IMC Controller and Lambda Tuning Method

Contributors

Name1: كريم أسامة السيد عبدالرحمن
Sec: 2

Name2: مصطفى المهدى أحمد عطوة
Sec: 2

1. Introduction to Internal Model Control (IMC)

Internal Model Control (IMC) is a powerful control design method based on the principle that if the controller contains a model of the **self regulatory process**, perfect control can theoretically be achieved. The controller uses this internal model to predict the plant's behavior and correct for disturbances.

The key idea in IMC is:

- Use a model to **simulate** the plant.
- Compare the real plant output to the model output.
- Correct the input based on the **difference** between actual output and predicted output.

2. Objective of IMC

The IMC controller is designed to achieve a desired closed-loop performance.

Suppose we want the closed-loop transfer function:

$T(s) = 1 / (\lambda s+1)$

where:

- λ (lambda) is a **tuning parameter** controlling the **speed of response**.

3. Mathematical Derivation of the IMC Controller

Assume the plant is modeled as a **First-Order Plus Dead Time (FOPDT)** model:

$$G(s)= (K/(\tau s + 1))e^{(-\theta s)}$$

Where:

- K = Process gain
- τ = Time constant
- θ = Dead time (delay)

We **cannot invert the dead time** (it is non-causal), so we only invert the delay-free part:

Approximate G(s) as:

$$G^*(s) = K / (\tau s+1)$$

Thus, the IMC controller is built on Ideal Process:

$$Q(s) = (G^*(s)^{-1})(1 / (\lambda s + 1))$$

then:

$$Q(s)=((\tau s+1) / K)(1/ (\lambda s + 1))$$

Thus:

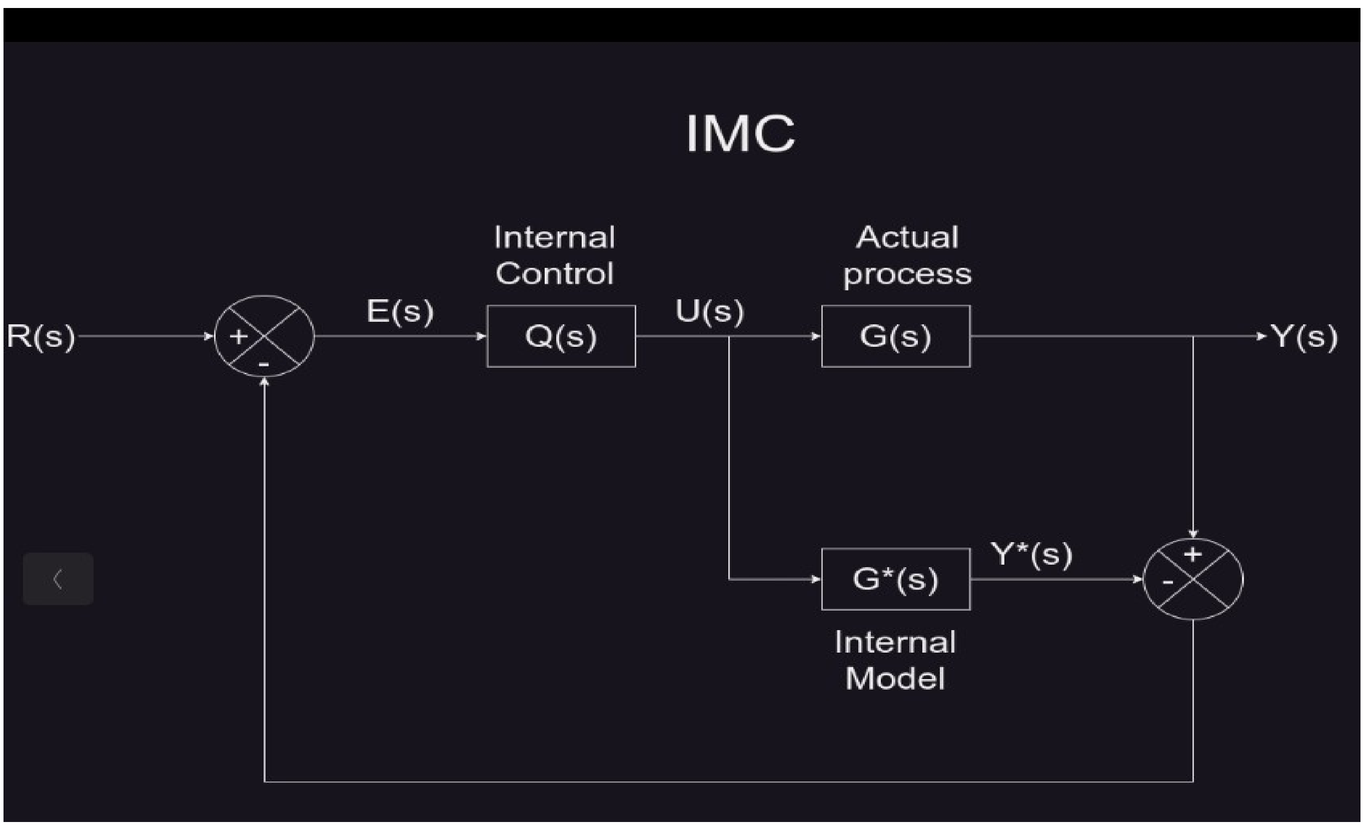
$$Q(s)=(\tau s + 1) / (K(\lambda s+1))$$

4. IMC Structure

IMC Built on comparing theoretical output from $Q(s)G^*(s)$ with real output $Q(s)G(s)$ feedback them:

The IMC structure consists of:

- A controller $Q(s)$
- The actual process $G(s)$
- A theoretical model of the process $G^*(s)$

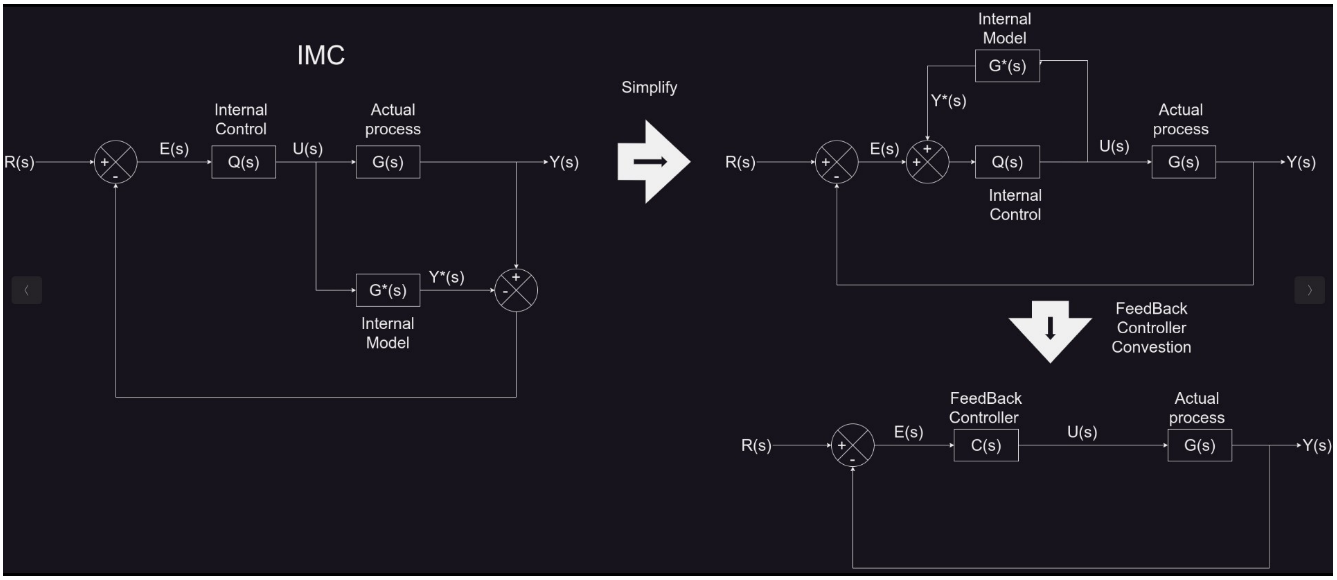


5. Conversion to Feedback Controller

We usually work with **feedback controllers** in practice. To convert IMC controller $Q(s)$ to a feedback controller $C(s)$, use:

$$C(s) = Q(s) / (1 - Q(s)G^*(s))$$

Delivered as:



6. Conversion to PID Controller

First Substituting G(s) and Q(s), and using a Padé approximation for the dead time:

$$e^{(-\theta s)} \approx (1 - \theta s/2) / (1 + \theta s/2)$$

The derivation (after some algebra) shows that C(s) will have a **PID form**:

$$C(s) = Kc(1 + 1/(Ti*s) + Td*s)$$

where Kc, Ti, and Td are PID parameters.

The resulting PID parameters are:

- **Proportional Gain**:

$$Kc = (\tau + 0.5\theta) / K(\lambda + 0.5\theta)$$

- **Integral Time**:

$$Ti = \tau + 0.5\theta$$

- **Derivative Time**:

$$Td = \tau\theta / (2\tau + \theta)$$

Notes:

- The dead time θ is included because it delays the system's response.
- λ is selected by the designer to tradeoff **performance** vs. **stability**.

7. Add Low-Frequency Integrator for Zero Steady-State Error

To ensure **zero steady-state error** while maintaining system stability, we modify the controller by **adding a low-frequency integrator** in parallel with the main PID controller. The total controller transfer function becomes

$$C_{total}(s) = C_{pid}(s) + K_i / (s + \epsilon)$$

Where:

- $C_{PID}(s)$ is the standard PID controller designed using IMC tuning rules,
- K_i is a small gain (typically between 0.1 and 1.0),
- ϵ is a small positive constant (typically between 0.01 and 0.1) that acts as a **low-pass filter** to prevent destabilization from the high gain at high frequencies.

8. Effect of λ (Tuning Parameter)

- **Small λ**: Fast response and error decrease, but less stable (more sensitive to model errors).
- **Large λ**: Slower response and error increase, but much more stable

Conclusion

The **IMC Lambda tuning method** provides a systematic way to tune PID controllers starting from the model of the plant.
By adjusting λ, we can control the tradeoff between speed and stability easily.
The method leads to simple, practical formulas that account for dead time and time constants, making it widely used in industrial PID tuning.