FRTN10 Multivariable Control, Lecture 12

Automatic Control LTH, 2016

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 Youla parameterization
2. Internal model control (IMC)
3. Dead-time compensation

[Glad&Ljung Section 8.4]

Lecture 12 – Outline

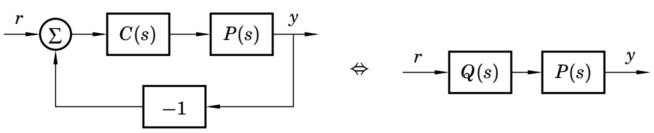
The Quola parameterization

Internal model control (IMC)

Dead-time compensation

Basic idea of Youla and IMC

Assume stable plant P. Model for design:

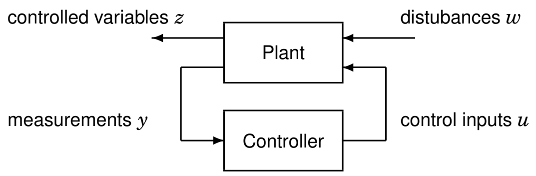


$$\frac{PC}{1+PC} = PQ$$

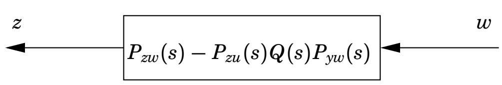
$$Q = \frac{C}{1+PC}$$

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

General idea for Lectures 12–14

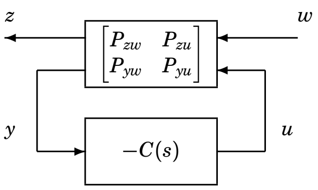


The choice of controller corresponds to designing a transfer matrix $Q(s)$, to get desirable properties of the following map from w to z:



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

The Youla (Q) parameterization



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

The controller is given by $C(s) = \left[I - Q(s)P_{yu}(s)\right]^{-1}Q(s)$

Stability

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

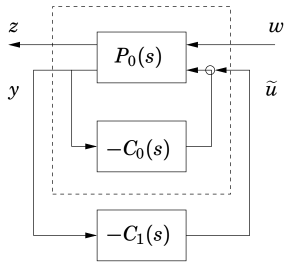
- Stabilty of Q implies stability of $P_{zw} - P_{zu}QP_{yw}$
- If $Q = C\left[I + P_{yu}C\right]^{-1}$ is unstable, then the closed loop is unstable.

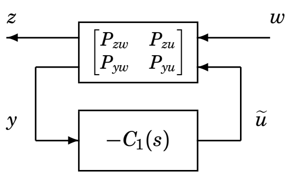
Hence, **all stabilizing controllers** are given by

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

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_{zu}, P_{yw}, and P_{yu} representing the stabilized system.

Next lecture: Synthesis by convex optimization

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

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

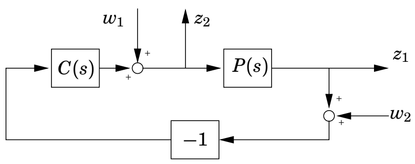
subj. to step response $w_i \to z_j$ is smaller than f_{ijk} at time t_k
step response $w_i \to z_j$ is bigger than g_{ijk} at time t_k } linear constraints
Bode magnitude $w_i \to z_j$ is smaller than h_{ijk} at ω_k } quadratic constraints

Here $Q(s) = \sum_k Q_k\phi_k(s)$, where $\phi_1, \ldots, \phi_m$ are some fixed basis functions, and $Q_0, \ldots, Q_m$ 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



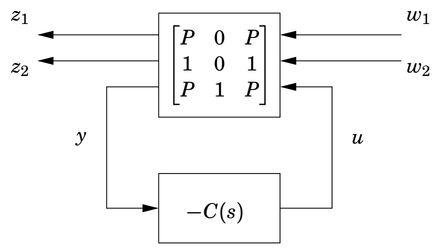
Assume we want to optimize the closed-loop transfer matrix from (w_1, w_2) to (z_1, z_2),

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

when $P(s) = \frac{20}{s(s+1)}$. How to 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 controller for DC-motor

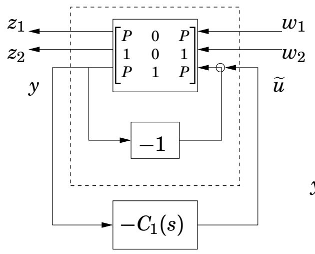


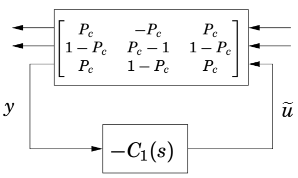
The plant $P(s) = \frac{20}{s(s+1)}$ is not stable, so introduce

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

where $C_0(s) = 1$ stabilizes the plant

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

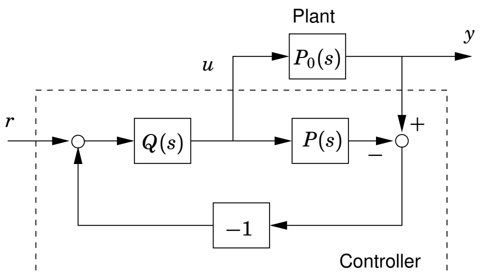
Lecture 12 – Outline

The Quola parameterization

Internal model control (IMC)

Dead-time compensation

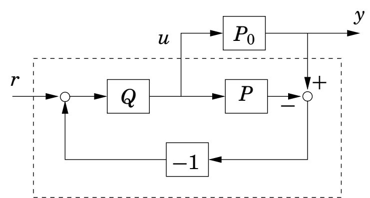
Internal model control (IMC)

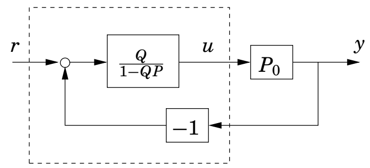


Feedback is used only if the real process $P_0(s)$ deviates from the model $P(s)$.

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

Two equivalent diagrams





IMC design rules

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 $Q(s) = P(s)^{-1}$. That is impossible for several reasons.

Practical design rules:

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

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

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

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 $1/\beta$.

Lecture 12 – Outline

The Quola parameterization

Internal model control (IMC)

Dead-time compensation

IMC design for dead-time processes

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 the delay. Then $Q = C_0/(1 + C_0P_1)$.

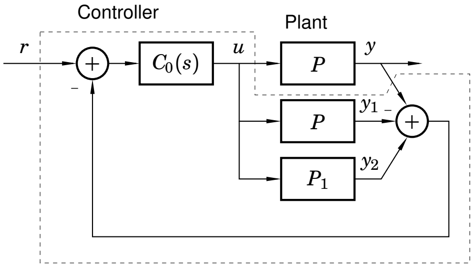
The rule of thumb tell us to use the same Q also for systems with delays. This gives

$$C(s) = \frac{Q(s)}{1 - Q(s)P_1(s)e^{-s\tau}} = \frac{C_0/(1 + C_0P_1)}{1 - e^{-s\tau}P_1C_0/(1 + C_0P_1)}$$

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

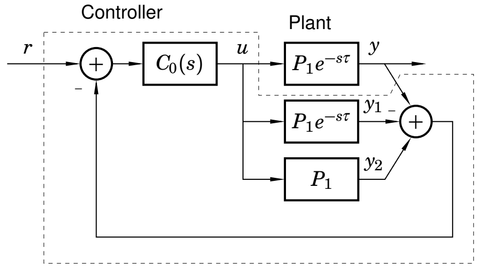
This modification of the $C_0(s)$ to account for time delays is known as a Smith predictor.

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.

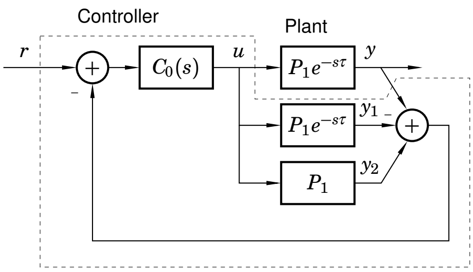
Smith predictor



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

- Delay eliminated from denominator
- Closed-loop response greatly simplified

Smith predictor – a success story!

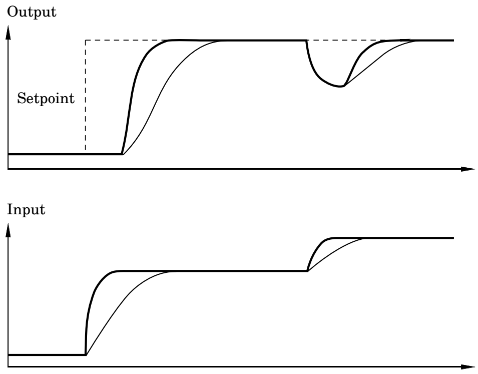


- Numerous modifications
- Many industrial applications

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)



Summary

- $Q(s)$ can be designed by hand for simple plants
  - Internal model control
  - Warning: Cancellation of slow poles can give poor disturbance rejection
- $Q(s)$ can be found via convex optimization, also for multivariable plants – see next lecture