CHAPTER 3: INTERNAL MODEL CONTROL

IMC: One Degree of Freedom Design

3.1 Preliminaries

The general IMC control loop is shown in Figure 1. The objective of the controller, $q(s)$ is to:

- Respond in a desired manner for set point change
- Counter the effect of disturbance that enters directly into the output.

Assumptions:

- The mathematical model, $\tilde{p}(s)$ is a perfect representation of the plant
- The process is linear
- There is no constraint on the control moves (i.e. manipulated variable).

However, extension of the above limitation to cover imperfect models and to accommodate constraints on the control effort is possible.

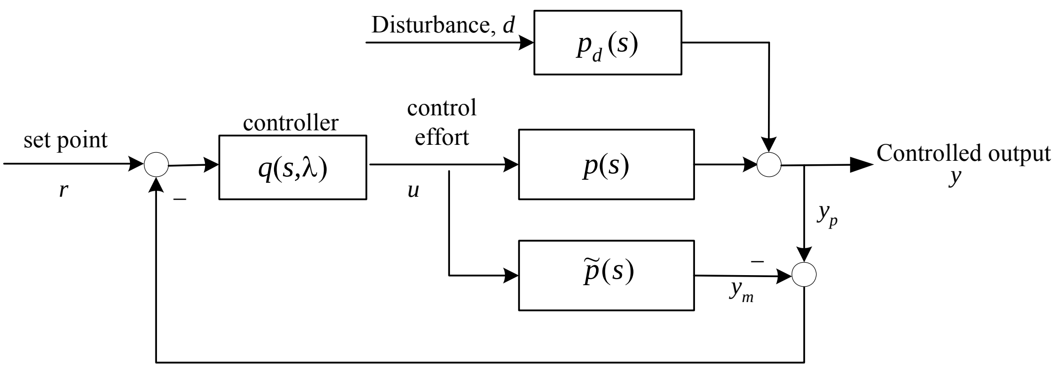


Figure 1: The IMC block diagram

- The usefulness of IMC structure is that it allows for controller design without having to be concerned with system stability provided that the model $\tilde{p}(s)$ is **perfect** and the process is **stable**.

- Even if the model is ***imperfect***, it Is still possible to design the controller $q(s,\lambda)$ without concern for system stability and then select the parameter λ to assure stability.

- If the *controller gain* is the inverse of the *model gain*, then the process output y will eventually follows the set point provided that
  - The process and model gains have the same sign.
  - The controller is tuned to assure stability.

3.2 Properties of IMC

3.2.1 Transfer Functions

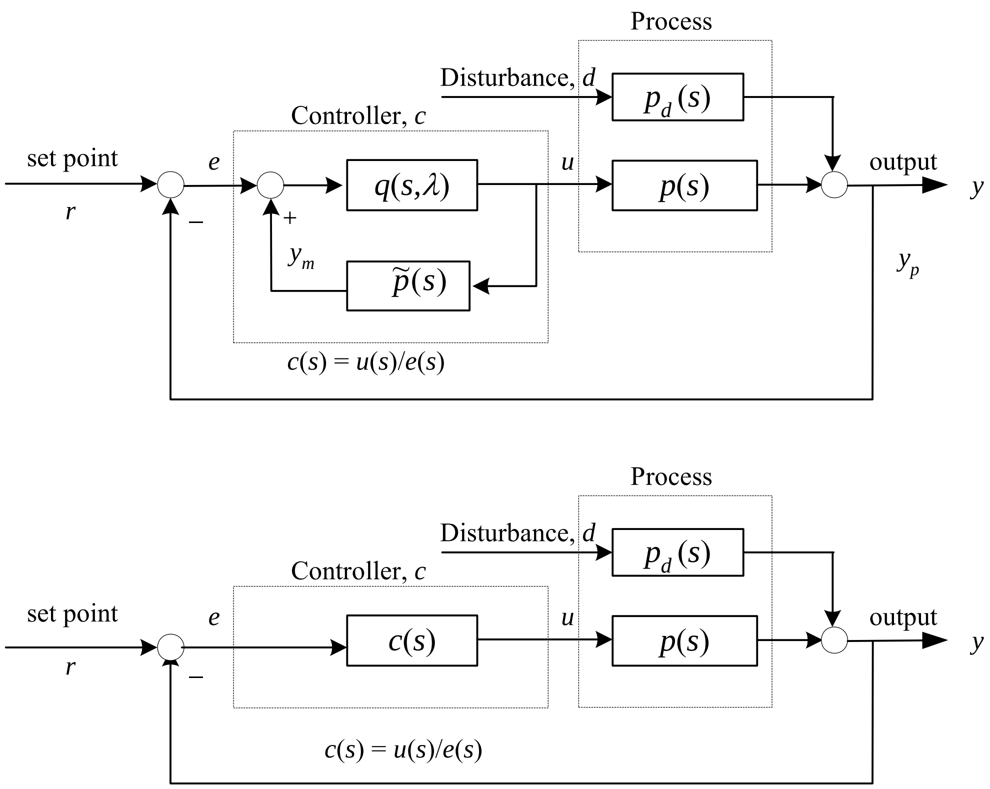


Figure 3: The modified IMC block diagram

The lower part of Figure 3 is put into standard feedback control loop.

$$c(s) = \frac{u(s)}{e(s)} = \frac{q(s)}{1 - q(s)\tilde{p}(s)} \qquad (1)$$

$$\frac{y(s)}{r(s)} = \frac{c(s)p(s)}{1 + c(s)p(s)} \qquad \textbf{(2)}$$

$$\frac{y(s)}{d(s)} = \frac{p_d(s)}{1 + c(s)p(s)} \tag{3}$$

$$\frac{u(s)}{r(s)} = \frac{c(s)}{1 + c(s)p(s)} \tag{4}$$

$$\frac{u(s)}{d(s)} = \frac{-p_d(s)c(s)}{1 + c(s)p(s)} \tag{5}$$

From all above we have:

$$\boxed{y(s) = \frac{p(s)q(s)}{1 + q(s)(p(s) - \tilde{p}(s))} r(s)} \tag{6}$$

$$\boxed{y(s) = \frac{(1 - \tilde{p}(s)q(s))p_d(s)}{1 + q(s)(p(s) - \tilde{p}(s))} d(s)} \tag{7}$$

3.2.2 No Offset Property of IMC (steady state property)

The IMC by design guarantees offset-free feedback response. We choose that the controller gain be the inverse of the model gain, therefore at steady state (i.e. $t \rightarrow \infty$, $s \rightarrow 0$) we have:

For set point change

$$y(0) = \frac{p(0)q(0)}{1 + q(0)p(0) - q(0)\tilde{p}(0)} \tag{8}$$

If we set:

$$q(0)\tilde{p}(0) = 1 \tag{9}$$

Then we get:

$$y(0) = \frac{p(0)q(0)}{q(0)p(0)} = 1 \tag{10}$$

which guarantees perfect tracking of the set point without offset.

For disturbance

$$y(s) = \frac{(1 - \tilde{p}(0)q(0))p_d(0)}{1 + q(0)p(0) - q(0)\tilde{p}(0)} \tag{11}$$

$$q(0)\tilde{p}(0) = 1 \tag{12}$$

$$y(s) = \frac{(0)\, p_d(0)}{q(0)\, p(0)} = 0 \tag{13}$$

which provide perfect rejection of disturbance without offset.

3.2.3 The concept of perfect control (Dynamic property)

The idea of perfect control is that would:

- Force the output to follow the set point instantaneously
- Suppress all disturbances

This implies:

$$y(s) = r(s) \tag{14}$$

$$\frac{y(s)}{d(s)} = 0 \tag{15}$$

Therefore, Equations 6 and 7 require:

$$p(s)q(s) = 1 \tag{16}$$

$$p(s) = \tilde{p}(s) \tag{17}$$

This means for perfect control we need:

- Perfect model
- The controller must perfectly invert the perfect model

Unfortunately, we can never have a perfect model and no controller can perfectly invert the process model.

3.2.4 The Sensitivity and complementary sensitivity functions

The set point tracking vs disturbance rejection dilemma

The sensitivity function for no disturbance dynamics is defined by Eq. 3 for typical feedback controller, *c* as follows:

$$\varepsilon(s) = \frac{y(s)}{d(s)} = \frac{1}{1 + c(s)p(s)} \tag{3}$$

And for the IMC controller, *q* by Eq. 7 as follows:

$$\boxed{\varepsilon = \frac{y(s)}{d(s)} = \frac{1 - \tilde{p}(s)q(s)}{1 + q(s)(p(s) - \tilde{p}(s))}} \tag{7}$$

While the complementary sensitivity function is given by Eq, 2 and 6 as:

$$\eta(s) = \frac{y(s)}{r(s)} = \frac{c(s)p(s)}{1 + c(s)p(s)} \tag{2}$$

$$\boxed{\eta = \frac{y(s)}{r(s)} = \frac{p(s)q(s)}{1 + q(s)(p(s) - \tilde{p}(s))}} \tag{6}$$

If *p*(s)*c*(s) is strictly proper (the degree of denominator is higher than the order of the numerator), then:

$$\lim_{s \to \infty} p(s)c(s) = 0 \tag{18}$$

This implies that:

$$\lim_{\omega \to \infty} \left| \varepsilon(j\omega) \right| = \lim_{\omega \to \infty} \left| \frac{1}{1 + pc(j\omega)} \right| = 1 \tag{19}$$

$$\lim_{\omega \to \infty} \left| \eta(j\omega) \right| = \lim_{\omega \to \infty} \left| \frac{pc(j\omega)}{1 + pc(j\omega)} \right| = 0 \tag{20}$$

Equation (19) implies that the sensitivity to disturbance can be made small only over a finite frequency range.

Equation (20) implies that the response to set point can be made equal to unity only over a finite frequency range.

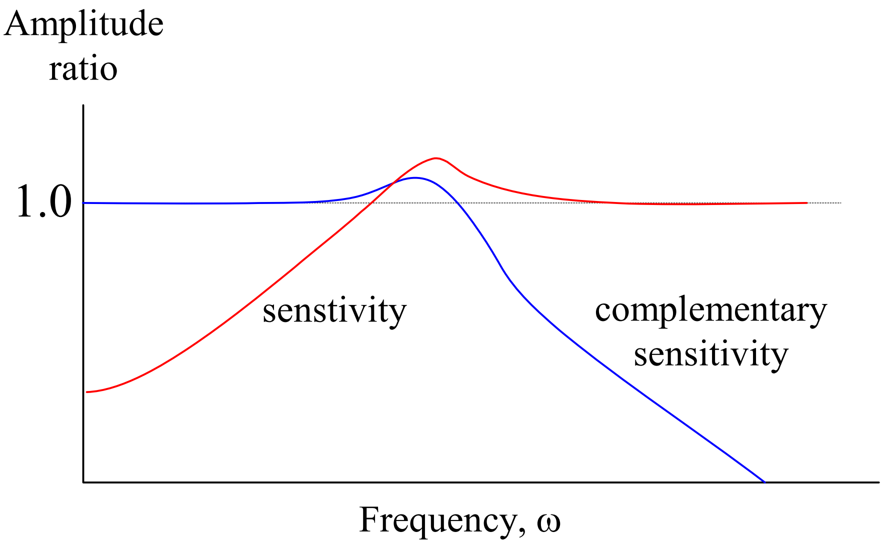


Figure 4: Frequency response of the sensitivity and complementary sensitivity functions

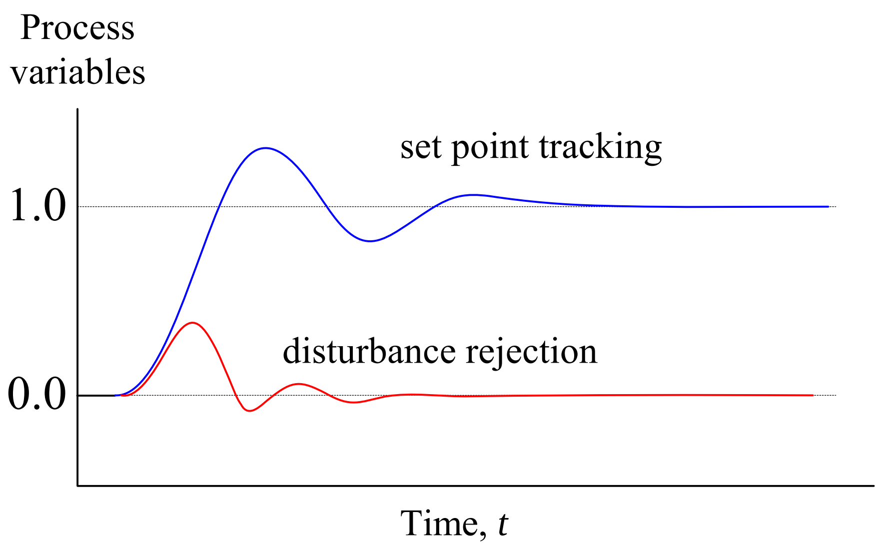
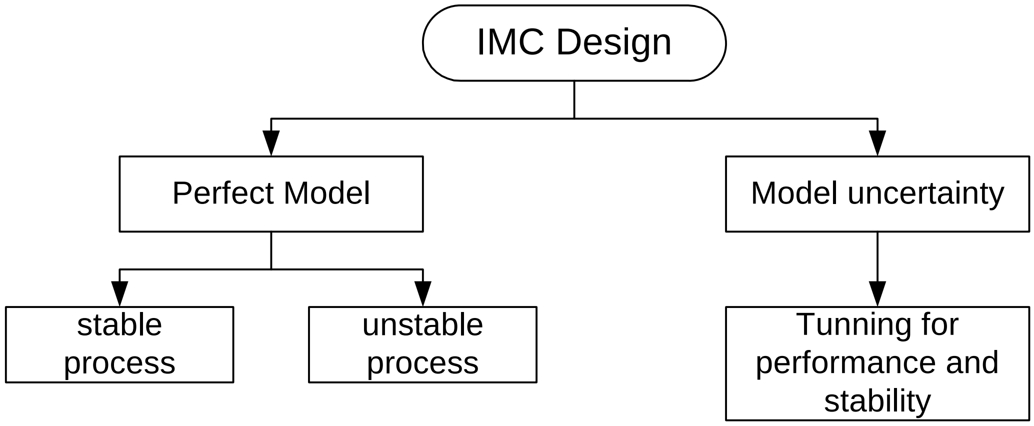


Figure 5: Typical response for set point change and disturbance change


IMC Design
Perfect Model
Model uncertainty
stable process
unstable process
Tunning for performance and stability

A- Perfect Model Case
A-1 Stable Processes

A perfect model implies that the model represent the dynamics of the plant exactly.

3.3 IMC design for 1DF (no Disturbance Lag, $P_d(s)$ = 1)

The basic concept of IMC design is that:

$$q(s) = \frac{1}{p(s)}$$

Example: FOPDT Process

Let the model be perfect, then

$$p(s) = \frac{ke^{-\theta s}}{\tau s + 1} \tag{21}$$

The inverse of the process dynamics is:

$$p(s)^{-1} = \frac{\tau s + 1}{k} e^{\theta s} \tag{22}$$

However the inverse is unrealizable because the term $e^{\theta s}$ requires future prediction of the output and the *lead* term **τs+1** require differentiation of the output.

Therefore, the only IMC controller for this process can be:

$$q(s) = \frac{\tau s + 1}{k(\lambda s + 1)} \tag{23}$$

while λ is a filter time constant that can be used as a tuning parameter to avoid excessive noise amplification and to accommodate modeling errors.

- The term 1/(λs+1) is the IMC filter added to make the controller $q(s)$ realizable and makes the closed-loop response over-damped.
- For modest modeling error, the filter time constant λ will be less than the process time constant τ.

For perfect model (i.e. $p(s) = \widetilde{p}(s)$), equations 6 and 7 become:

$$y(s) = \frac{p(s)q(s)}{1 + q(s)(p(s) - \widetilde{p}(s))} r(s) = \frac{p(s)q(s)}{1} r(s)$$

(24)

$$= \frac{ke^{-\theta s}}{\tau s + 1} \frac{\tau s + 1}{k(\lambda s + 1)} r(s) = \frac{e^{-\theta s}}{(\lambda s + 1)} r(s)$$

$$y(s) = \frac{(1 - \widetilde{p}(s)q(s))\, p_d(s)}{1 + q(s)(p(s) - \widetilde{p}(s))} d(s) = \frac{1 - \widetilde{p}(s)q(s)}{1} d(s)$$

(25)

$$= (1 - p(s)q(s))d(s) = \left(1 - \frac{e^{-\theta s}}{\lambda s + 1}\right) d(s)$$

Note that the term $e^{-\theta s}/(\lambda s+1)$ ➔ 1 at steady state, i.e. as s➔0, therefore

For set point

$y \rightarrow r$ at response speed depending on the value of λ

For any disturbance

$y \rightarrow 0$ at response speed depending on the value of λ

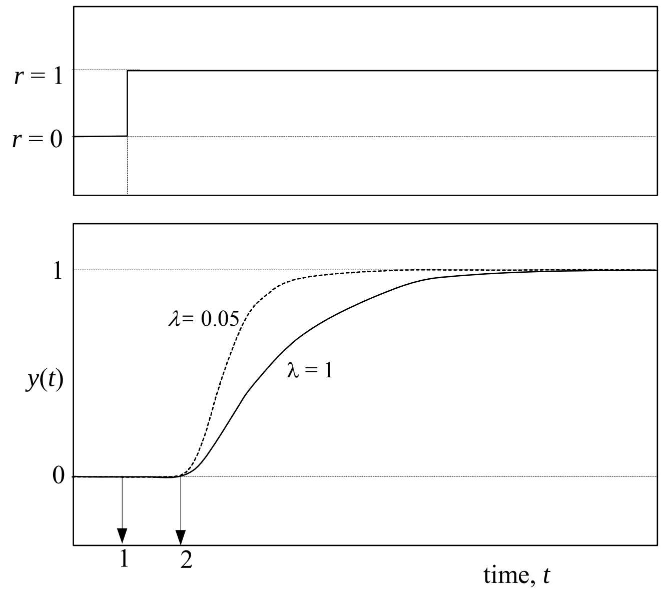


Figure 6: Effect of filter parameter λ.

It is obvious that a smaller value for λ makes the closed-loop response faster. But how much fast the response can be made?

The choice of the filter parameter depends on the allowable noise amplification. Industrial practice uses the following criterion:

$$\left| q(\infty)/q(0) \right| \leq 20$$

Methods for designing the filter parameter will be discussed later.

3.3.1 IMC design for minimum-phase processes

- Minimum-phase processes are processes with no zeros near the imaginary axis or in the right half plane.
- When the process is non-minimum-phase, then its inverse is stable and not overly oscillatory.

In general the process model can be given by:

$$p(s) = \frac{N(s)}{D(s)} \qquad 26$$

The IMC controller for this case is:

$$q(s) = \frac{D(s)}{N(s)(\lambda s + 1)^r} \qquad 27$$

where *r* is the relative order of *N*(s)/*D*(s) and must satisfy:

$$\lambda \geq \left(\lim_{s \to \infty} \frac{D(s)N(0)}{20 s^r N(s) D(0)} \right)^{1/r} \qquad 28$$

Note that equation 28 provides the minimum value for λ.

3.3.2 IMC design for processes with zeros near the imaginary axis

- If the numerator of the model, *N*(s) has roots near the imaginary axis (low damping ratios, ζ<<1), then controller (which is the inverse of the model) will amplify noise excessively at intermediate frequencies.

There are two remedies for this problem:

- Increase the filter time constant, λ sufficiently to reduce the effect of the noise. However, this will excessively increase the settling time of the control system.
- To avoid inverting the low damping zeros.

Example* 3.1: *process with imaginary zero

$$p(s) = \frac{s^2 + 0.001s + 1}{(s+1)^4}$$

Note: use TF, zero and pole MATLAB command

The system has imaginary roots

The relative order of the system $r = 4 - 2 = 2$

According to the design equation (14), the IMC controller is:

$$q(s) = \frac{(s+1)^4}{(s^2 + 0.001s + 1)(\lambda s + 1)^2}$$

and according to equation (15) we have:

$$\lambda = \left(\lim_{s \to \infty} \frac{(s+1)^4}{20s^2(s^2 + 0.001s + 1)} \right)^{1/2}$$

$$\lambda = 0.22$$

Hence the final form of the IMC controller is:

$$q(s) = \frac{(s+1)^4}{(s^2 + 0.001s + 1)(0.22s + 1)^2}$$

To examine the effect of the time filter on the controller frequency response, we use the Bode plot as shown in Figure 7. At frequency ω = 1 and λ = 0.22, the system has a peak of 70dB which equals 3200. This large value will amplify noises entering at that value of frequency.

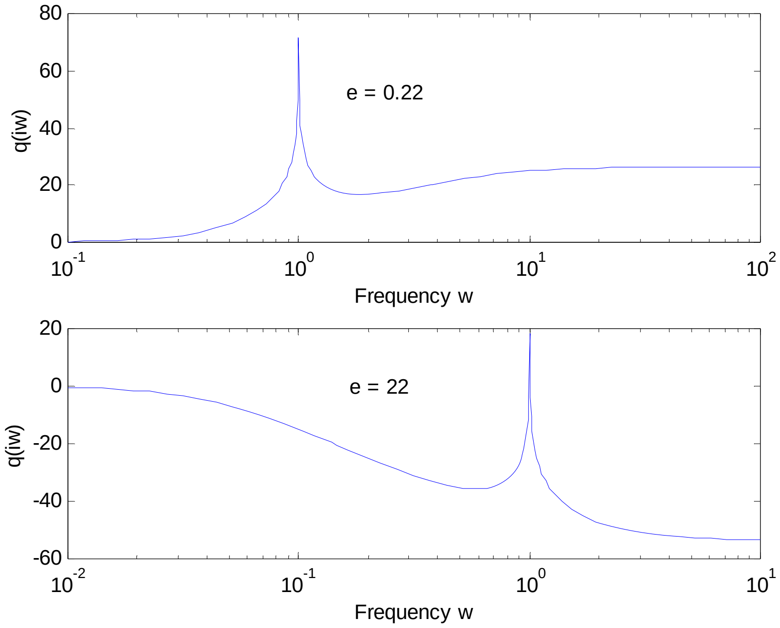


Figure 7: Bode plot of the IMC controller at λ = 0.22 and λ = 22

When the filter time constant is increased to 22, the bode plot of the controller is shown in Figure 7. The peak is reduced to 20 dB which equals 10. This remarkable attenuation of the noise occurs at large settling time:

$$y(s) = \frac{p(s)q(1)}{1} r(s) = \frac{1}{(\lambda s + 1)} r(s)$$

Since the filter time constant is increased from 0.22 to 22 to achieve acceptable noise attenuation, the settling time is increased by 22/0.22 ≈100 times. This will affect the system response to set point change as shown in Figure 8.

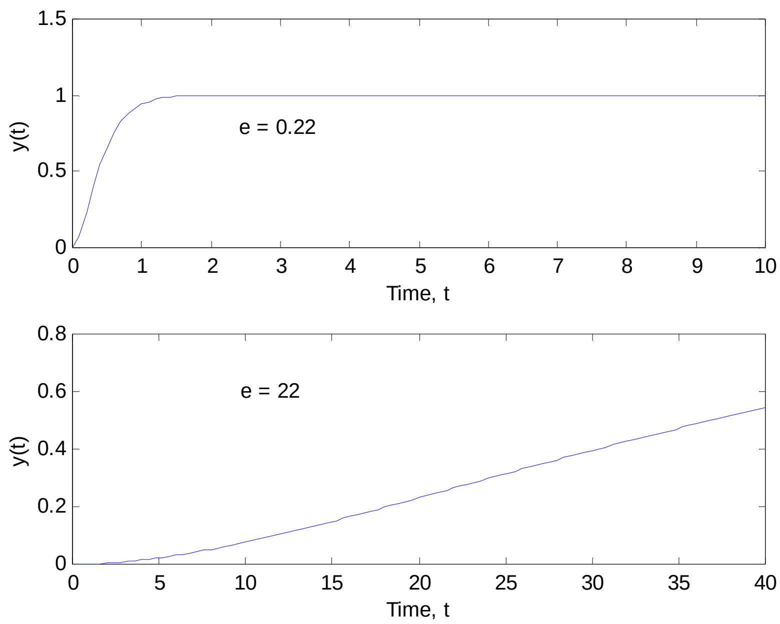


Figure 8: System response to unity set point change

Alternatively to keep the filter time constant and therefore the settling time as it is, i.e. $\lambda = 0.22$ and in the same time reduce the peak of the bode plot, we modify the IMC controller to:

$$q(s) = \frac{(s+1)^4}{(s^2 + 2\zeta s + 1)(0.22s+1)^2}$$

the output response to set point is thus given by:

$$\frac{y(s)}{r(s)} = p(s)q(s) = \frac{s^2 + 0.001s + 1}{(s^2 + 2\zeta s + 1)(0.22s+1)^2}$$

In this case the damping factor ζ is left as a design parameter as shown in the following figures:

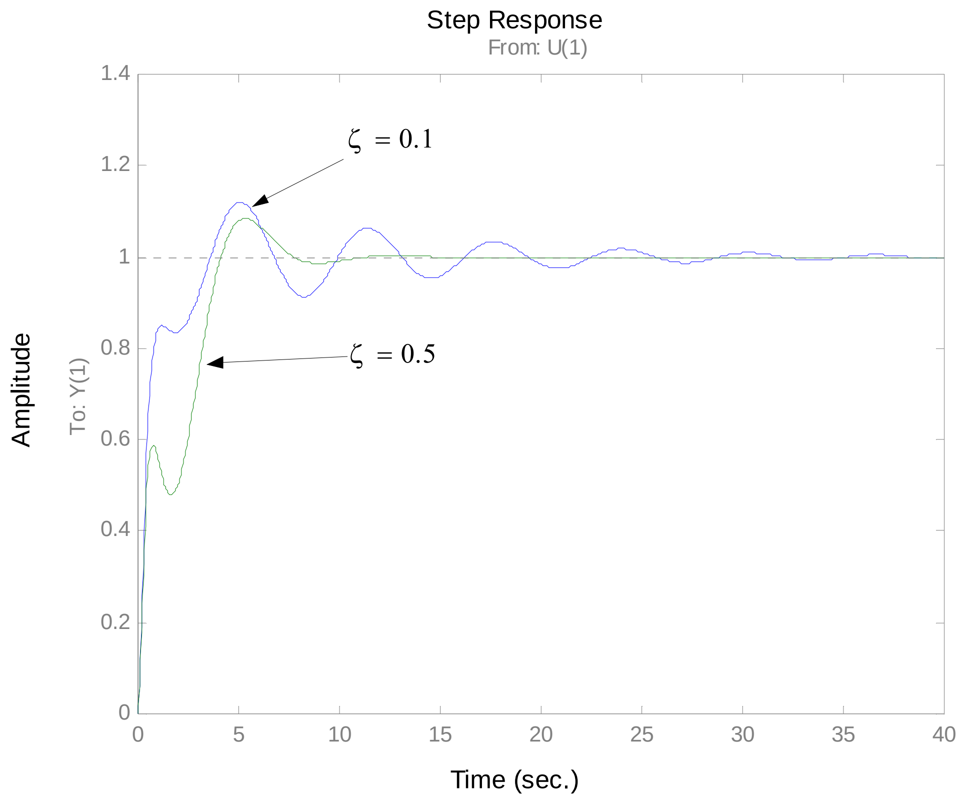


Figure 9: Time response for example 2.1

Larger damping factor gives less oscillation but slower response and vice verse. Thus, it remains an engineering choice.

3.3.3 IMC design for processes with right half plane zeros

- When *N*(s) has RHP zeros, then the process is non-minimum-phase and its inverse is **unstable**.
- Morari and Zafiriou suggested to factor the original model into:

$$p(s) = \frac{N_{-}(s)N_{+}(s)}{D(s)} e^{-\theta s}$$

N_(s) contains only the LHP zeros, none of which have low damping ratio $N_{+}(s)$ contains only RHP zeros, and it is recommended to be written in factorized form as follows:

$$N_{+}(s) = \prod_{i,j} (-\tau_i s + 1)(\tau_j s^2 + 2\tau_j \zeta s + 1)$$

The IMC controller in this case is as follows:

$$q(s) = \frac{D(s)}{N_{-}(s)N_{+}(-s)(\lambda s+1)^{r}}$$

Where $N_{+}(-s)$ is the mirror image of $N_{+}(s)$.

Example* 3.2*:

$$p(s) = \frac{s-1}{27(s+1/3)^{3}}$$

can be written as:

$$p(s) = \frac{-(-s+1)}{(3s+1)^{3}}$$

The IMC controller is

$$q(s) = \frac{-(3s+1)^{3}}{(s+1)(\lambda s+1)^{2}}$$

and the closed-loop response for set point change is given by:

$$\frac{y(s)}{r(s)} = p(s)q(s) = \frac{(-s+1)}{(s+1)(\lambda s+1)^{2}}$$

The system response to set point change of 1 and filter time constant of λ = 0 is shown in Figure 10.

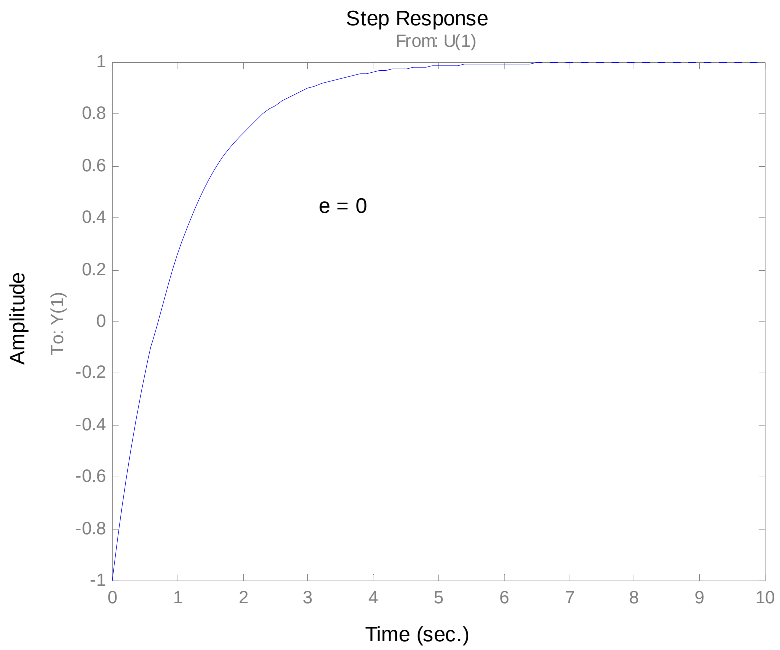


Figure 7: set point change response for example 3.3.1

***Example* 3.3:**

$$p(s) = \frac{1+3e^{-s}}{(s+1)^2}$$

Using fifth order Pade approximation for the time delay:

$$p(s) = \frac{4(0.225s^2 - 0.0964s + 1)(0.0156s^2 + 0.0348s + 1)(0.00235s^2 + 0.0616s + 1)}{(s+1)^2(0.137s+1)(0.0175s^2 + 235s + 1)(0.0138s^2 + 0.128s + 1)} = \frac{N(s)}{D(s)}$$

The relative order is one (r = 1) and numerator ($N(s)$) has one positive root and two negative roots, therefore it can be split into:

$$N\_(s) = (0.0156s^2 + 0.0348s + 1)(0.00235s^2 + 0.0616s + 1)$$

$$N + (s) = (0.225s^2 - 0.0964s + 1)$$

Therefore, the IMC controller is:

$$q(s) = \frac{D(s)}{N_+(-s)N_-(s)(\lambda s + 1)} =$$

$$\frac{(s+1)^2(0.137s+1)(0.0175s^2+235s+1)(0.0138s^2+0.128s+1)}{4(0.225s^2+0.0964s+1)(0.0156s^2+0.0348s+1)(0.00235s^2+0.0616s+1)(\lambda s+1)}$$

The close-loop response is therefore:

$$\frac{y(s)}{r(s)} = p(s)q(s) = \frac{(0.225s^2 - 0.0964s + 1)}{(0.225s^2 + 0.0964s + 1)(\lambda s + 1)}$$

The closed-loop response to $\lambda = 0$ and $\lambda = 0.2$ is shown in Figure 10 below.

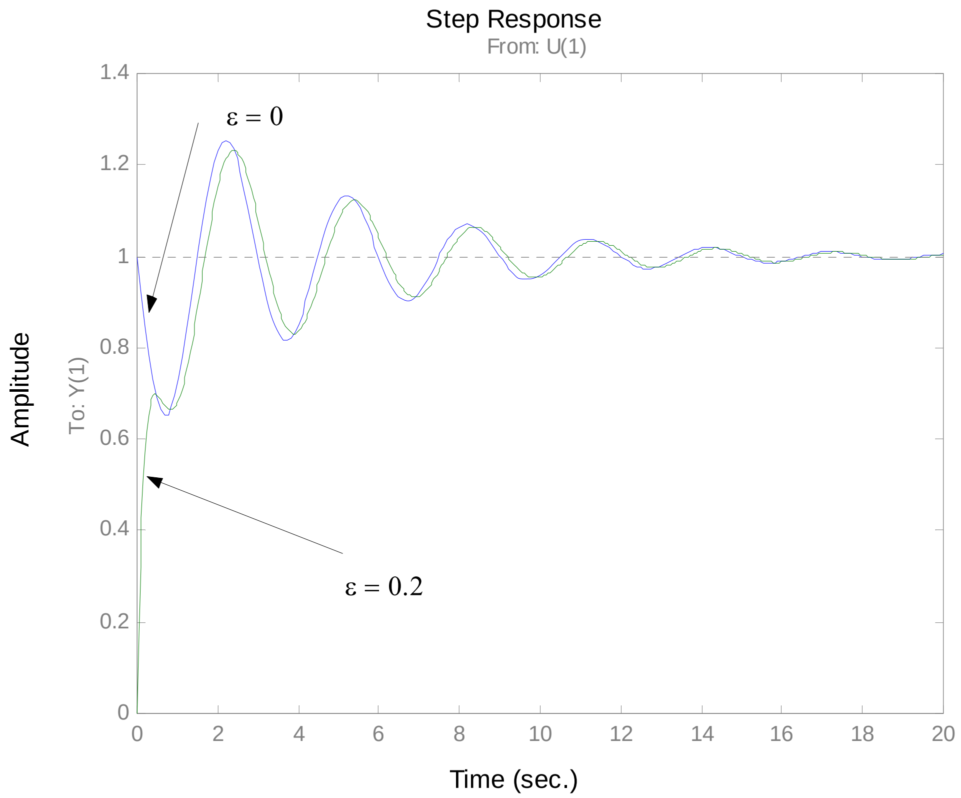


Figure 10: Time response for example 2.3

Conclusion:

- We have designed 1DF (set point only) IMC for *stable* minimumphase and non-minimum-phase processes.

- In case of non-minimum-phase processes, special attention should be made.

- The 1DF IMC controller is good for set point change and can be used disturbance rejection provided that both the reference *signal r and the disturbance d vary in the same manner.*

- If *r* and *d* behave differently, then 1DF controller, *q* should be chosen either for good response to step in *r* or good response for changes in *d*, or elsewhere in between.

- The alternative is 2 DF controller design.

APPENDIX

Useful MATLAB commands

tf	creates transfer function
*	multiply two transfer functions
series	same as *
+, -	add or subtract transfer functions
parallel	connect transfer function in parallel structure
/	divide transfer functions
set	add time delay to a transfer function
pade	approximate time delay by a transfer function
tfdata	extract the numerator & denominator of a transfer function
conv	multiply two polynomials in *s*
deconv	divide two polynomials in *s*
pole	finds the poles of a transfer function
zero	finds the zeros of a transfer function
roots	find the roots of a polynomial in *s*.
residue	perform partial fraction of a transfer function
tcf	factorize the transfer function
tfn	expand a factorize transfer function
Bode	plots the frequency response of a transfer function