SE4

Prof. Davide Manca – Politecnico di Milano

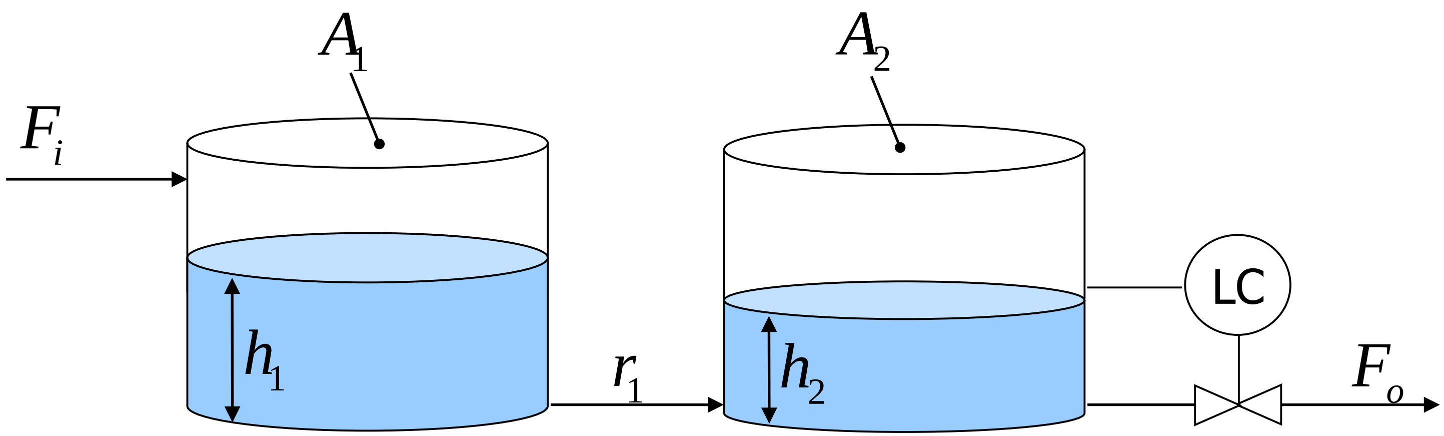
Dynamics and Control of Chemical Processes

Solution to Lab #4

Design of a control system

POLITECNICO
MILANO 1863

System representation



Data:

$$F_i = 9.4 \ \mathrm{m^3/s}$$

$$A_1 = 30 \ \mathrm{m^2}$$

$$A_2 = 50 \ \mathrm{m^2}$$

$$r_1 = 1.2 \ \mathrm{s/m^2}$$

I.C.:

$$h_2 = 6.6 \, \mathrm{m}$$

New set-point

$$h_2 = 8.6 \, \mathrm{m}$$

Design of the proportional-integral controller

Performance criteria

- Integral of Square Error (ISE):

$$ISE = \int_0^{+\infty} \varepsilon^2(t)\,dt \quad \Rightarrow \quad \underset{K_C,\tau_I}{Min}(ISE) = \underset{K_C,\tau_I}{Min}\left(\int_0^{+\infty} \varepsilon^2(t)\,dt\right)$$

- Integral of the Absolute value of Error (IAE):

$$IAE = \int_0^{+\infty} \left|\varepsilon(t)\right| dt \quad \Rightarrow \quad \underset{K_C,\tau_I}{Min}(IAE) = \underset{K_C,\tau_I}{Min}\left(\int_0^{+\infty} \left|\varepsilon(t)\right| dt\right)$$

- Integral of the Time-weighted Absolute Error (ITAE):

$$ITAE = \int_0^{+\infty} t\left|\varepsilon(t)\right| dt \quad \Rightarrow \quad \underset{K_C,\tau_I}{Min}(ITAE) = \underset{K_C,\tau_I}{Min}\left(\int_0^{+\infty} t\left|\varepsilon(t)\right| dt\right)$$

- Where: $\varepsilon(t) = y_{SP}(t) - y(t)$

Design of the proportional-integral controller

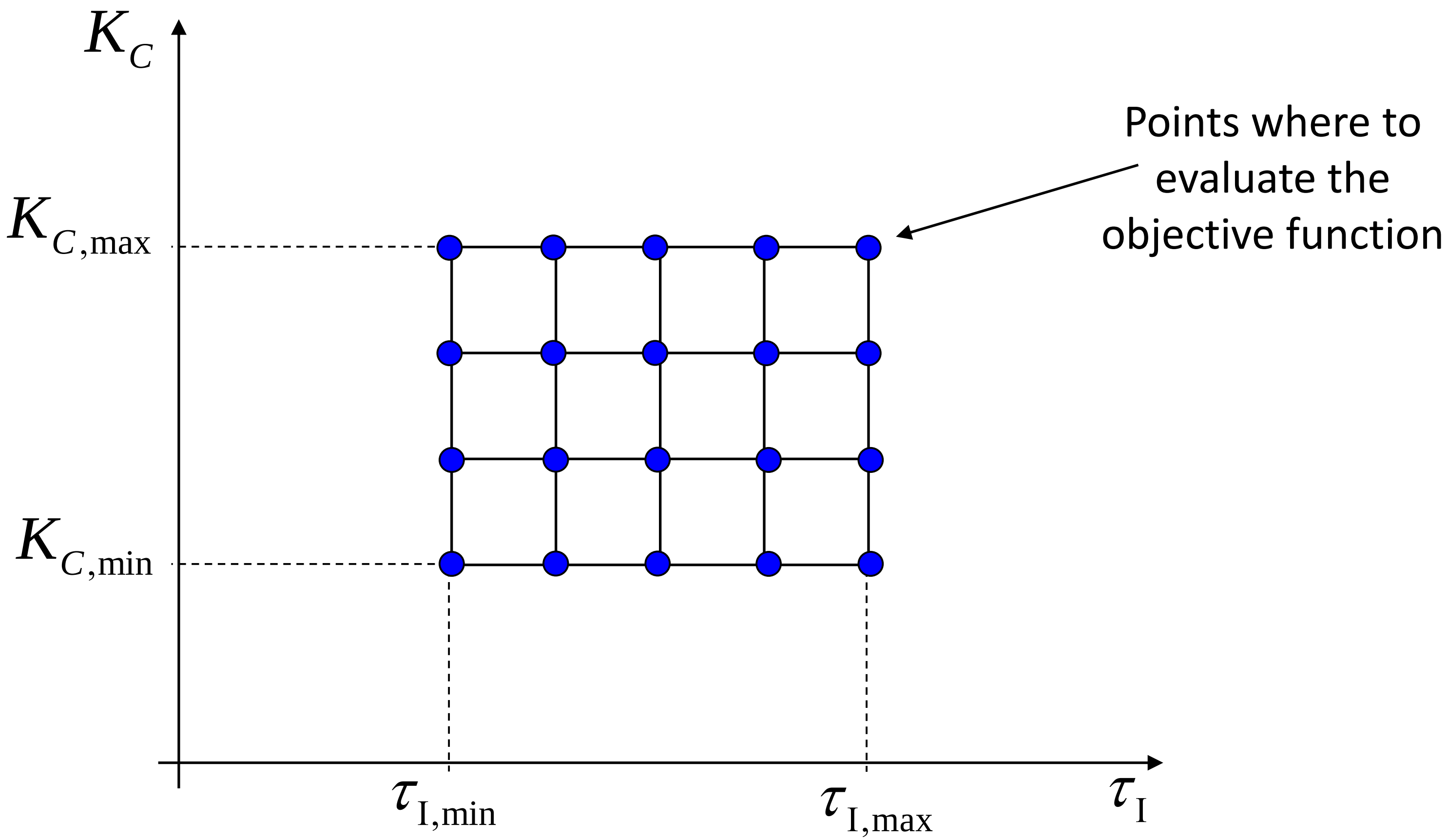
Pick controller parameters to minimize integral:

ISE Longer settling time

IAE Allows larger deviation than ISE

ITAE Weights errors occurring later more heavily

Minimum search: brute-force method



Optimizer use

$$K_C^{(n)} \quad , \quad \tau_I^{(n)}$$

Evaluation of the dynamics

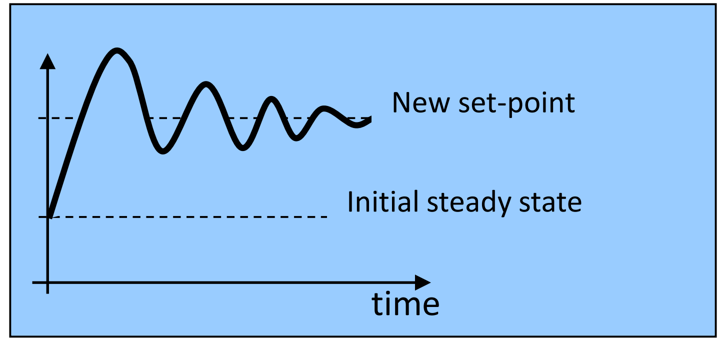
Objective function evaluation

Evaluation of the new controller parameters

$$K_C^{(n+1)} \quad , \quad \tau_I^{(n+1)}$$

$$\left|K_C^{(n)} - K_C^{(n-1)}\right| \le \varepsilon_{K_C} \quad , \quad \left|\tau_I^{(n)} - \tau_I^{(n-1)}\right| \le \varepsilon_{\tau_I}$$

$$K_C^{(opt)} \quad , \quad \tau_I^{(opt)}$$



$$\underset{K_C,\tau_I}{Min}\left(ISE\right) = \underset{K_C,\tau_I}{Min}\left(\int_0^{+\infty} \varepsilon^2\left(t\right)dt\right)$$

MATLAB optimizer

A constrained optimization is performed in order to avoid that the optimizer chooses unfeasible values of the variables respect to which the optimization is being done:

```
xmin = [min(kc) min(taui)];        % Lower limits
xMax = [max(kc) max(taui)];        % Upper limits
x0 = [kc0 taui0];           % Initial values

[x] = fmincon(@(x)funct(x,params),x0,[],[],[],[],xmin,xmax);
Kc = x(1);
taui = x(2);

end
```

MATLAB implementation

```
function ISE=funct(x)
   ti=x(2);
   Kc=x(1);
global  fold integral told SP
   ...                                  %Definition of constraints
   SP=6.6;                              %Set point
   h10=F*r1+6.6;                        %Steady state conditions (See Lab 3)
   h20=6.6;                             %h2 initial value
   y0=[h10 h20];                        %Vector of initial values
   integral=0;                          %definition of the integral
   fold=h20;                            %trapezoid base
   told=0;                              %trapezoid leg

option = odeset('RelTol',1E-8,'AbsTol',1E-10,'OutputFcn',@printo);
   [time,height]=ode23s(@(t,y)tank_LC(t,y,F,A1,A2,r1, Kc, ti),[0 600],y0,option);
```

MATLAB implementation

```
err=(height(:,2)-SP);          %%Error definition
err2=err.^2;                   %%Squared error (ISE)
stepm=length(time);            %Number of steps
    I=0;                       %Initialization of the integral

for m=2:stepm
       %Calculation of the integral (Trapezoid method)
        I= I+(err2(m)+err2(m-1))*(time(m)-time(m-1))/2.;
end
ISE=I;
end
```

MATLAB implementation

```
function dy=tank_LC(t,y,F,A1,A2,r1, Kc, ti)
    global  fold integral told SP
    h1=y(1);
    h2=y(2);

    if t>=20
        Fin=F*2;        %Disturbance: it doubles the inlet flow
    else
        Fin=F;
    end
    integraltime= integral + (fold-SP + h2-SP)*(t-told)/2;
    F1=(h1-h2)/r1;                          %Inlet flow, from the stady state assumption
    F2=F+Kc*(h2-SP)+Kc/ti*integraltime;     %Outlet flow, controlled variable
    dy(1,:)=(Fin-F1)/A1;                    %Governing equations
    dy(2,:)=(F1-F2)/A2;

end
```

MATLAB Output function

```
function status = printo(t,y,flag)       %Function printo
   global  told fold SP integral

   if strcmp(flag,'init')

   elseif strcmp(flag,'done')

   else
       h2=y(2);
       err=h2-SP;
       integral=integral+(fold-SP+err)*(t-told)/2;
       fold=h2;
       told=t;
   end
   status=0;
   end
```

Bibliography

- Stephanopoulos G. (1984) ‘Chemical Process Control. An Introduction to Theory and Practice’, Prentice-Hall, Englewood Cliff.
- Luyben, W., Tyréus, B., Luyben, M. (1998) ‘Plantwide Process Control’, McGraw-Hill, New York.
- Forzatti, P., Lietti, L. (2000) ‘Strumentazione industrial chimica. Elimenti di regolazione’, CUSI, Milan.