

PID Tuning – Frequency Response Method

Presented to Dr. Ahmad othman

Ahmad Ibrahim Kasim Mohammed | 6302

Ahmad El-Zaki Ahmad Hassan | 6303

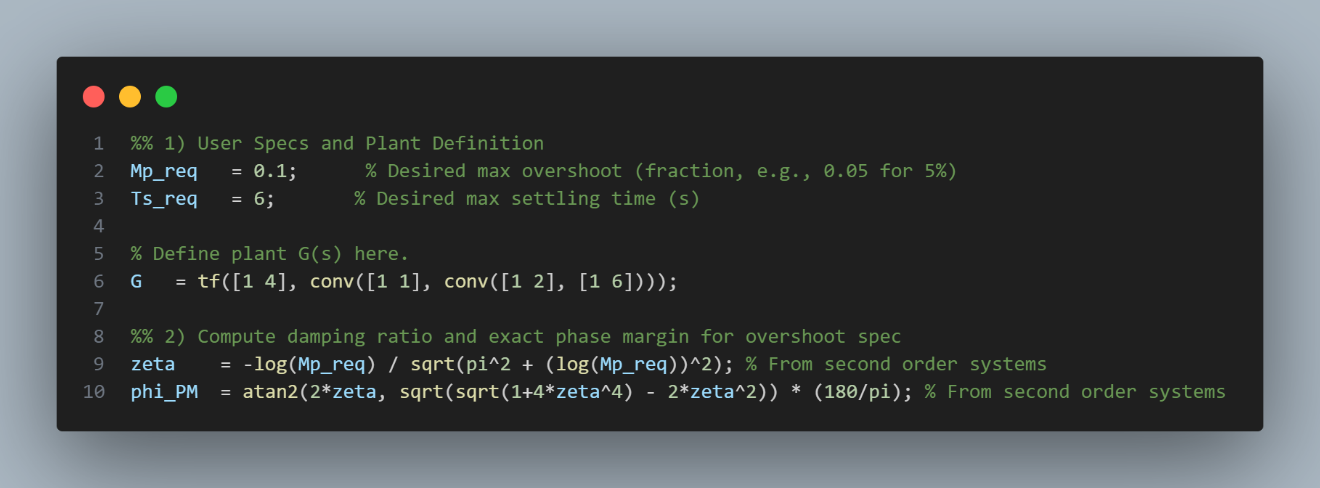
## **1. Introduction**

PID controllers are fundamental in control systems due to their versatility and ease of implementation. Among various tuning methods, the **frequency-response approach** provides a direct link between time-domain performance and frequency-domain stability measures, such as phase margin and crossover frequency.

This report first outlines the standard frequency-domain PID tuning method. Then, it presents a semi-automated implementation based on **second-order system approximations**, enabling control over **maximum overshoot** and **settling time** through systematic calculations.

## **2. Classical Frequency-Response PID Tuning**

### **2.1 Design Specifications**



Define time-domain specifications:

* Maximum overshoot: M\_p (e.g., 0.05 for 5%)
* Settling time: T\_s (in seconds)

These are then converted into frequency-domain targets:

* **Damping ratio** \zeta based on overshoot:
* **Required phase margin** \phi\_{PM} for a second-order approximation:
* **Target open-loop phase** at crossover frequency:

### **2.2 Crossover Frequency Selection**

A screen shot of a computer program

AI-generated content may be incorrect.

The crossover frequency \omega\_c is chosen where the plant’s phase response matches the above target phase. At that frequency, the magnitude of the open-loop transfer function is forced to 1 by adjusting the proportional gain K\_p.

### **2.3 Controller Formulas**

A computer screen shot of a black screen

AI-generated content may be incorrect.

Based on the plant characteristics and the selected \omega\_c, controller gains are calculated as follows:

* **P Controller:**
* **PI Controller:**
* **PID Controller:**

### **2.4 Performance Evaluation**

A screen shot of a computer program

AI-generated content may be incorrect.

The closed-loop system is:

Performance metrics such as rise time T\_r, settling time T\_s, overshoot M\_p, and steady-state error e\_{ss} are computed from the step response. Phase margin and gain margin are evaluated from the loop transfer function.

## **3. Semi-Smart PID Tuning Using Second-Order Approximations**

This MATLAB-based method automates tuning and checks performance specifications through a frequency-domain approach, incorporating:

* Explicit **overshoot and settling time constraints**
* **Second-order approximation** of the system for performance estimation
* Handling of **integrators** in the plant

### **3.1 Inputs and Setup**

* Specify desired M\_p and T\_s
* Define the plant G(s) using tf(...)
* Detect and account for **integrators** by scaling T\_i

### **3.2 Frequency Domain Computation**

* Calculate \zeta and \phi\_{PM} from M\_p
* Use interpolation to find the crossover frequency \omega\_c where the plant phase matches -180^\circ + \phi\_{PM}
* Evaluate plant magnitude at \omega\_c to compute controller gains

### **3.3 Controller Loop**

A screen shot of a computer program

AI-generated content may be incorrect.

For each controller type (P, PI, PID):

* Compute gains using formulas above
* Build open-loop transfer function L(s) = C(s)G(s)
* Evaluate closed-loop T(s) and extract:
  + T\_s (settling time)
  + M\_p (overshoot)
  + T\_r (rise time)
  + e\_{ss} (steady-state error)
* Check if specs are met:
  + If yes: plot response and print info
  + If no: discard controller

A screenshot of a computer

AI-generated content may be incorrect.

### **3.4 Visualizations and Results**

* Annotated step response for successful controllers
* Bode plot of the plant with target phase margin and frequency
* Automatic rejection of controllers that do not meet both M\_p and T\_s
* Simulink simulation if a controller is found viable.

## **4. Summary of Workflow**

1. Define M\_p and T\_s
2. Model G(s)
3. Compute \zeta, \phi\_{PM}, and target phase
4. Interpolate plant phase to find \omega\_c
5. Compute controller gains
6. Simulate step response
7. Evaluate and select only valid controllers
8. Showcase a quick Simulink simulation if a controller is found viable.

## **5. Advantages of the Semi-Smart Method**

* Directly links time-domain specifications to controller design
* Automates gain calculation and performance verification
* Adapts to integrating plants by modifying integral time
* Avoids over-tuning or under-tuning by filtering out unsuitable designs
* Easily extendable to more sophisticated methods or additional constraints

## **6. Conclusion**

This report covers both traditional frequency-domain PID tuning for 2nd order systems and extends it in a practical semi-automated method using second-order system approximations for higher order systems. By integrating time-domain specifications into frequency-based controller design, this approach simplifies the tuning process while maintaining robust performance. The MATLAB implementation provides a flexible framework for applying these principles to a wide range of control systems.