PMSM Position Control
Homework Report

Student Name: Mustafa Yücel
Student ID: 040200626
Lecturer: Prof. Dr. Metin Gökaşan

June 9, 2025

Contents

1 Introduction

This report presents the design and implementation of a position control system for a Surface-mounted Permanent Magnet Synchronous Motor (SPMSM) operating under load disturbances. The main control objective is to ensure accurate and stable shaft position tracking in response to a step reference input, even in the presence of mechanical inertia and friction.

The control scheme employs a direct position feedback loop without a cascaded speed control structure. The rotor position $\theta_m(t)$ is measured using an incremental encoder mounted on the motor shaft. The torque is directly regulated by controlling the q-axis stator current i_q, while the d-axis current i_d is maintained at zero to maximize torque per ampere and simplify control.

Assuming $i_d = 0$ leads to a constant rotor flux linkage λ_{PM}, making it possible to approximate $i_q \approx i_q^{ref}$. This simplification is particularly valid for SPMSM, where the permanent magnets are surface-mounted, and magnetic saliency is negligible. Thus, torque production becomes linearly proportional to the q-axis current.

The hysteresis current control strategy is used in the inner current loop to ensure fast and accurate current regulation. The position controller is responsible for generating the reference torque (or i_q^{ref}) required to track the reference position with minimal steady-state error and acceptable transient performance.

Key objectives of this control design include:

- Zero steady-state error for step reference inputs,
- Fast settling time with minimal overshoot,
- Robustness against load disturbances and system nonlinearities.

2 System Block Diagram (Part a)

The system architecture comprises several interconnected control and plant components, each represented as a distinct functional block. Figure 1 shows a detailed signal flow diagram for the entire position control system, including current regulation, torque generation, and electromechanical dynamics.

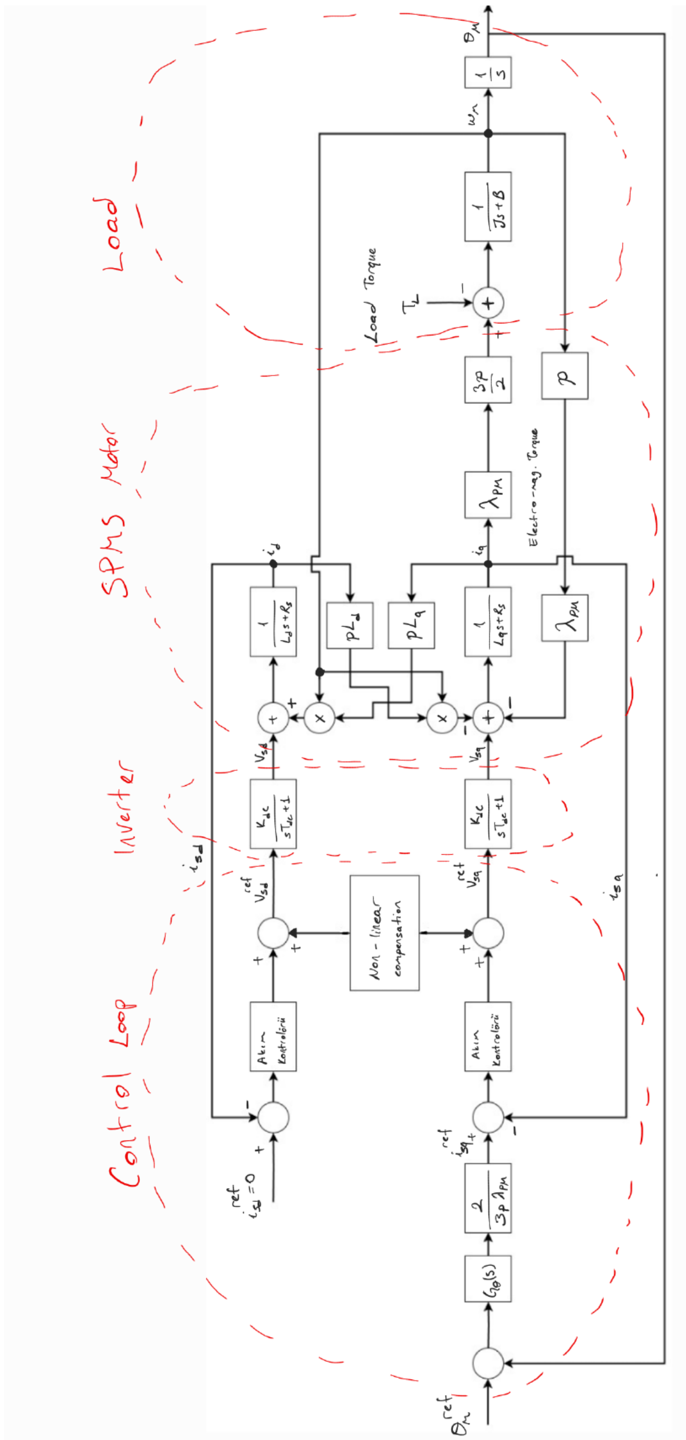

Figure 1: Detailed block diagram of the SPMSM position control system. (Horizontal)
Note: For simplicity, the block diagram omits i_d since for a Surface-mounted PMSM (SPMSM), $i_d^{ref} = 0$ is a valid assumption due to the non-salient rotor. Torque is primarily controlled through i_q, and the current controllers ensure $i_q \approx i_q^{ref}$.

- **Reference Input (θ_{ref}):** The desired shaft angle provided by an external trajectory planner or user input.

- **Position Controller ($G_\theta(s)$):** A proportional-integral-derivative (PID) controller that converts position error into a torque-producing current reference i_q^{ref}. At least one integrator is required in the loop to ensure zero steady-state error for a step reference input.
- **Current Controllers:** Hysteresis-based or PI current controllers regulate i_q and i_d to track their references. Since $i_d^{ref} = 0$, only i_q directly contributes to torque.
- **Nonlinear Compensation:** Cross-coupling terms and back-EMF effects are compensated to linearize the plant dynamics.
- **Electromechanical Plant:** The motor torque is computed as $T_e = \frac{3p}{2}\lambda_{PM} i_q$. The mechanical dynamics are modeled by $\frac{1}{Js+B}$, incorporating motor and load inertia and viscous damping.
- **Encoder Feedback:** The shaft position θ_m is measured and fed back to close the control loop.

Control Loop Structure

The control system is configured as a unity-feedback loop:

- The position error signal is computed as $e(t) = \theta_{ref}(t) - \theta_m(t)$.
- The controller $G_\theta(s)$, implemented as a PID controller, produces the torque-producing current reference i_q^{ref}.
- i_d^{ref} is set to zero. The i_d and i_q currents are regulated via current controllers to track their respective references.
- Electromagnetic torque T_e is computed from i_q and applied to the mechanical plant, modeled by $\frac{1}{J_{eq}s+B_{eq}}$.
- The measured position $\theta_m(t)$ is fed back to close the loop.

The transfer function of the position controller is:

$$G_\theta(s) = K_p + \frac{K_i}{s} + K_d s$$

Note: A complete vector control diagram including i_d, i_q axes, cross-coupling compensation, and inverter dynamics is beyond the scope of this report but can be added in future work.

Mathematical Modeling Using Park Transformation

The internal dynamics of the SPMSM are modeled using the Park transformation. Assuming equal inductances in d and q axes ($L_d = L_q = L_s$), the stator voltage equations are:

$$v_d = R_s i_d + \frac{d\lambda_d}{dt} - p\omega_m \lambda_q$$
$$v_q = R_s i_q + \frac{d\lambda_q}{dt} + p\omega_m \lambda_d$$

The flux linkages are:

$$\lambda_d = L_s i_d + \lambda_{PM}$$
$$\lambda_q = L_s i_q$$

The electromagnetic torque is calculated by:

$$T_m = \frac{3p}{2}\lambda_{PM} i_q$$

These equations highlight the coupling between d and q axes due to back-EMF and rotational effects.

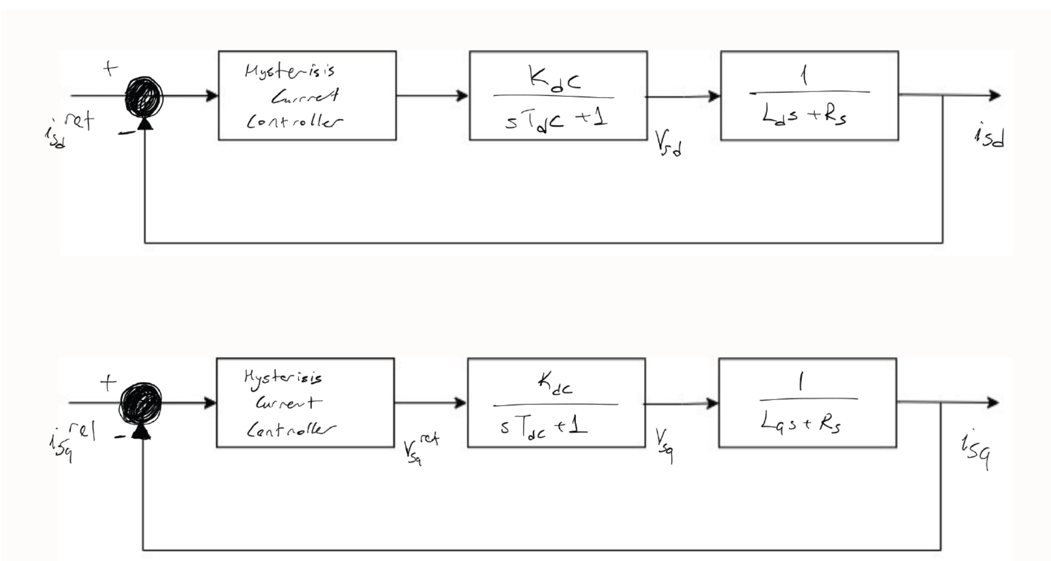

Figure 2: Simplified (compensated) i_d and i_q current control loops, including hysteresis controller, DC gain, and plant dynamics.

When nonlinearities (e.g., cross-coupling and back-EMF terms) are compensated, the current dynamics simplify to:

$$v_d = L_s \frac{di_d}{dt} + R_s i_d, \quad v_q = L_s \frac{di_q}{dt} + R_s i_q$$

Thus, both current loops can be approximated as:

$$G_d(s) = G_q(s) = \frac{1}{L_s s + R_s}$$

These first-order systems enable straightforward PI or hysteresis current controller design.

Simplified Position Control Structure Under Hysteresis Control

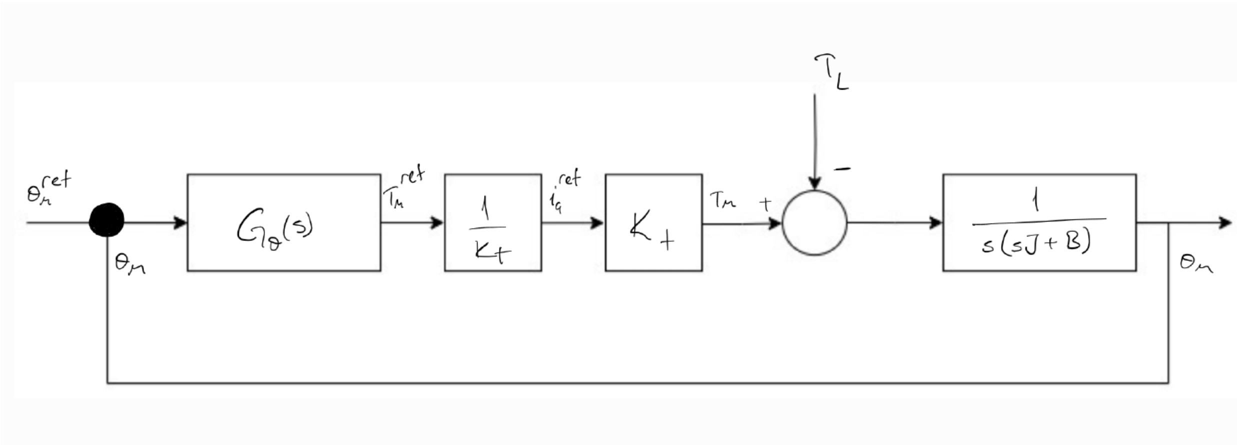

Figure 3: Simplified position control system of SPMSM. Hysteresis current control enables the assumption $i_d^{ref} = 0$, allowing direct torque control via i_q. The torque constant is given by $K_t = \frac{3p}{2}\lambda_{PM}$, and is used to convert i_q to electromagnetic torque T_m.

Note: In this structure, it is assumed that the inverse of K_t is applied to transform the reference torque T_m^{ref} into a current reference i_q^{ref}, which is then tracked by the inner current loop.

3 System Parameters

Parameters used for modeling and analysis:

- Mass: $M = 2.4$ kg
- Linear Damping: $B = 0.8$ Ns/m
- Radius: $r = 0.5$ m
- Drum Inertia: $J_t = 0.2\ \text{kg}\,\text{m}^2$
- Drum Damping: $B_t = 0.6$ Nm s/rad
- Motor Inertia: $J_m = 0.2\ \text{kg}\,\text{m}^2$
- Motor Damping: $B_m = 0.2$ Nm s/rad
- Torque Constant: $K_t = 1$ Nm/A
- Back-EMF Constant: $K_b = 1$ Vs/rad

Motor Type Assumption: The motor is assumed to be a Surface-mounted PMSM (SPMSM). In this configuration, the rotor does not exhibit saliency, so the inductances are equal ($L_d = L_q$), and there is no reluctance torque. This assumption simplifies the torque expression and allows for direct i_q control without the need for i_d-axis modulation.

4 Controller Design (Part b)

Control System Block Diagram

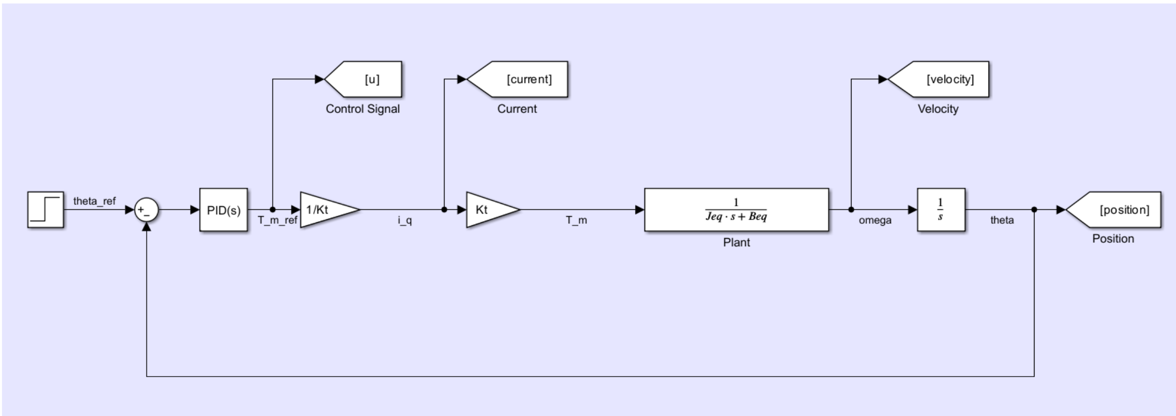

Figure 4: Control system block diagram (PID controller without saturation)

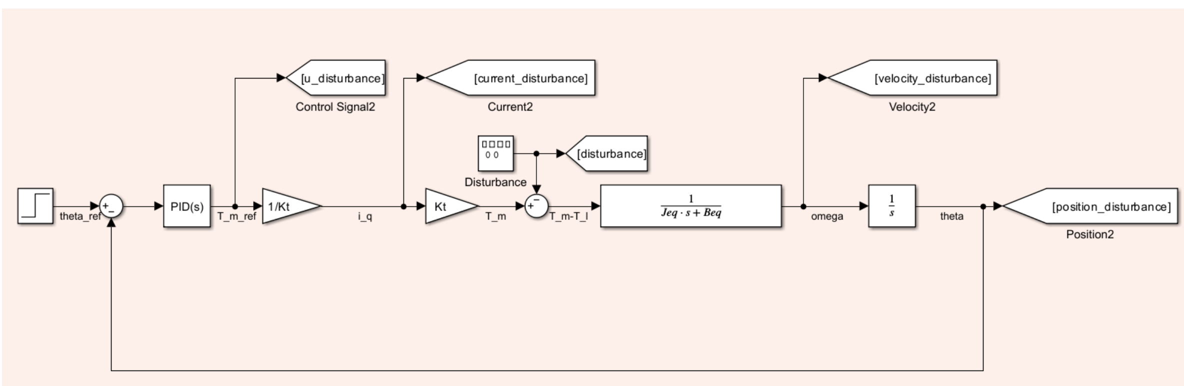

Figure 5: Case 4 – Control system with both actuator saturation and periodic disturbance torque

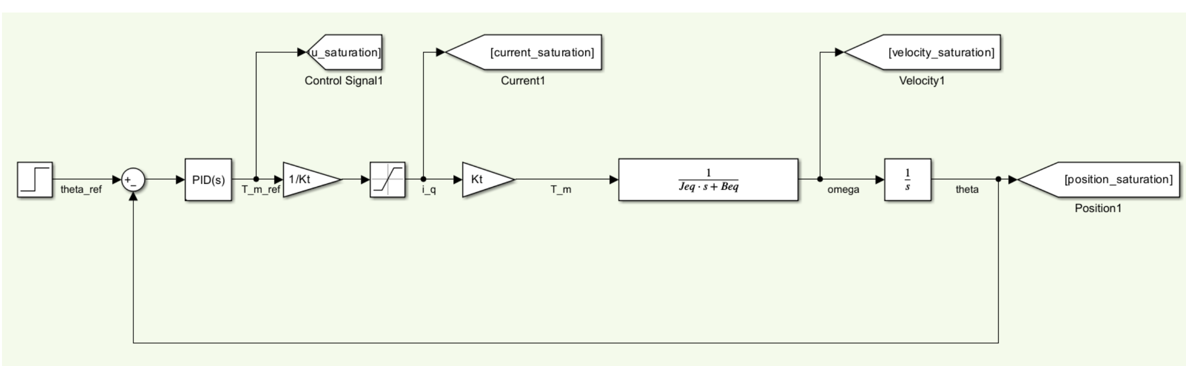

Figure 6: Control system block diagram with current saturation

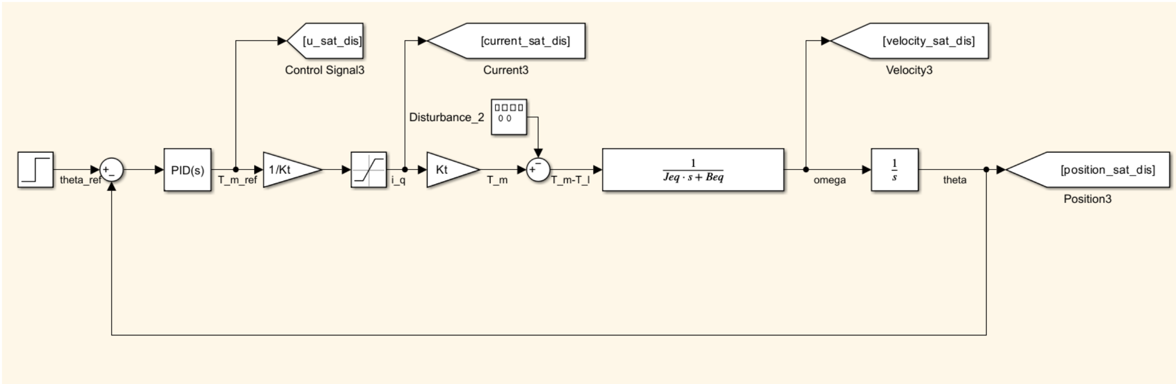

Figure 7: Control system block diagram with saturation and disturbance

Figure Descriptions:

- **Figure 10:** PID controller without external load or saturation constraints.
- **Figure 11:** PID controller with external disturbance torque $T_L(t) = 0.5 \cdot \text{square}(0.125 \cdot t)$.
- **Figure 12:** PID controller with actuator saturation limits (± 5 Nm, ± 20 A).
- **Figure 13:** Combined case with both periodic disturbance and actuator saturation.

System Modeling

The equivalent inertia and damping are:

$$J_{eq} = J_m + J_t + Mr^2 = 1.0 \text{ kg} \cdot \text{m}^2, \quad B_{eq} = B_m + B_t + Br^2 = 1.0 \text{ Nm} \cdot \text{s/rad}$$

The simplified plant transfer function:

$$P(s) = \frac{\theta_m(s)}{i_q(s)} = \frac{1}{s^2 + s}$$

Initial PI Design Attempt

We first attempt to design a **PI controller** of the form:

$$C(s) = K_p + \frac{K_i}{s}$$

The plant transfer function is:

$$P(s) = \frac{1}{s^2 + s}$$

Thus, the open-loop transfer function becomes:

$$C(s)P(s) = \left(K_p + \frac{K_i}{s}\right) \cdot \frac{1}{s^2 + s} = \frac{K_p s + K_i}{s(s^2 + s)} = \frac{K_p s + K_i}{s^3 + s^2}$$

The closed-loop transfer function is then:

$$G_{\text{cl}}(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{K_p s + K_i}{s^3 + s^2 + K_p s + K_i}$$

So, the **closed-loop characteristic equation denominator** becomes:

$$s^3 + s^2 + K_p s + K_i$$

To meet the desired settling time $T_s = 8$ seconds, we choose dominant poles with a real part $\sigma = 0.5$, and include a fast third pole at $s = -2.5$. Let the third pole be $s = -a$, yielding the desired characteristic polynomial:

$$(s + 0.5)(s + 2.5)(s + a)$$

Expanding this step by step:

$$\begin{aligned}(s + 2.5)(s + a) &= s^2 + (2.5 + a)s + 2.5a \\ (s + 0.5)(s^2 + (2.5 + a)s + 2.5a) &= s^3 + (a + 3)s^2 + (3a + 1.25)s + 1.25a\end{aligned}$$

This is the **target polynomial** that our closed-loop system should match.

Now we compare the coefficients of the closed-loop characteristic equation:

$$s^3 + s^2 + K_p s + K_i$$

with the target polynomial:

$$s^3 + (a + 3)s^2 + (3a + 1.25)s + 1.25a$$

By matching corresponding terms:

$$\begin{aligned}a + 3 = 1 &\Rightarrow a = -2 \\ 3a + 1.25 = -6 + 1.25 = -4.75 &\Rightarrow K_p = -4.75 \\ 1.25a = -2.5 &\Rightarrow K_i = -2.5\end{aligned}$$

Conclusion: As a result, the gains derived for the PI controller are:

$$K_p = -4.75, \quad K_i = -2.5$$

Both gains are negative, which is generally undesirable in practical control systems due to issues with stability and implementation. Therefore, a PI controller is **not sufficient** to meet the design specifications, and a higher-order controller such as **PID** is necessary.

Why Not Use a PD Controller?

A PD controller has no integrator, and thus the open-loop transfer function would have no pole at the origin. This makes the closed-loop system a **Type 0** system, which is incapable of eliminating steady-state error for a step reference input.

Since the main performance criterion specified in the problem includes **zero steady-state error**, a PD controller alone is insufficient. Therefore, at least one integrator is required in the control structure, which makes PI or PID controllers more suitable options.

PID Controller Design with Polynomial Matching

The plant transfer function is given by:

$$P(s) = \frac{1}{s^2 + s}$$

We consider a PID controller of the form:

$$C(s) = K_p + \frac{K_i}{s} + K_d s$$

The open-loop transfer function is:

$$L(s) = C(s)P(s) = \left(K_p + \frac{K_i}{s} + K_d s\right) \cdot \frac{1}{s^2 + s} = \frac{K_d s^2 + K_p s + K_i}{s(s^2 + s)} = \frac{K_d s^2 + K_p s + K_i}{s^3 + s^2}$$

The closed-loop transfer function becomes:

$$T(s) = \frac{L(s)}{1 + L(s)} = \frac{K_d s^2 + K_p s + K_i}{s^3 + (1 + K_d)s^2 + K_p s + K_i}$$

Therefore, the **characteristic polynomial** of the closed-loop system is:

$$s^3 + (1 + K_d)s^2 + K_p s + K_i$$

Desired Characteristic Polynomial:
The desired settling time is $T_s = 8$ seconds. Using the relation:

$$T_s = \frac{4}{\sigma} \Rightarrow \sigma = 0.5$$

we choose all real poles: two dominant poles at $s = -0.5$ (double pole) and a faster third pole at $s = -2.5$. This yields the desired polynomial:

$$(s + 0.5)^2(s + 2.5) = s^3 + 3.5s^2 + 2.75s + 0.625$$

Matching with Closed-Loop Polynomial:

$$\underbrace{s^3 + (1 + K_d)s^2 + K_p s + K_i}_{\text{Closed-loop characteristic equation}} = \underbrace{s^3 + 3.5s^2 + 2.75s + 0.625}_{\text{Target polynomial}}$$

By comparing coefficients:

$$\begin{aligned} 1 + K_d = 3.5 \quad &\Rightarrow \quad K_d = 2.5 \\ K_p &= 2.75 \\ K_i &= 0.625 \end{aligned}$$

Final PID Controller Parameters

- $\boxed{K_p = 2.75}$
- $\boxed{K_i = 0.625}$
- $\boxed{K_d = 2.5}$

Conclusion: The selected PID gains result in a closed-loop polynomial that exactly matches the desired system dynamics. The dominant poles achieve the specified settling time, while the additional third pole ensures minimum-order implementation without adversely affecting system response.

Justification of Pole Placement and Settling Time Target

The desired settling time was given as $T_s = 8$ seconds. Using the approximation:

$$T_s \approx \frac{4}{\sigma}$$

we obtain $\sigma = 0.5$. This corresponds to placing the dominant poles at $s = -0.5$.

In this design, we use three real poles: a double pole at $s = -0.5$ and a third faster pole at $s = -2.5$. The resulting characteristic polynomial is:

$$(s + 0.5)^2(s + 2.5) = s^3 + 3.5s^2 + 2.75s + 0.625$$

This structure produces a smooth and overdamped response with minimal overshoot, as all poles are real and negative. While the rise time may be slower than that of complex poles, the settling time requirement is still satisfied due to the dominant pole real part $\sigma = 0.5$.

Additional Pole Placement Justification:
According to the project specification, any extra poles or zeros that may impair second-order behavior should be placed with real parts at least five times farther from the dominant poles. Since the dominant poles are located at $s = -0.5$, the additional pole was placed at $s = -2.5$, satisfying this requirement ($2.5 = 5 \times 0.5$). This placement ensures that the third pole does not significantly influence the transient dynamics dominated by the slower poles.

Conclusion: The use of three real poles provides a stable and smooth transient response while meeting the specified 8-second settling time.

Validation of Design Specifications

The designed PID controller was evaluated against the initial design requirements, which included:

- Settling time $T_s = 8$ seconds,
- Zero steady-state error for a step input,
- Use of a minimum-order controller structure.

Settling Time Requirement:
To satisfy the settling time criterion, a dominant pole placement method was applied. The desired dominant poles were selected as two real repeated poles at $s = -0.5$, along with a faster third pole at $s = -2.5$. The real part of the dominant poles $\sigma = 0.5$ corresponds to a theoretical settling time of:

$$T_s = \frac{4}{\sigma} = \frac{4}{0.5} = 8 \text{ seconds}$$

which exactly meets the design goal.

Steady-State Accuracy:
The designed open-loop transfer function has an integrator (a pole at the origin), which ensures that the closed-loop system is of Type 1. For a position control problem with a step input reference, this guarantees zero steady-state error:

$$\lim_{t \to \infty} [\theta_{ref}(t) - \theta_m(t)] = 0$$

Minimum Order Justification:
Initially, a PI controller was considered due to its lower order. However, controller tuning to meet the specified performance led to negative gain values, which are not physically realizable or stable in practice. Therefore, a PID controller — the next minimal structure — was used. The resulting third-order system was sufficient to satisfy the dynamic response and stability criteria without overcomplicating the design.

> *Hence, the final controller design using PID structure meets all given specifications in terms of transient response, steady-state accuracy, and simplicity.*

Note: The difficulty in achieving desired performance with a PI controller can also be attributed to the limited phase margin provided by the plant. Since the plant has no zero and includes an integrator, it provides relatively low phase lead near the crossover frequency. This makes it challenging to place the closed-loop poles effectively using a PI controller while maintaining all gain values positive.

5 Controller Implementation on Ideal System

Once the PID controller was designed using the calculated plant model, it was implemented on the ideal system — without any saturation constraints or external disturbance torque. This test allows us to observe the pure performance of the controller in ideal conditions.

Simulink Model – Ideal System

The controller was implemented in MATLAB Simulink using a simplified configuration without saturation or external disturbance. Figure 8 shows the structure of the simulation model.

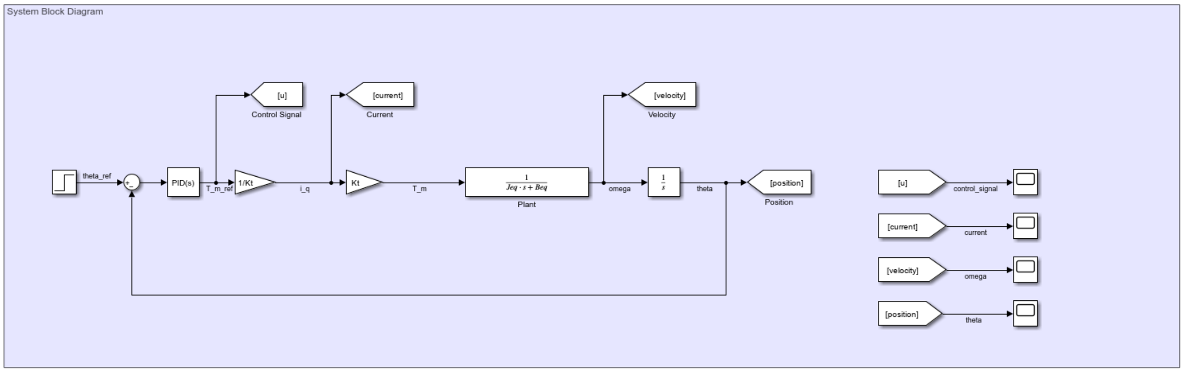

Figure 8: Simulink block diagram for the ideal PID-controlled PMSM system

Block Explanation:

- The PID controller block was implemented using the theoretically derived gains: $K_p = 2.75$, $K_i = 0.625$, and $K_d = 2.5$, based on pole placement for an 8-second settling time.
- The current controller is modeled assuming $i_d = 0$, with i_q directly controlling torque.
- The motor plant incorporates total equivalent inertia J_{eq} and damping B_{eq}, computed from the combined motor and load parameters.
- A unit step input is applied to the position reference $\theta_{ref}(t)$ to evaluate tracking performance.

This is justified for SPMSM motors, where the rotor's surface-mounted magnet geometry leads to negligible saliency (i.e., $L_d \approx L_q$), eliminating reluctance torque. As a result, torque generation depends linearly on i_q, and field weakening is typically not required under normal operating conditions. Therefore, direct torque control via i_q is both sufficient and effective.

Position Response

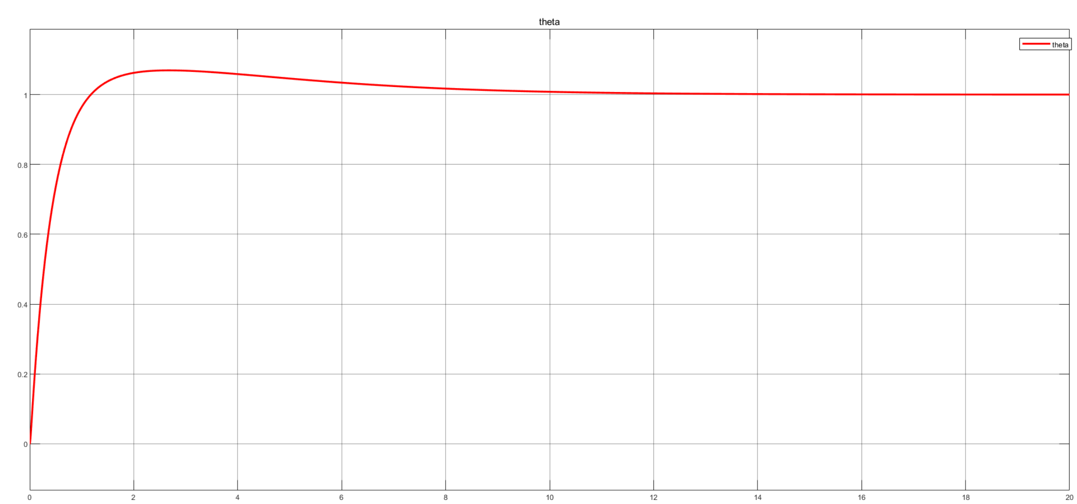

Figure 9: Case 1 – Position $\theta_m(t)$ under ideal conditions (no load, no saturation)

Observation:

- Settling time is approximately 8 seconds.
- Slight overshoot (7.57%) is observed.
- The steady-state error is zero.

Step Response Characteristics

The system response characteristics obtained from the MATLAB `stepinfo` function are summarized in Table 1. These confirm that the controller achieves the required transient performance.

Table 1: Step Response Characteristics

Parameter	**Value**
Rise Time	0.7625 s
Settling Time	7.5796 s
Overshoot	7.0197%
Peak	1.0702
Peak Time	2.6894 s
Steady-State Error	0

Current and Speed Response

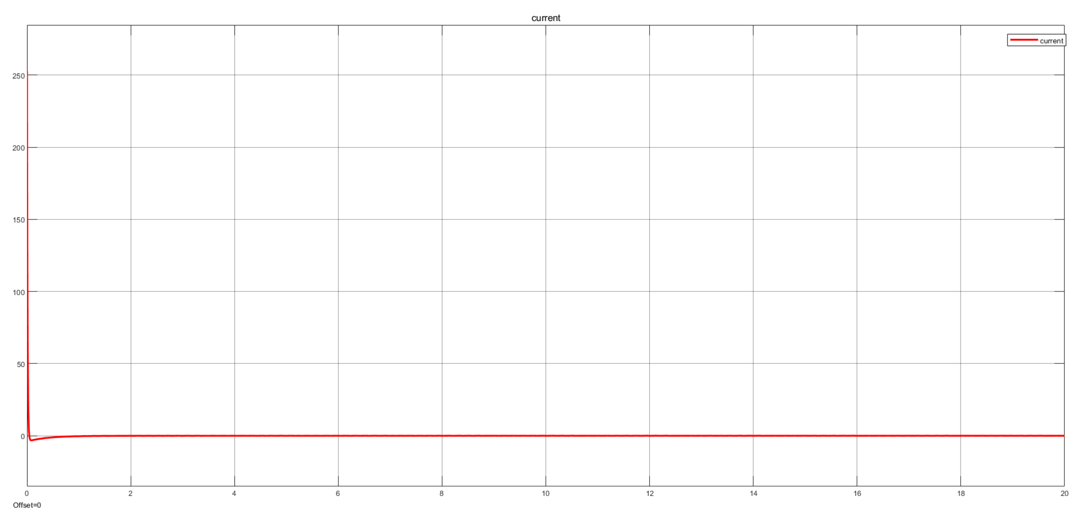

Figure 10: Case 1 – $i_q(t)$ (current) under ideal conditions

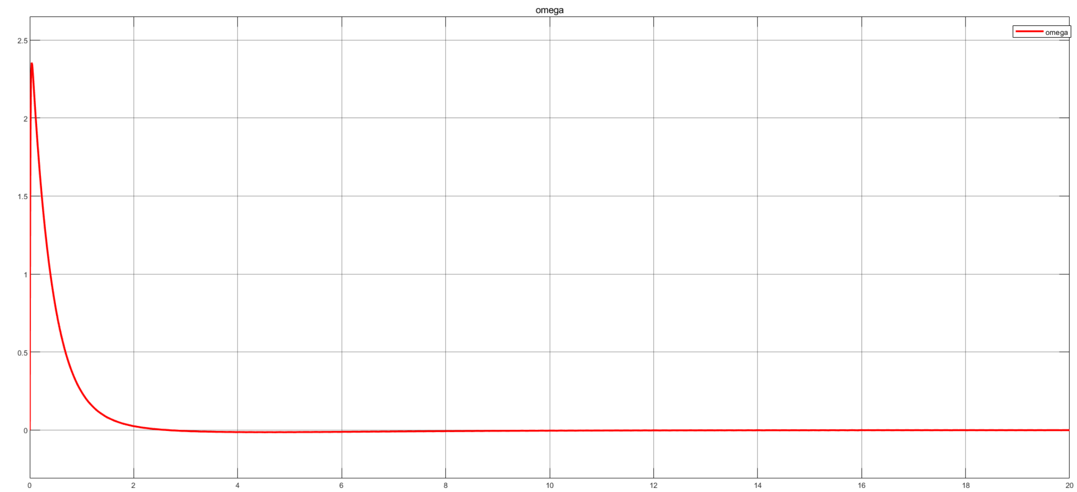

Figure 11: Case 1 – $\omega_m(t)$ (angular velocity) under ideal conditions

6 Why Compare Additional Scenarios?

Although the PID controller performs well in an ideal environment, real-world applications face two major challenges:

- **Actuator saturation:** The motor cannot generate infinite current or torque. Hardware limits (e.g., ± 20 A current, ± 5 Nm torque) must be enforced to protect the system.
- **External disturbances:** Periodic or unexpected torques due to variable loads (e.g., from belt tension or friction variations) can affect performance.

Thus, to validate the controller's robustness, four simulation scenarios are designed. The system was tested under the following configurations:

- **Case 1:** PID only (baseline).
- **Case 2:** PID + periodic disturbance torque: $T_L(t) = 0.5 \cdot \text{square}(0.125 \cdot t)$.
- **Case 3:** PID + actuator saturation (± 5 Nm torque, ± 20 A current).
- **Case 4:** PID + both disturbance and saturation.

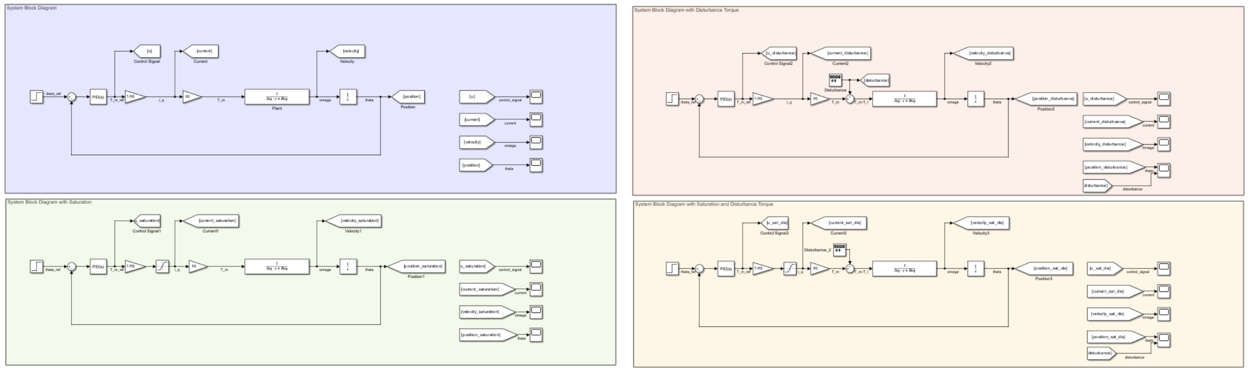

Figure 12: Simulink block diagrams representing each simulation scenario

7 Grouped Simulation Results and Comparative Analysis

In this section, grouped responses of position, current, and angular velocity are shown for each simulation case. This provides a visual and analytical comparison of the controller's performance under increasing complexity.

Position Response Comparison

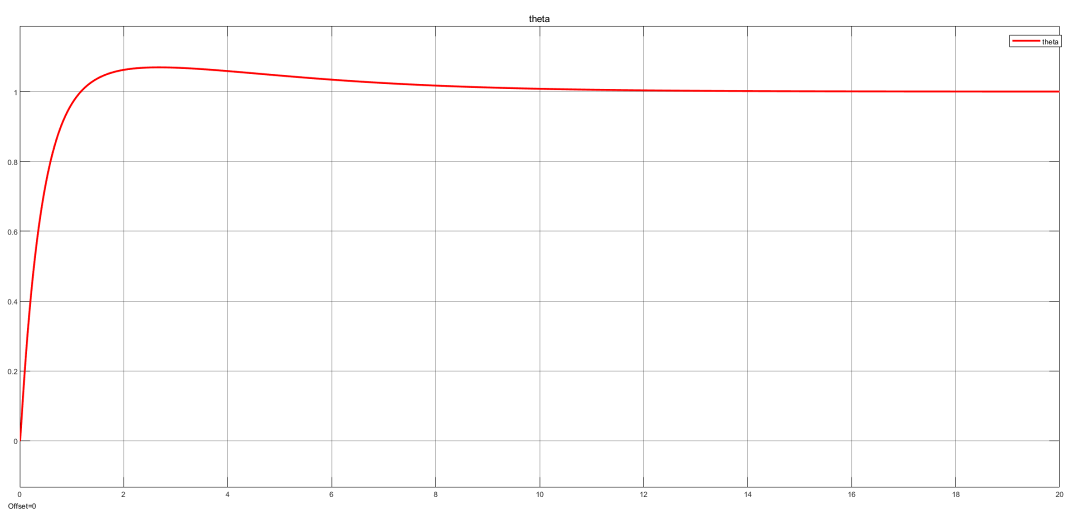

Figure 13: Case 1 – $\theta_m(t)$ without disturbance or saturation

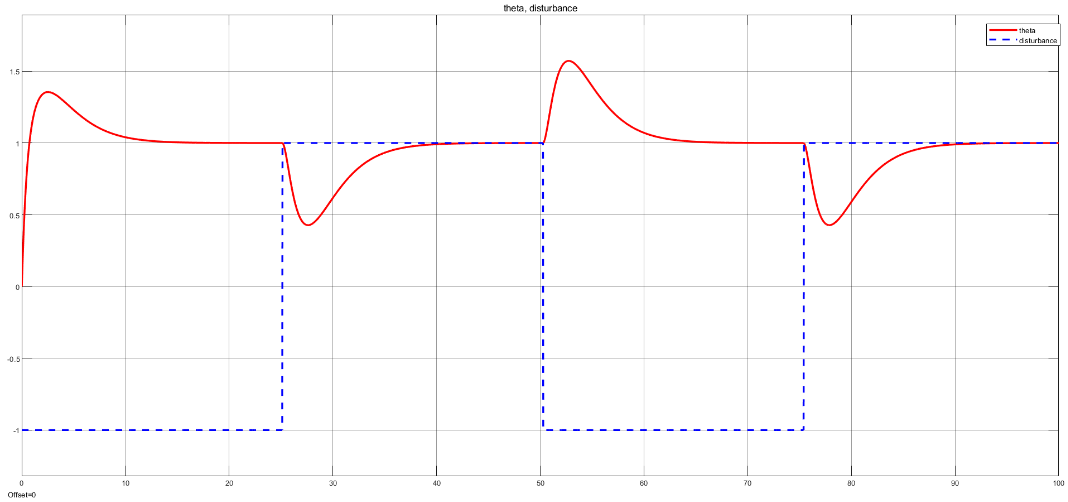

Figure 14: Case 2 – $\theta_m(t)$ with periodic square disturbance

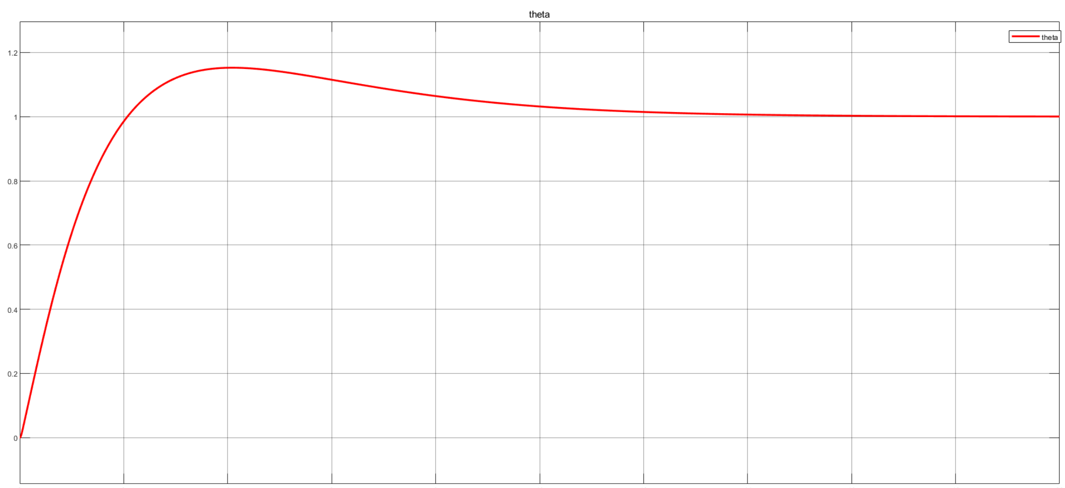

Figure 15: Case 3 – $\theta_m(t)$ with torque and current saturation

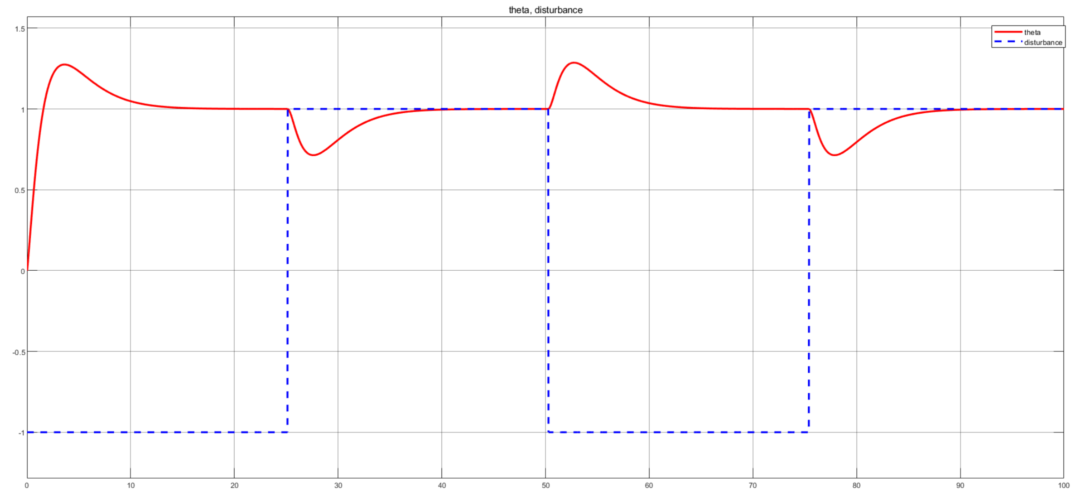

Figure 16: Case 4 – $\theta_m(t)$ with both saturation and disturbance

Analysis:

- Disturbance (Case 2) causes periodic ripples but controller maintains tracking.
- Saturation (Case 3) delays response slightly and reduces overshoot.
- Combined effects (Case 4) amplify ripples but stability is preserved.

Current Response Comparison (i_q)

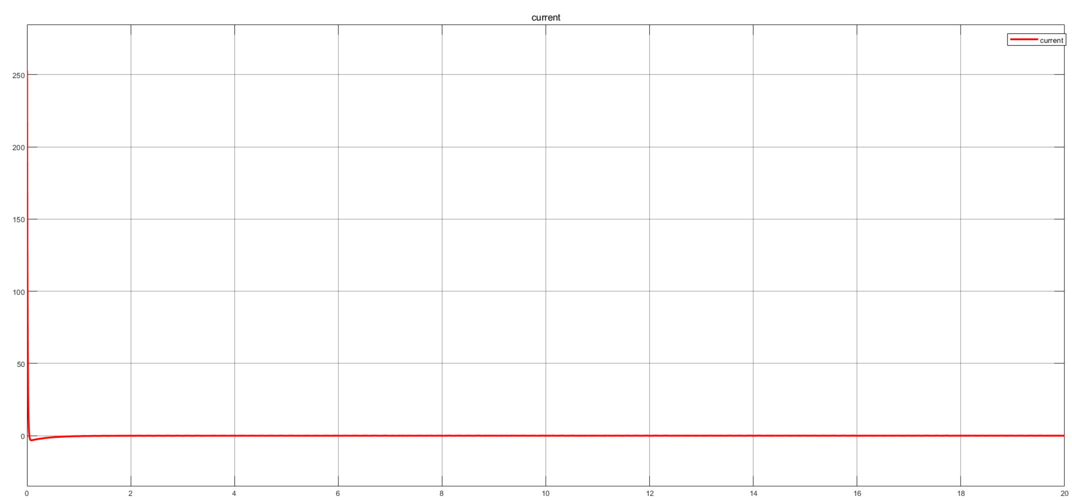

Figure 17: Case 1 – $i_q(t)$ ideal

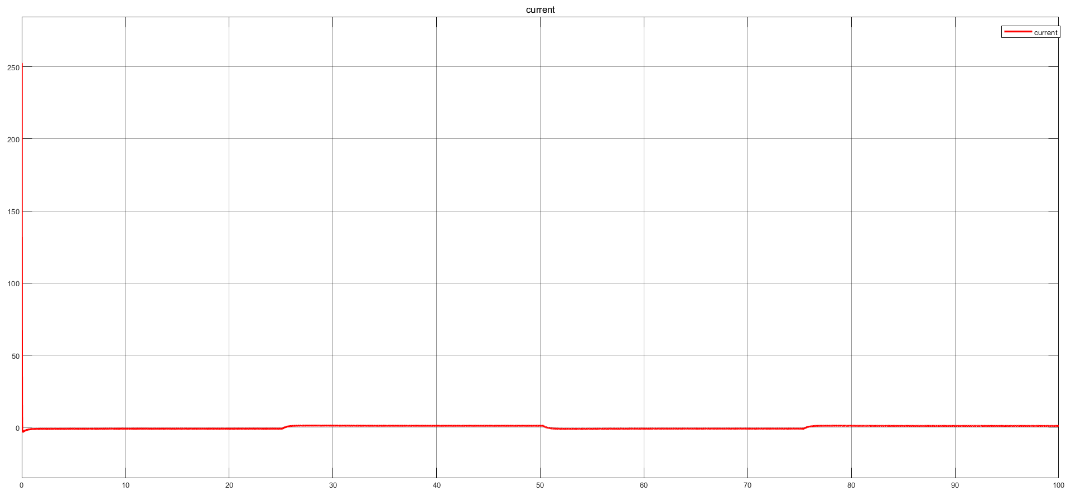

Figure 18: Case 2 – $i_q(t)$ with disturbance

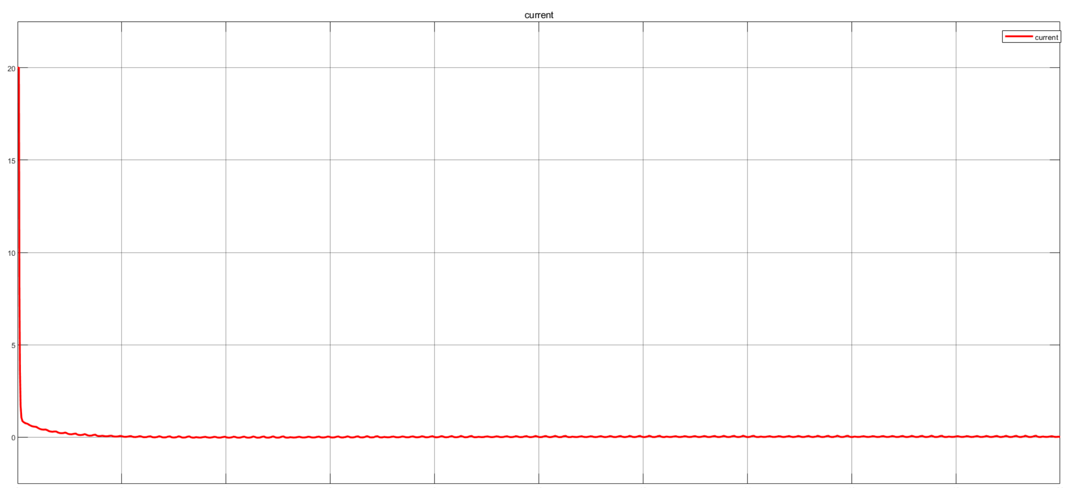

Figure 19: Case 3 – $i_q(t)$ with saturation

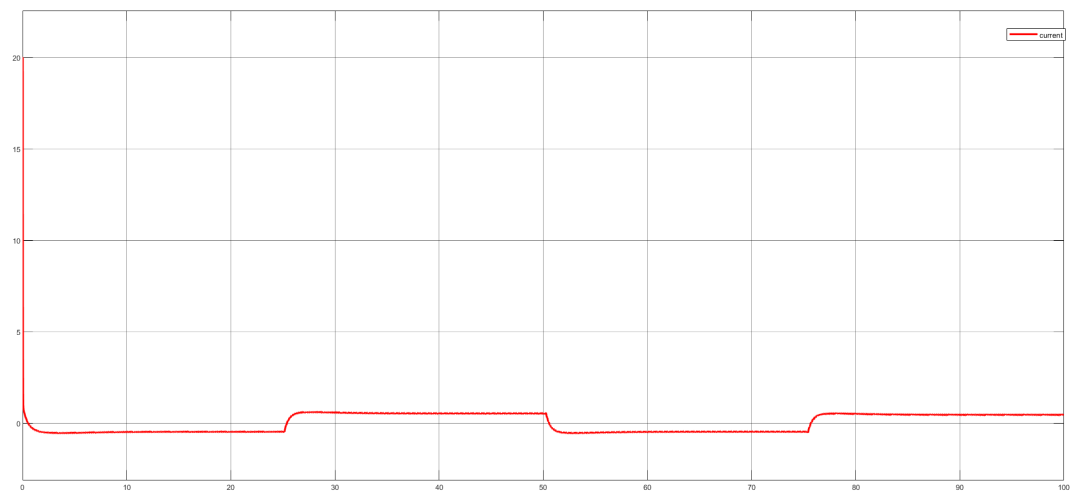

Figure 20: Case 4 – $i_q(t)$ with disturbance and saturation

Analysis:

- Ideal and disturbance cases exceed physical current limits.
- Saturation effectively bounds $i_q(t)$ to safe levels.
- Disturbance in saturated systems leads to current ripple at boundary.

Angular Velocity (ω_m) Comparison

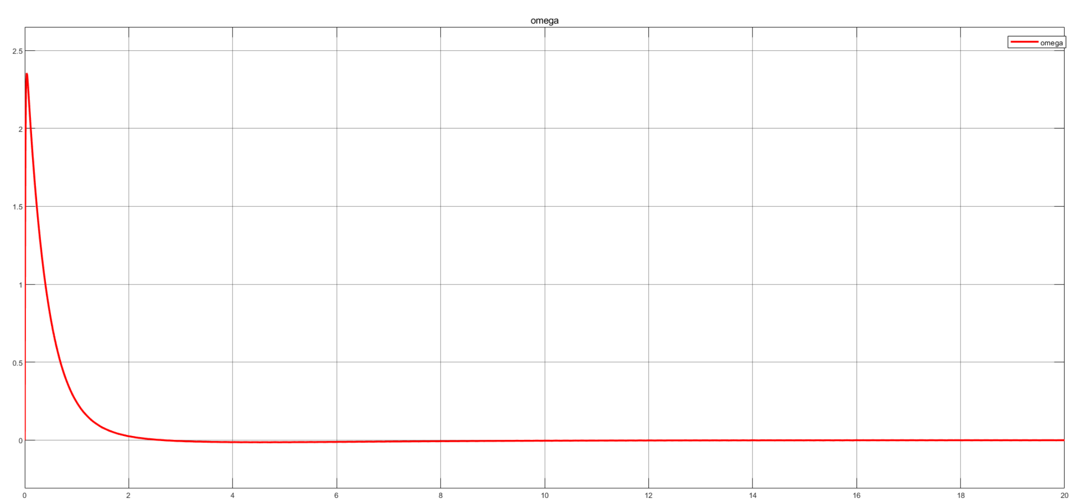

Figure 21: Case 1 – Angular speed $\omega_m(t)$ ideal

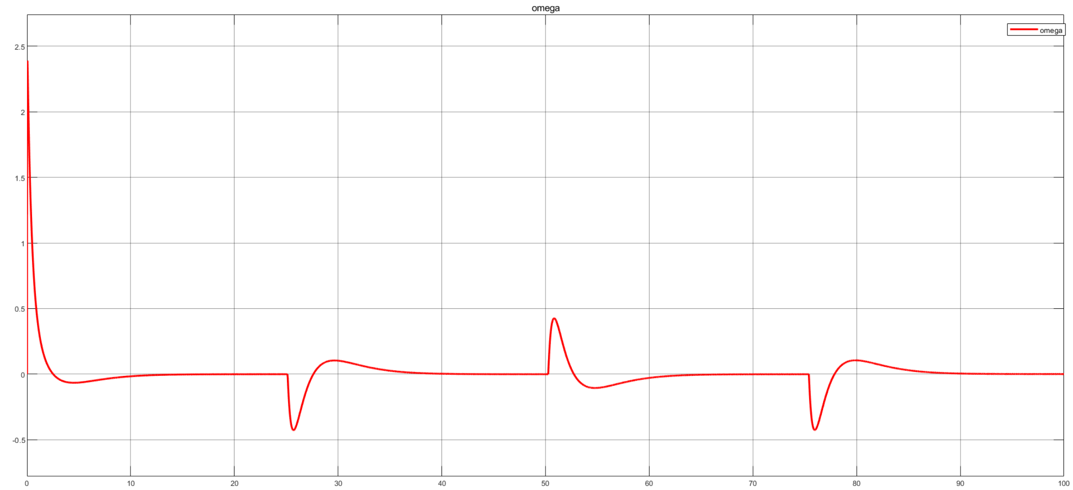

Figure 22: Case 2 – $\omega_m(t)$ with disturbance

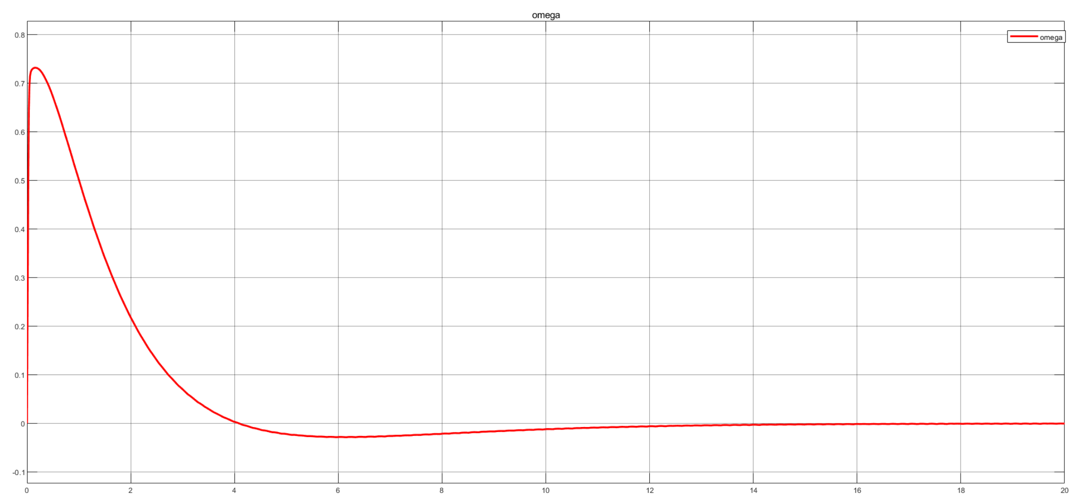

Figure 23: Case 3 – $\omega_m(t)$ with saturation

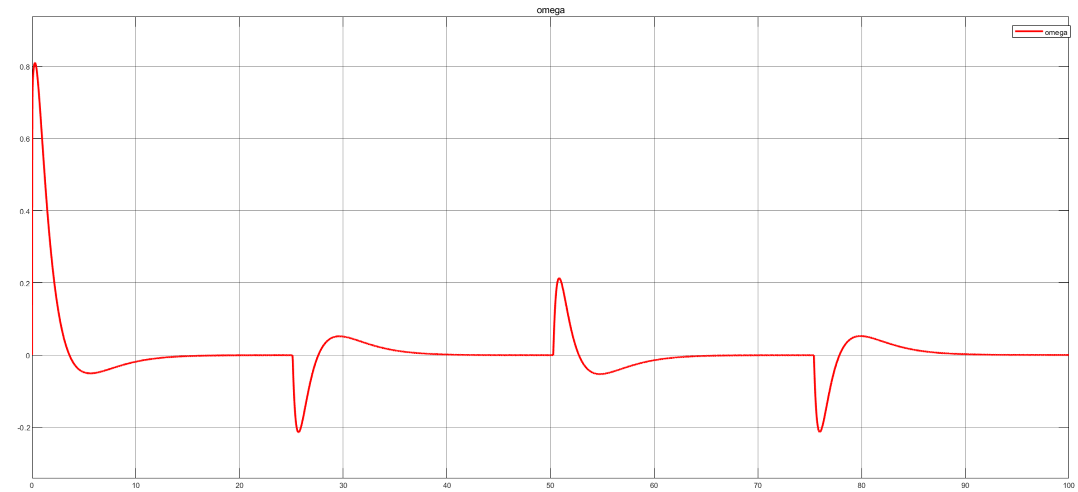

Figure 24: Case 4 – $\omega_m(t)$ with saturation and disturbance

Analysis:

- Angular speed mirrors changes in torque/current.
- Ripple increases under disturbance.
- Saturation smooths peaks but can slightly reduce responsiveness.

8 Conclusion

In this study, a PID controller was designed and implemented for a Permanent Magnet Synchronous Motor (PMSM) position control system. The specific motor type used in this work is a Surface-mounted PMSM (SPMSM), which assumes negligible saliency and enables the simplification $i_d = 0$. This assumption simplifies control implementation and is valid for SPMSM due to the symmetric rotor structure.

The primary design goals were achieving a settling time of $T_s = 8$ seconds and ensuring zero steady-state error for a step input. The design was validated through both theoretical pole-placement analysis and multiple simulation scenarios.

Key findings are summarized as follows:

- **Ideal Case:** The controller achieves the target performance under ideal conditions, confirming the correctness of the closed-loop pole placement and gain selection.
- **Disturbance Handling:** Under periodic torque disturbances, the system maintains tracking with only minor oscillations, indicating good disturbance rejection.
- **Saturation Effects:** When torque and current limits are imposed, the controller respects the physical bounds and maintains stable operation, though with slightly reduced dynamic performance.

- **Worst-Case Scenario:** Even under combined disturbance and saturation conditions, the system remains stable and functional, validating the robustness of the design.

Overall, the PID controller design is successful in meeting all specified performance requirements for SPMSM-based position control. For future work, control performance under parameter uncertainties can be improved using adaptive or robust control strategies. Additionally, anti-windup mechanisms can be implemented to further enhance the response under saturation.

In addition to the primary control performance, simulations were implemented in MATLAB Simulink with realistic conditions including sampling constraints and actuator limits. The use of a step reference signal provided a clear benchmark for evaluating transient and steady-state performance. An initial attempt with a PI controller was found insufficient due to non-physical gain values, further supporting the necessity of a full PID controller. These results affirm that the SPMSM control system not only meets design specifications but also maintains robustness under practical limitations.

In this revised design, three real poles were selected instead of complex conjugate pairs to achieve an overdamped but smooth response, while still satisfying the 8-second settling time requirement.

References

[1] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[2] M. Gökaşan, *Servo Motors Chapter 7 - PMSM*, Servo Motors Lecture Notes, 2025.

Appendix

A.1 Simulation Parameters

- Simulation Time: 20 seconds (ideal case), 100 seconds (cases with disturbance)
- Sampling Time: 0.001 seconds
- Reference Input: Step to 1 at $t = 0$
- Disturbance Torque: $T_L(t) = 0.5 \cdot \text{square}(0.125 \cdot t)$ (Case 2 & 4)
- Saturation Limits: ±5 Nm torque, ±20 A current (Case 3 & 4)

A.2 Controller Parameters

- PID Gains:
  - $K_p = 2.75$
  - $K_i = 0.625$
  - $K_d = 2.5$

A.3 MATLAB Simulink Notes

- The Simulink model uses blocks for PID Controller, Plant Dynamics (1/s² + s), Step Input, Saturation, and Disturbance Injection.
- The plant dynamics are modeled using the transfer function $P(s) = \frac{1}{J_{eq}s^2+B_{eq}s}$.
- The current controller is assumed ideal ($i_q = i_q^{ref}$).
- Scope blocks are used to collect $\theta_m(t)$, $i_q(t)$, and $\omega_m(t)$.

A.4 MATLAB Code

Listing 1: PID Controller Simulation Code

```
% ==========================================================
% KON417E Project - PMSM Position Control with PID Controller
% ==========================================================
clc; clear;
syms s;

%% ---------------- Motor Parameters ----------------
Kt = 1;          % Torque constant (Nm/A)
Kb = 1;          % Back EMF constant (Vs/rad)
```

```matlab
M  = 2.4;       % Load mass (kg)
B  = 0.8;       % Load damping (Ns/m)
r  = 0.5;       % Pulley radius (m)
Jt = 0.2;       % Drum inertia (kg*m^2)
Bt = 0.6;       % Drum damping (Nm*s/rad)
Jm = 0.2;       % Motor inertia (kg*m^2)
Bm = 0.2;       % Motor damping (Nm*s/rad)

%% ------------------------- Equivalent Inertia and Damping -------------------------
Jeq = Jm + Jt + M*r^2;
Beq = Bm + Bt + B*r^2;

%% ------------------------- Plant Transfer Function -------------------------
num_P = 1;
den_P = [Jeq, Beq, 0];
P = tf(num_P, den_P);  % Open-loop plant transfer function

%% ------------------------- PID Controller Design -------------------------
Kp = 2.75;
Ki = 0.625;
Kd = 2.5;
C = pid(Kp, Ki, Kd);  % PID controller object

%% ------------------------- Closed-loop System -------------------------
T_closed = feedback(C * P, 1);

%% ------------------------- Step Response -------------------------
figure;
step(T_closed, 20);
title('Step Response of Closed-Loop System with PID Controller');
xlabel('Time (s)');
ylabel('Position \theta_m (rad)');
grid on;

%% ------------------------- Step Info -------------------------
info = stepinfo(T_closed)

%% ------------------------- Pole-Zero Map -------------------------
figure;
pzmap(T_closed);
title('Pole-Zero Map of the Closed-Loop System');
grid on;
```