I - Q(s)P_{yu}(s)\big]^{-1}Q(s)$

All stabilizing controllers

Suppose the plant $P = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix}$ is stable. Then

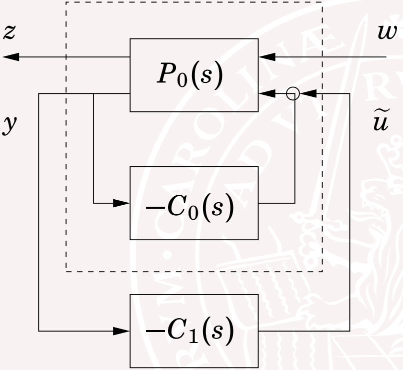
- Stabilty of Q implies stability of $P_{zw} - P_{zu}QP_{yw}$
- If $Q = C\big[I + P_{yu}C\big]^{-1}$ is unstable, then the closed loop is unstable.

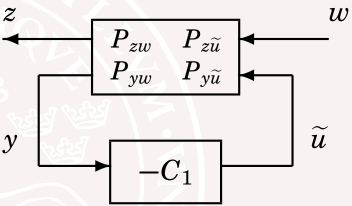
Hence, if P is stable then **all stabilizing controllers** are given by

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

where $Q(s)$ is an arbitrary stable transfer function.

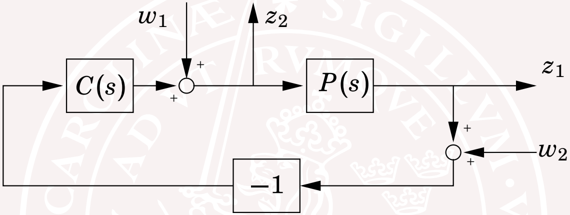
Dealing with unstable plants





If $P_0(s)$ is unstable, let $C_0(s)$ be some stabilizing controller. Then the previous argument can be applied with P_{zw}, $P_{z\tilde{u}}$, P_{yw}, and $P_{y\tilde{u}}$ representing the stabilized system.

Example – DC-motor



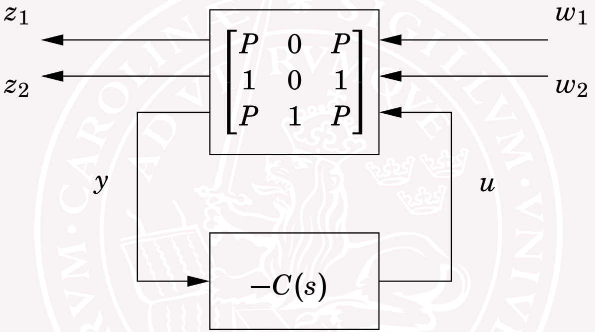
Assume we want to optimize the closed-loop transfer matrix from $(w_1, w_2)^T$ to $(z_1, z_2)^T$,

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

when $P(s) = \frac{20}{s(s+1)}$. How to parameterize all stabilizing controllers $C(s)$?

Stabilizing controller for DC-motor

Generalized plant model:

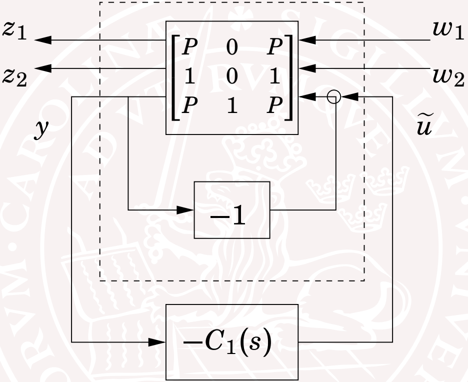


$P(s) = \frac{20}{s(s+1)}$ is not stable, so introduce

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

where $C_0(s) = 1$ stabilizes the plant; $P_c(s) = \frac{P(s)}{1+P(s)} = \frac{20}{s^2+s+20}$

Redrawn diagram for DC-motor example

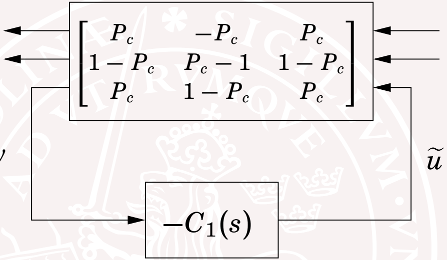


$$z_1 = Pw_1 + P(\widetilde{u} - y)$$

$$z_2 = w_1 + \widetilde{u} - y$$

$$y = Pw_1 + w_2 + P(\widetilde{u} - y) \quad \Rightarrow \quad y = \frac{P}{1+P}w_1 + \frac{1}{1+P}w_2 + \frac{P}{1+P}\widetilde{u}$$

Redrawn diagram for DC-motor example



$$P_{y\widetilde{u}} = P_c$$

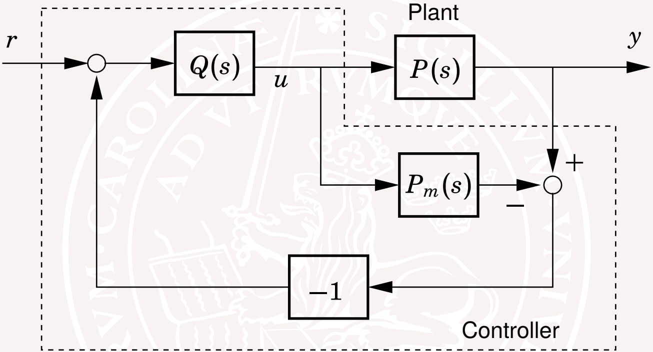
$$G_{zw} = \underbrace{\begin{bmatrix} P_c & -P_c \\ 1-P_c & P_c-1 \end{bmatrix}}_{P_{zw}} - \underbrace{\begin{bmatrix} P_c \\ 1-P_c \end{bmatrix}}_{P_{z\widetilde{u}}} Q \underbrace{\begin{bmatrix} P_c & 1-P_c \end{bmatrix}}_{P_{yw}}$$

Apply optimization (Lec. 13) to find $Q(s)$. Then $C(s) = 1 + \frac{Q(s)}{1-Q(s)P_{y\widetilde{u}}(s)}$

Lecture 12 – Outline

1. The Quola parameterization
2. Internal model control (IMC)

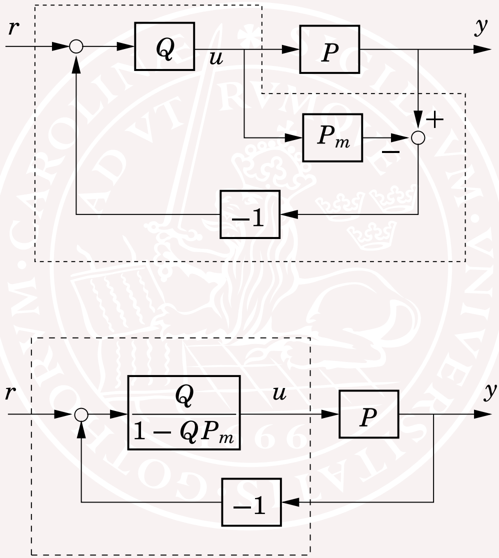
Internal model control (IMC)



Feedback is used only if the real plant $P(s)$ deviates from the model $P_m(s)$. $Q(s)$, $P(s)$, $P_m(s)$ must be stable.

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

Two equivalent diagrams



IMC design rules

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

For perfect reference following, one would like to have $Q(s) = P^{-1}(s)$, but that is not possible (why?)

Design rules:

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

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

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

  (cf. feedforward design in Lecture 4)

IMC design rules

- 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.

IMC design example 1 — first-order plant

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

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

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

Note that $T_i = \tau$

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

IMC design example 2 — non-minimum phase plant

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

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

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

Note that, again, $T_i = \tau$

The gain is adjusted in accordance with the fundamental limitation imposed by the RHP zero in $1/\beta$.

IMC design for dead-time processes

Consider the plant model

$$P_m = P_{0m} e^{-s\tau}$$

where the deadtime τ is assumed known and constant.

Let $C_0 = Q/(1 - QP_{0m})$ be a controller designed for the delay-free plant model P_{0m}. Then

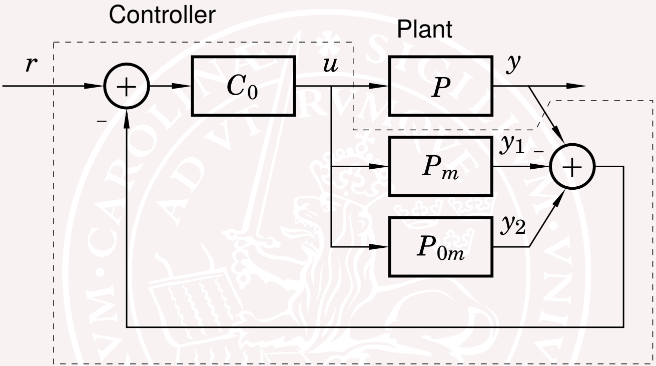
$$Q = \frac{C_0}{1 + C_0 P_{0m}}$$

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

$$C = \frac{Q}{1 - QP_{0m}e^{-s\tau}} = \frac{C_0}{1 + (1 - e^{-s\tau})C_0 P_{0m}}$$

This modification of C_0 to account for a time delay is known as a Smith predictor.

Smith predictor



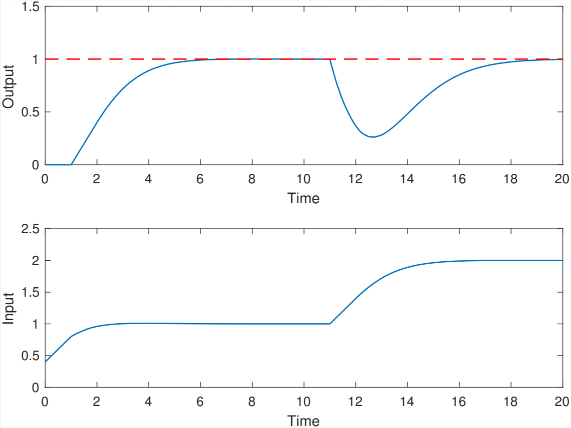
Ideally y and y_1 cancel each other and only feedback from y_2 “without delay” is used. If $P = P_m$ then

$$Y(s) = \frac{C_0(s)P_{0m}(s)}{1 + C_0(s)P_{0m}(s)} e^{-s\tau} R(s)$$

Example

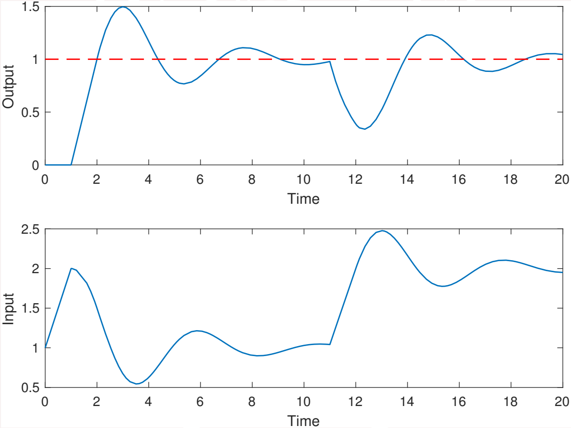
Plant: $P(s) = \dfrac{1}{s+1}e^{-s}$, nominal controller: $C_0(s) = K\left(1 + \dfrac{1}{s}\right)$

Simulation with $K = 0.4$, no Smith predictor:



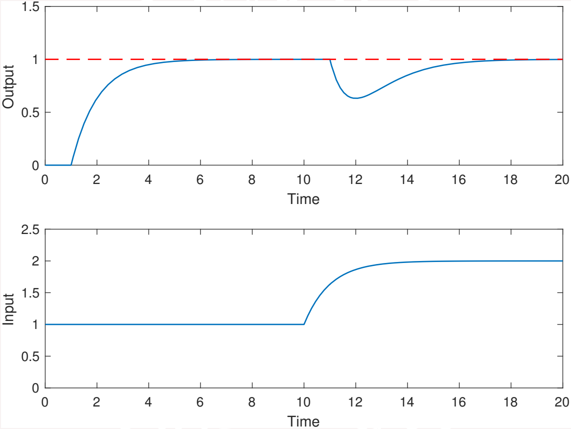
Example

Simulation with $K = 1$, no Smith predictor:



Example

Simulation with $K = 1$ with Smith predictor:



(But do not the forget the fundamental limitation imposed by the time delay!)

Lecture 12 – summary

- Idea: Parameterize the closed loop as

$$G_{yr} = PQ$$ SISO case, for IMC design

or

$$G_{zw} = P_{zw} - P_{zu} Q P_{yw}$$ General MIMO case, suitable for optimization

for some stable Q.

- After designing Q, the controller is given by

$$C = \frac{Q}{1 - QP}$$ SISO case

or

$$C = \left[I - Q P_{yu} \right]^{-1} Q$$ General MIMO case