Course 28451: Optimizing Plantwide Operation

Exercise 4B: Dynamics & control of bioreactor: Loop pairing and tuning

This exercise deals with two important topics in control design: Given a certain number of possible manipulated and controlled variables, how to decide the loop pairing (i.e. which variable j is manipulated in order to control a variable i), and, once the pairing is established, how to tune a PID controller with information from the model. You will use the continuous fermentation bioreactor presented and linearised in exercise 4A and you will select the variable pairing based on the relative gain array (RGA). You will test loop pairing and tuning by simulating the bioreactor in Matlab/Simulink

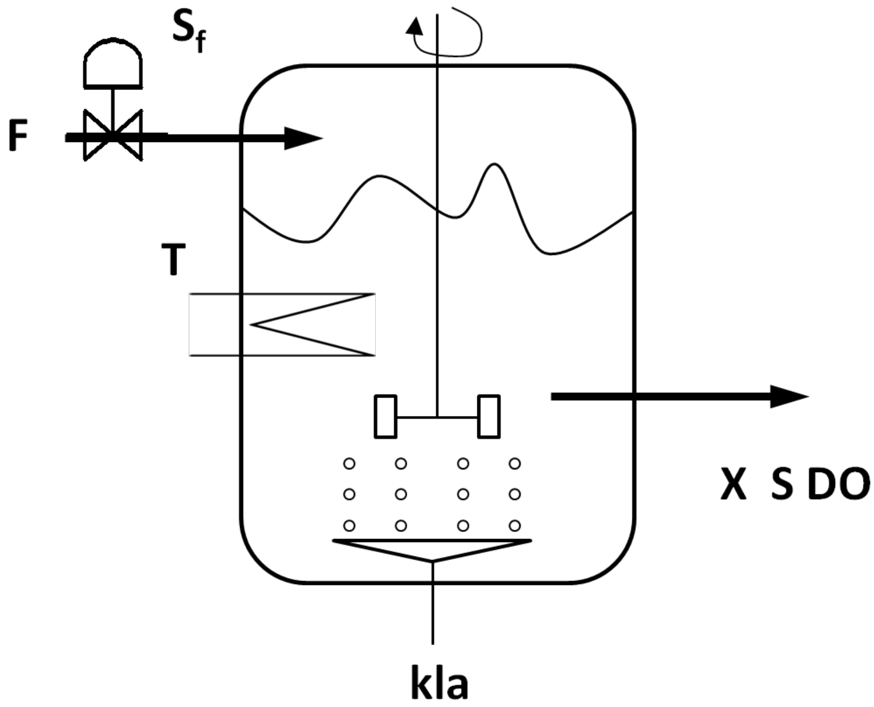

Figure 1: Continuous fermentation bioreactor (details in ex. 3A)

Determination of the relative gain array (RGA)

The relative gain array (see its definition and properties in Seborg's book, Chapter 18, 2^{ed}) provides a useful criterion for variable pairing. In order to determine the RGA, first we must compute the steady-state gain matrix of the plant. In order to do so, let's examine the continuous state-space formulation of the plant at **steady-state**:

$$0 = AX + BU$$

$$Y = CX + DU$$

and the definition of the steady state gain matrix:

$$Y = KU$$

By simple manipulation of the equations above, it becomes clear that:

$$K = -CA^{-1}B + D$$

Then the RGA (here denoted Λ) is easily calculated as:

$$\Lambda = K \otimes H$$

where $\otimes$ denotes the Schur product, element by element multiplication:

$$\lambda_{ij} = K_{ij} H_{ij}$$

and where H is defined as $(K^{-1})^T$; the transpose of the matrix inverse of K.

Matlab hint 1*. If the plant transfer function (G) is available, you can also determine the steady state gain matrix evaluating G at steady state (s=0). This can be easily done in Matlab as:*

```
G=tf(lsys);
K=freqresp(G,0); % Frequency response at s=0
```

Loop tuning

Proportional Integral Derivative (PID) controllers are by the most widely used type of feedback control. Therefore, a large number of methods[1] exist to tune the controllers (i.e. to set the values of the proportional gain, integral time and derivative time). Here, we propose to use three PI controllers. They will be tuned with the internal mode control (IMC) based settings after approximating the transfer function to a first-order process.

Matlab hint 2*. In order to approximate the transfer functions to a form that appears in the IMC table 1, use the command* minreal *(minimal realization or pole-zero cancelation) but be careful that the gain is estimated correct.*

Table 1. IMC settings for first-order transfer function

Model	K_cK	τ_I	τ_D
$\frac{K}{\tau s + 1}$	$\frac{\tau}{\tau_C}$	τ	-

[1] For an exhaustive relationship of tuning methods you can check:
- O'Dwyer, A. *Handbook of Controller Tuning Rules*, Imperial college Press, London 2003
- Åström, K. J. and Hägglund, T. *PID Controllers: Theory, Design and Tuning*, 2nd ed. ISA, Research Triangle Park, NC

Hint 3. High order functions can also be approximated to low order functions (appearing in the IMC table) using Skogestad's half rule[2] (Seborg's book, Chapter 6.3, 2nd ed)

About the PID tuner in Matlab/Simulink

The PID block in Simulink can be tuned automatically. By default, the algorithm chooses a crossover frequency (loop bandwidth) based upon the plant dynamics, and designs for a target phase margin of 60°. If you specify the crossover frequency (how fast you want the controller to act) or phase margin (how robust you want the controller to be) using the algorithm computes PID gains that best meet those targets.

A phase margin of 60° is indeed enough to ensure stability but this is only true for SISO systems and only when the model matches all the components of the plant, i.e. when sensors and actuators are also modeled or when we can ensure that the time response and delay associated with sensors and actuators are much shorter than the process.

We do not recommend to use the PID tuner without understanding, at least to a certain extent, the methods used thereof. A good introduction appears in Seborg's book (Chapter 14, 2nd ed or 3rd ed)

In this exercise, the following is asked:

a) Using the linearised model obtained in exercise 4A, calculate the state-space matrices *A, B, C, D*. Comment on the values of the output matrix *C* and the feedthrough matrix *D*
b) Determine the steady state gain matrix K. Examining the values of K, could you have any hint of what loop pairing would be advisable?
c) Determine the RGA. Based on it, what loop pairing would you choose? Why? Is there more than one recommended pairing?
d) Approximating the transfer functions to a low order process e.g. first order (for example through zero-pole cancellation using the *minreal* command or the half rule by Skogestad), calculate the settings for the PI controllers based on the IMC recommendations. What would be the settings for a conservative tuning? And for a tighter tuning?
e) Given the following constraints on :
   F: $0 \le F \le 1$ L/h T: $293.5 \le T \le 303.5$ K
   Test the pairing and tuning estimated by simulating the process starting from steady state and check its response to
   i) keeping the process at steady state.
   ii) rejecting an external disturbance (-10% step in the feed concentration).
   iii) tracking a set point change (+25% step in the DO set point).

This exercise forms part B of exercise 4. A written report on the combined results of exercise 4 part A and B with interpretation and discussion is to be handed in on DTU Learn as a single pdf file.

The report should contain a problem statement (you may rephrase the exercise), the results and interpretation / discussion of the results. The following is suggested for the documentation:

---

[2] Skogestad, S. (2003) J. Process Control. 13, 291-309

- A few representative plots (with appropriate titles and labels), in particular to illustrate question e) on disturbance rejection and setpoint tracking.
- A description of the solution procedure.
- Answers to the specific questions in the problem.