Process automation – Written exercises

There are two main control strategies: feedback control and feedforward.

Feedback control the controlled variables are the measured ones and the measurements are used to adjust the controlled variables. The disturbances are not measured because the system reacts to the effect of the disturbance and not on the cause. The advantages are that the corrective actions are taken regardless the cause of the disturbance and it is used to reach the set point. The disadvantage is that we may not able to compute the corrective action until the effect of the disturbance is seen on the process.

Feedforward control the measured variables are the disturbance and not the controlled ones. The advantage is that corrective actions are taken before the effect of the disturbance appears (we can also cancel the effect of a disturbance on the controlled variable). the disadvantage is that there may be many disturbances and it is not economical to measure them all; we need to understand which are the one really affecting the process (so we need a process model).

Time delay systems

We will consider processes with delay that can be written as:

$$P(s) = P_0(s)e^{-\theta s}$$

Where P_0 is the delay free function and θ is the delay, we will consider that it is positive.

The open loop transfer function is

$$F(s) = G(s)P(s) = G(s)P_0(s)e^{-\theta s} = F_0 e^{-\theta s}$$

The module of $F(s)$ is the same of $F_0(s)$ since the module of the exponential is 1. Anyway, concerning the phase, we have:

$$\measuredangle\{F(j\omega)\} = \measuredangle\{F_0(j\omega)e^{-\theta j\omega}\} = \measuredangle\{F_0(j\omega)\} - \omega\theta, \forall\omega$$

So the presence of the delay may lead the system to instabilities.

Gain margin → maximum gain variation the system can stand before becoming unstable:

$$m_g = \left|\frac{1}{F(j\tilde{\omega})}\right|_{db} = -20\log_{10} F(j\tilde{\omega}) \qquad \tilde{\omega}\; st \arg F(j\tilde{\omega}) = -180°$$

Phase margin → maximum phase lag the system can stand before becoming unstable:

$$m_\psi = \arg F(j\omega_c) - (-180°) \qquad \omega_c\; st\; |F(j\omega_c)| = 1$$

Delay margin → maximum delay the system can stand before becoming unstable:

$$\theta_{max} = \frac{m_\psi}{\omega_c}$$

If the system has already a delay: $m_\tau = \frac{m_\psi}{\omega_c} - \theta > 0$

If the delay is small ($\theta < 0.1\tau$) so it one order of magnitude less smaller with respect to the time constant of the system, we can approximate it with a polynomial function using the **Padè approximation**

$$e^{-\theta s} = \frac{\left(1 - \frac{\theta}{2}s\right)}{1 + \frac{\theta}{2}s}$$

Or 2-2 Padè approximation:

$$e^{-\theta s} = \frac{1 - \frac{\theta}{2}s + \frac{\theta^2}{12}s^2}{1 + \frac{\theta}{2}s + \frac{\theta^2}{12}s^2}$$

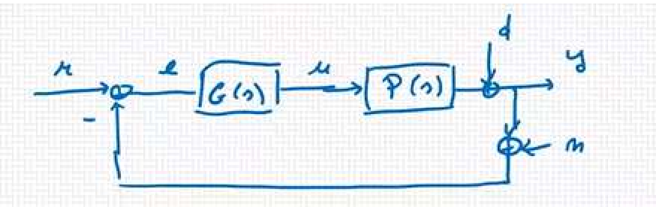
We have three step procedure when we deal with delays:

1. Use Padè approximation: $e^{-\theta s} \approx G_1(s)$
2. Design a controller $C(s)$ for the process $P_0(s)G_1(s)$ (we have substituted the exponential with the approximation in the process)
3. Verify that the controller $C(s)$ stabilized the real process since we have designed it using an approximation. That is, check that $m_{\tau_0} > 0$ with m_{τ_0}delay margin of $C(s)P_0(s)$.

Internal model control

We have a nominal model that computes the nominal output and the error is defined as the difference between the measured and the nominal output. In classical control scheme the error is defined as the difference between the reference and the measured output.

Classical control



$$y(s) = \frac{P(s)G(s)}{1 + P(s)G(s)}(r - n) + \frac{1}{1 + P(s)G(s)}d$$

$$y(s) = T(s)(r - n) + S(s)d$$

With $T(s) \rightarrow complementary\ trasfer\ function, S(s) \rightarrow sensitivity\ trasfer\ function$

$$T(s) = 1 - S(s)$$

Sensitivity transfer function It represents the robustness characteristic of the system; we would like it to be 0 so to not have the effect of the disturbance on the output. It is not possible to have it for all the frequency, we can have it just from the frequency of interest (ω_1, ω_2):

$$S(j\omega) \approx 0\ \forall\omega\ (\omega_1, \omega_2)$$

Complementary transfer function It represents the accuracy of the regulation, we would like it to be 1 so to have the output equal to the reference. This is not possible for all the frequency so we can have it just from the frequency of interest (ω_1, ω_2).

$$T(j\omega) \approx 1\ \forall\omega\ (\omega_1, \omega_2)$$

Anyway, it is also the transfer function between the noise and the input. The noise usually appears at high frequency $(\omega > \omega_n)$; in order to not have the effect of the noise on the output we would like:

$$T(j\omega) \approx 0 \quad \forall\omega > \omega_n$$

Equivalent controller The IMC controller can be implemented also with a classical controller. Let $Q(s)$ be the IMC controller and $G(s)$ the equivalent classical controller:

$$G(s) = \frac{Q(s)}{1 - \tilde{P}(s)Q(s)}$$

$$Q(s) = \frac{G(s)}{1 + \tilde{P}(s)G(s)}$$

With $\tilde{P}(s)$ nominal model of the process.

Internal model control

IMC principle The error in the IMC scheme is defined as $e = y_m - \tilde{y} = \left(P - \tilde{P}\right)u + d + n$. If we are in nominal conditions ($d = 0, n = 0, \tilde{P} = P$) then the error is zero and the open loop dynamic is equal to the closed loop dynamic (in fact, we don't need the feedback since we don't have the error). Therefore, if $Q(s)$ is able to stabilize the nominal process, then we need the feedback only to stabilize the uncertainties.

Complementary transfer function If we are in nominal conditions so the error is zero:

$$T(j\omega) = P(j\omega)Q(j\omega)$$

Sensitivity transfer function In nominal conditions:

$$S(j\omega) = 1 - P(j\omega)Q(j\omega)$$

So they are easier to design since the controller just appears at the numerator.

If P is stable and we are in nominal conditions, to make the open loop stable we just need to design a controller Q stable. This will also guarantee that the equivalent controller is stable too.

IMC design

We require three main characteristics:

- Stability: the controller has to be stable, so its poles have to be in the left half part of the complex plane.
- Properness: the controller has to be proper, so the number of poles $\geq$ to the number of zeros. We will render it proper considering a **filter.**
- Causality: it has to rely on past and present measurements.

We have to consider the <u>asymptotic behaviour</u> of the system in terms of the error to polynomial input.

Type 1 system We would like to have 0 steady state error for step input and disturbances. The condition is:

$$T(0) = 1$$

Type 2 system We would like to have 0 steady state error for ramp input and disturbances. The condition is:

$$\begin{cases} T(0) = 0 \\ \dfrac{dT(s)}{dt}\Big|_{s=0} = 0 \end{cases}$$

Factorization of P

The first thing to do when we want to design the controller is to factorize the nominal process $\tilde{P}(s)$ as:

$$\tilde{P}(s) = \widetilde{P_+}(s)\widetilde{P_-}(s)$$

With $\left|\widetilde{P_+}(0)\right| = 1$

The factorization is performed differently depending on the cost function. First, we have to put the nominal process in the bode form:

$$\tilde{P}(s) = \frac{\prod_i(1 - \beta_i s)\prod_i(1 - \gamma_i s)}{\prod_i(1 + \alpha_i s)} e^{-\theta s}$$

IAE cost function: P_+collects the delay and the positive zeros.

$$\widetilde{P_+}(s) = e^{-\theta s} \prod_i (1 - \beta_i s)$$

$\widetilde{P_-}(s)$ collects the remaining terms.

ISE cost function P_+ collects the delay and the positive zeros but has a denominator whose poles corresponds to the zero with sign changed (so that they are stable poles).

$$\widetilde{P_+}(s) = \frac{\prod_i (1 - \beta_i s)}{\prod_i (1 + \beta_i s)} e^{-\theta s}$$

$\widetilde{P_-}(s)$ collects the remaining terms but we are adding some poles that are not present in the original process, so we have to add these poles as zeros in it.

Then we design the controller as:

$$Q = \widetilde{P_-^{-1}}(s)$$

We have to render the controller proper. In order to do this, we add a filter. For **type 1 system** we use:

$$f(s) = \frac{1}{(1 + \lambda s)^m}$$

Where m should be set in order to render it proper.

For **type 2 system** we use:

$$f(s) = \frac{\left(2\lambda - \frac{d\tilde{P}_+(s)}{ds}\Big|_{s=0}\right) s + 1}{(\lambda s + 1)^2}$$

We have to consider ALWAYS a filter, even when the controller seems realizable.

Then we have to check the *asymptotic behaviour*:

Type 1 system:

$$\tilde{T}(0) = 1 \rightarrow \tilde{T}(0) = \tilde{P}(0)Q(0) \rightarrow f(0) = 1$$

Type 2 system:

$$\begin{cases} \tilde{T}(0) = 0 \\ \dfrac{d\tilde{T}(s)}{dt}|_{s=0} = 0 \end{cases} \Rightarrow \begin{cases} f(0) = 1 \\ \dfrac{d}{ds} \widetilde{P_+}(s) f(s)|_{s=0} = 0 \end{cases}$$

the parameter λ has to be design in order to guarantee stability.

Robust stability

Usually the nominal process is not the perfect model since there may be some uncertainties on the parameters; for instance we consider:

$$P(s) = K_P \frac{N_P(s)}{D_P(s)} \qquad K_P \in [a, b]$$

So we have an uncertainty on the model parameters. We say that a controller G stabilizes the process if it is able to stabilize all the functions whose parameter is between a and b. So we define:

$$\mathcal{P}_p = \{P(s) = K_P \frac{N_P(s)}{D_P(s)} \ \ s.t. K_p \in [a,b]\}$$

And we have to define a controller that stabilizes all the functions in this set.

We can have two kind of uncertainties:

Additive uncertainty We have $P(s) = \tilde{P}(s) + \Delta(s)$ with $\Delta(s)$ parametric additive uncertainty.

In this case, we need to find an upper bound for Δ. So we define $w(j\omega) : |\Delta(j\omega)| < |w(j\omega)|$ and the sufficient condition for the stability is that:

$$|w(j\omega)Q(j\omega)| < 1 \ \forall\omega$$

Multiplicative uncertainty We have $P(s) = \tilde{P}(s)(1 + \Delta_m(s))$.

We need to define an upper bound for Δ_m. So we define $l_m(j\omega)$: $|l_m(j\omega)| > |\Delta_m(j\omega)|$.

The sufficient condition for stability is that:

$$\left|l_m(j\omega)\tilde{T}(j\omega)\right| < 1$$

$$\tilde{T}(j\omega) = \tilde{P}(j\omega)Q(j\omega)$$

Smith predictor

When we cannot use the Padè approximation, we have to use the smith predictor principle. It says that if the nominal model of the process and of the uncertainty are perfect, then the input output characteristic of the system with a delay, are the same of the delay free system.

Therefore, here are the steps to follow when we have a process $P(s) = P_0(s)e^{-\theta s}$ and we cannot use Padè approximation:

1. Design the IMC controller Q for P_0.
2. Obtain the equivalent classical controller $G_0 = \frac{Q_0}{1-Q_0\widetilde{P_0}}$
3. Obtain the smith predictor controller $G = \frac{G_0}{1+G_0(\widetilde{P_0}-\tilde{P})}$

Multiplicative uncertainty of the delay We have $P(s) = \frac{k}{1+s\tau}e^{-\theta s}, \ \ \theta = \bar{\theta} + \delta, \ \ |\delta| \le \bar{\delta}$ (the delay must be bounded). The equation to check for robustness in case of the multiplicative uncertainty is:

$$\left|l_m(j\omega)\tilde{T}(j\omega)\right| < 1$$

$$l_m(j\omega) = \begin{cases} e^{-j\omega\bar{\delta}} - 1 & \omega \le \frac{\pi}{\bar{\delta}} \\ 2 & \omega > \frac{\pi}{\bar{\delta}} \end{cases}$$

To plot it, we use the bode graph with the normalized axis ($\widetilde{\omega} = \omega\bar{\delta}$), so we have to compute $\tilde{T}(j\widetilde{\omega})$. Everything works for $\lambda \ge \bar{\delta}$ ($\lambda > 0.67\bar{\delta}$).