CONTEMPORARY METHODOLOGICAL APPROACHES IN ELECTRICAL AND ELECTRONICS ENGINEERING

EDITOR

PROF. KENAN DÖŞOĞLU, PH.D.

DOI:10.5281/zenodo.19165662

Contemporary Methodological Approaches in Electrical and Electronics Engineering

Edıtor

Prof. Kenan Döşoğlu, Ph.D.

Publisher
Platanus Publishing®

Editor in Chief
Prof. Kenan Döşoğlu, Ph.D.

Cover & Interior Design
Platanus Publishing®

The First Edition
March, 2026

ISBN
978-625-8513-85-7

Platanus Publishing®
Address: Natoyolu Cad. Fahri Korutürk Mah. 157/B, 06480, Mamak,
Ankara, Turkey.

Phone: +90 312 390 1 118
web: www.platanuspublishing.com
e-mail: platanuskitap@gmail.com

Platanus Publishing®

CONTENTS

CHAPTER 1

An Adaptive V/F Control Approach for Three-Phase Induction Motor Speed Control Using Sliding Mode Techniques

Seyfettin Vadi[1] *& Ramazan Bayındır*[2]

INTRODUCTION

Three-phase induction motors are among the most widely used electrical machines in modern industrial automation systems. Owing to their robust construction, low cost, minimal maintenance requirements, and stable operation over a wide power range, these motors are extensively preferred in industrial applications. However, from the perspective of speed and torque control, induction motors present several challenges due to their inherently nonlinear and dynamically complex characteristics (Bayramoglu and Kömürcügil, 2012).

One of the most commonly employed methods for speed control of induction motors is the Voltage/Frequency (V/F) control technique, which maintains a constant stator magnetic flux to ensure stable torque production. Nevertheless, conventional V/F control schemes operate with a fixed voltage-to-frequency ratio and therefore exhibit limited dynamic performance, particularly under varying load conditions and changing reference speeds Biricik, Kömürcügil and Basu, 2015).

To overcome the limitations of classical V/F control, this chapter proposes a speed control approach based on Sliding Mode Control (SMC). Sliding mode control is a modern control technique distinguished by its strong robustness against model uncertainties and external disturbances Cömert and Kasnakoğlu, 2016). In the proposed framework, a sliding surface is defined as a function of the motor speed error and its derivative, and the resulting control signal dynamically determines the voltage amplitude applied to the motor windings, i.e., the adaptive V/F ratio (K). This adaptive mechanism enables the system to respond to variable reference commands and sudden load disturbances in a stable and reliable manner (Cömert and Kasnakoğlu, 2017).

Furthermore, the chattering phenomenon, commonly encountered in classical sliding mode control implementations and potentially harmful to physical

[1] Assist. Prof. Dr., Gazi Unıversity, Faculty of Technology, Department of Electrical - Electronic Engineering, ORCID: 0000-0002-4244-9573

[2] Prof. Dr., Gazi Unıversity, Faculty of Technology, Department of Electrical - Electronic Engineering, ORCID: 0000-0001-6424-0343

systems, is mitigated in this study by using a continuous saturation function Efe and Kasnakoğlu, 2008). By introducing a boundary layer around the sliding surface, high-frequency components of the control signal are effectively suppressed, resulting in smoother control action and improved practical applicability (Kürkçü and Kasnakoğlu, 2015).

The proposed control system is modeled in the MATLAB/Simulink environment and evaluated through simulation studies under various operating conditions, including different reference speed transitions (e.g., 1300 rpm to 1700 rpm) and load torque variations (e.g., 50 Nm to 175 Nm). The simulation results demonstrate that the sliding mode control–based strategy achieves accurate reference speed tracking under both steady-state and transient conditions, ensures rapid convergence to steady-state operation, and provides high robustness against load disturbances.

STATE-SPACE REPRESENTATION OF THE THREE-PHASE INDUCTION MOTOR

2.1 CLARK-PARK TRANSFORMATIONS

To simplify analysis of the three-phase system, the motor variables are transformed into a stationary reference frame. Accordingly, Clarke and Park transformations are applied to obtain the d–q axis representation of the induction motor. The Clarke transformation maps the three-phase quantities to a stationary stator reference frame, while the Park transformation converts them to a rotating reference frame, thereby enhancing the applicability of control algorithms. Through these transformations, the vector equations of the induction motor are linearized, allowing the system to be analyzed in state-space form (Nurettin, and Inanç, 2022).

In a three-phase system, the phase currents are expressed as:

$$i_a(t) = I_m \cos(\omega t) \tag{1}$$

$$i_b(t) = I_m cos\,(\omega t - \frac{2\pi}{3}) \tag{2}$$

$$i_c(t) = I_m cos\,(\omega t + \frac{2\pi}{3}) \tag{3}$$

Since these currents are time-varying, their direct control is inherently complex. Therefore, coordinate transformations, such as the Clarke and Park transformations, are employed to convert the three-phase time-varying quantities into equivalent components in stationary and rotating reference frames, thereby significantly simplifying control design and analysis (Shtessel, Edwards, Fridman and Levant, 2014).

2.1.1 Clarke Transformation

The Clarke transformation maps the three-phase a–b–c signals onto the two-dimensional stationary stator reference frame defined by the α–β axes. This transformation enables the representation of three-phase phasors in a fixed two-dimensional plane, thereby simplifying the analysis and control of the system (Yaylacı, 2011).

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \qquad (4)$$

$$i_\alpha = \frac{2}{3}(i_a - \frac{1}{2} i_b - \frac{1}{2} i_c) \qquad (5)$$

$$i_\beta = \frac{2}{3}(\frac{\sqrt{3}}{2} i_b - \frac{\sqrt{3}}{2} i_c) \qquad (6)$$

2.1.2 Park Transformation

The Park transformation provides the transition from the stationary α–β reference frame to the rotating d–q reference frame. As a result, sinusoidal quantities are converted to constant (DC) values, which significantly facilitate the design and implementation of control algorithms (Young, Uzguner and Hedrick, 1999).

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} \begin{bmatrix} \cos{(\theta_e)} & sin(\theta_e) \\ -sin(\theta_e) & cos(\theta_e) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \qquad (7)$$

$$i_d = i_\alpha \cos(\theta_e) + i_\beta sin\,(\theta_e) \qquad (8)$$

$$i_q = -i_\alpha \sin(\theta_e) + i_\beta cos\,(\theta_e) \qquad (9)$$

$$\theta_e = Electrical\ angle$$

$$i_d = Flux\ component$$

$$i_q = Torque\ component$$

As a result of these transformations, the motor currents are represented in a fixed reference frame, whereby the originally sinusoidal quantities are reduced to constant values. Consequently, the system attains a quasi-linear structure, suitable for the application of various control algorithms, such as Sliding Mode Control (SMC), Linear Quadratic Regulator (LQR), and Proportional–Integral–Derivative (PID) control (Young, Uzguner and Hedrick, 1999).

In the stator d–q reference frame:

$$v_{ds} = R_s i_{ds} + \frac{d\psi_{ds}}{dt} - \omega_e \psi_{qs} \qquad (10)$$

$$v_{qs} = R_s i_{qs} + \frac{d\psi_{qs}}{dt} + \omega_e \psi_{ds} \qquad (11)$$

Rotor in the d-q axis:

$$0 = R_r i_{dr} + \frac{d\psi_{dr}}{dt} - (\omega_e - \omega_r)\psi_{qr} \qquad (12)$$

$$0 = R_r i_{qr} + \frac{d\psi_{qr}}{dt} + (\omega_e - \omega_r)\psi_{dr} \qquad (13)$$

(the rotor tips are short-circuited, $v_{dr} = v_{qr} = 0$)

2.2 Flux–Current Transformations:

As the state variables in the state-space formulation are expressed in current form, the corresponding flux-to-current transformations are derived.

$$\psi_{ds} = L_s i_{ds} + L_m i_{dr} \qquad (14)$$

$$\psi_{qs} = L_s i_{qs} + L_m i_{qr} \qquad (15)$$

$$\psi_{dr} = L_r i_{dr} + L_m i_{ds} \qquad (16)$$

$$\psi_{qr} = L_r i_{qr} + L_m i_{qs} \qquad (17)$$

$$\frac{d\psi_{ds}}{dt} = L_s \frac{di_{ds}}{dt} + L_m \frac{di_{dr}}{dt} \qquad (18)$$

$$\frac{d\psi_{qs}}{dt} = L_s \frac{di_{qs}}{dt} + L_m \frac{di_{qr}}{dt} \qquad (19)$$

$$\frac{d\psi_{dr}}{dt} = L_r \frac{di_{dr}}{dt} + L_m \frac{di_{ds}}{dt} \qquad (20)$$

$$\frac{d\psi_{qr}}{dt} = L_r \frac{di_{qr}}{dt} + L_m \frac{di_{qs}}{dt} \qquad (21)$$

2.3 Stator d-axis voltage equation

Initial state equation:

$$v_{ds} = R_s i_{ds} + \frac{d\psi_{ds}}{dt} - \omega_e \psi_{qs} \qquad (22)$$

Equation (22) is reformulated by substituting the corresponding derivative and flux expressions.

$$v_{ds} = R_s i_{ds} + (L_s \frac{di_{ds}}{dt} + L_m \frac{di_{dr}}{dt}) - \omega_e (L_s i_{qs} + L_m i_{qr}) \tag{23}$$

$$v_{ds} = R_s i_{ds} + L_s \frac{di_{ds}}{dt} + L_m \frac{di_{dr}}{dt} - \omega_e L_s i_{qs} - \omega_e L_m i_{qr} \tag{24}$$

$$L_s \frac{di_{ds}}{dt} + L_m \frac{di_{dr}}{dt} = v_{ds} - R_s i_{ds} + \omega_e L_s i_{qs} + \omega_e L_m i_{qr} \tag{25}$$

$$\frac{di_{ds}}{dt} = \frac{1}{L_s} (v_{ds} - R_s i_{ds} + \omega_e L_s i_{qs} + \omega_e L_m i_{qr} - L_m \frac{di_{dr}}{dt}) \tag{26}$$

$$\dot{x_1} = \frac{1}{L_s} (v_{ds} - R_s x_1 + \omega_e L_s x_2 + \omega_e L_m x_4 - L_m \dot{x}_3) \tag{27}$$

By using Eqs. (23)–(27), the time derivative of the state variable i_{ds} is obtained.

2.4 Stator q-axis voltage equation

Initial state equation:

$$v_{qs} = R_s i_{qs} + \frac{d\psi_{qs}}{dt} + \omega_e \psi_{ds} \tag{28}$$

Equation (28) is reformulated by substituting the corresponding derivative and flux expressions.

$$v_{qs} = R_s i_{qs} + \left(L_s \frac{di_{qs}}{dt} + L_m \frac{di_{qr}}{dt}\right) + \omega_e (L_s i_{ds} + L_m i_{dr}) \tag{29}$$

$$v_{qs} = R_s i_{qs} + L_s \frac{di_{qs}}{dt} + L_m \frac{di_{qr}}{dt} + \omega_e L_s i_{ds} + \omega_e L_m i_{dr} \tag{30}$$

$$L_s \frac{di_{qs}}{dt} + L_m \frac{di_{qr}}{dt} = v_{qs} - R_s i_{qs} - \omega_e L_s i_{ds} - \omega_e L_m i_{dr} \tag{31}$$

$$\frac{di_{qs}}{dt} = \frac{1}{L_s} (v_{qs} - R_s i_{qs} - \omega_e L_s i_{ds} - \omega_e L_m i_{dr} - L_m \frac{di_{qr}}{dt}) \tag{32}$$

$$\dot{x_2} = \frac{1}{L_s} (v_{qs} - R_s x_2 - \omega_e L_s x_1 - \omega_e L_m x_3 - L_m \dot{x_4}) \tag{33}$$

By using Eqs. (29)–(33), the time derivative of the state variable $\boldsymbol{i_{qs}}$ is obtained.

2.5 Rotor d-axis voltage equation:

Initial state equation:

$$0 = R_r i_{dr} + \frac{d\psi_{dr}}{dt} - (\omega_e - \omega_r)\psi_{qr} \qquad (34)$$

Equation (34) is reformulated by substituting the corresponding derivative and flux expressions.

$$0 = R_r i_{dr} + (L_r \frac{di_{dr}}{dt} + L_m \frac{di_{ds}}{dt}) - (\omega_e - \omega_r)L_r i_{qr} + L_m i_{qs} \qquad (35)$$

$$0 = R_r i_{dr} + L_r \frac{di_{dr}}{dt} + L_m \frac{di_{ds}}{dt} - (\omega_e - \omega_r)L_r i_{qr} - (\omega_e - \omega_r)L_m i_{qs} \qquad (36)$$

$$L_r \frac{di_{dr}}{dt} + L_m \frac{di_{ds}}{dt} = -R_r i_{dr} + (\omega_e - \omega_r)L_r i_{qr} + (\omega_e - \omega_r)L_m i_{qs} \qquad (37)$$

$$\frac{di_{dr}}{dt} = \frac{1}{L_r}(-R_r i_{dr} + (\omega_e - \omega_r)L_r i_{qr} + (\omega_e - \omega_r)L_m i_{qs} - L_m \frac{di_{ds}}{dt}) \qquad (38)$$

$$\dot{x_3} = \frac{1}{L_r}(-R_r x_3 + (\omega_e - x_5)L_r x_4 + (\omega_e - x_5)L_m x_2 - L_m \dot{x_1}) \quad (39)$$

By using Eqs. (35)–(39), the time derivative of the state variable i_{dr} is obtained

2.6 Rotor q-axis voltage equation

Initial state equation:

$$0 = R_r i_{qr} + \frac{d\psi_{qr}}{dt} + (\omega_e - \omega_r)\psi_{dr} \qquad (40)$$

Equation (40) is reformulated by substituting the corresponding derivative and flux expressions.

$$0 = R_r i_{qr} + L_r \frac{di_{qr}}{dt} + L_m \frac{di_{qs}}{dt} + (\omega_e - \omega_r)L_r i_{dr} + L_m i_{ds} \qquad (41)$$

$$0 = R_r i_{qr} + L_r \frac{di_{qr}}{dt} + L_m \frac{di_{qs}}{dt} + (\omega_e - \omega_r)L_r i_{dr} + +(\omega_e - \omega_r)L_m i_{ds} \qquad (42)$$

$$L_r \frac{di_{qr}}{dt} + L_m \frac{di_{qs}}{dt} = -R_r i_{qr} - (\omega_e - \omega_r)L_r i_{dr} - (\omega_e - \omega_r)L_m i_{ds} \qquad (43)$$

$$\frac{di_{qr}}{dt} = \frac{1}{L_r}(-R_r i_{qr} - (\omega_e - \omega_r)L_r i_{dr} - (\omega_e - \omega_r)L_m i_{ds} - L_m \frac{di_{qs}}{dt}) \qquad (44)$$

$\dot{x_4} = \frac{1}{L_r}(-R_r x_4 - (\omega_e - x_5)L_r x_3 - (\omega_e - x_5)L_m x_1 - L_m \dot{x_2})$ (45)

By using Eqs. (41)–(45), the time derivative of the state variable $\imath_{qr}$ is obtained.

The differential equations obtained in the d–q reference frame are arranged to construct the state-space model. However, since some of the equations contain derivatives of other state variables on their right-hand sides (e.g., $\dot{\imath_{qr}}$), the system becomes cross-coupled.

For this reason, the classical state-space form $\dot{x} = Ax + Bu$ cannot be applied directly. Instead, the system is reformulated as $M\dot{x} = Nx + Bu$,

where the cross-coupling is eliminated through the matrix M. As a result of this transformation, the system is linearized, and the state-space matrices A and B are obtained from $\dot{x} = M^{-1}Nx + M^{-1}Bu$.

$$M\dot{x} = Nx + B \quad (46)$$

By means of Eq. (46), the cross-derivative terms are transferred to the matrix M.

$$M\dot{x} = Nx + Bu \quad (47)$$

$$\dot{x} = M^{-1}Nx + M^{-1}Bu \quad (48)$$

$$A = M^{-1}N \quad (49)$$

In the stator d-axis equation

$$L_s\dot{\imath}_{ds} + L_m\dot{\imath}_{dr} = v_{ds} - R_s i_{ds} + \omega_e L_s i_{qs} + \omega_e L_m i_{qr} \quad (50)$$

$$[M]_{1,:} = [L_s, 0, L_m, 0] \quad (51)$$

In the stator q-axis equation

$$L_s\dot{\imath}_{qs} + L_m\dot{\imath}_{qr} = v_{qs} - R_s\imath_{qs} - \dot{}\omega_e L_s\imath_{ds} - \omega_e L_m\imath_{dr} \quad (52)$$

$$[M]_{2,:} = [0, L_s, 0, L_m] \quad (53)$$

In the rotor d-axis equation

$$L_r\dot{\imath}_{dr} + L_m\dot{\imath}_{ds} = -R_r\imath_{dr} + (\omega_e - \omega_r\dot{)}L_r\imath_{qr} + (\omega_e - \omega_r)L_m\imath_{qs} \quad (54)$$

$$[M]_{3,:} = [L_m, 0, L_r, 0] \quad (55)$$

In the rotor q-axis equation

$$L_r\dot{i}_{qr} + L_m\dot{i}_{qs} = -R_r i_{qr} - (\omega_e - \omega_r\dot{)}L_r i_{dr} - (\omega_e - \omega_r)L_m i_{ds} \tag{56}$$

$$[M]_{4,:} = [0, L_m, 0, L_r] \tag{57}$$

In Eqs. (50), (52), (54), and (56), the derivative terms on the left-hand side constitute the matrix M, while the right-hand side forms the matrix $Nx + Bu$.

$$M = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \tag{58}$$

$$\dot{x} = \begin{bmatrix} \dot{i}_{ds} \\ \dot{i}_{qs} \\ \dot{i}_{dr} \\ \dot{i}_{qr} \end{bmatrix} \tag{59}$$

$$Nx + Bu = \begin{bmatrix} v_{ds} - R_s i_{ds} + \omega_e L_s i_{qs} + \omega_e L_m i_{qr} \\ v_{qs} - R_s i_{qs} - \omega_e L_s i_{ds} - \omega_e L_m i_{dr} \\ -R_r i_{dr} + (\omega_e - \omega_r)L_r i_{qr} + (\omega_e - \omega_r)L_m i_{qs} \\ -R_r i_{qr} - (\omega_e - \omega_r)L_r i_{dr} - (\omega_e - \omega_r)L_m i_{ds} \end{bmatrix} \tag{60}$$

Since the system is cross-coupled—owing to the presence of derivative terms of other state variables in some equations—the matrix M is employed. Consequently, the matrix M is non-diagonal.

2.7 MECHANICAL EQUATIONS

2.7.1 Torque Equation

The electromagnetic torque expression is obtained in the form $i_{dr}i_{qs} - i_{qr}i_{ds}$. This expression is directly combined with the rotor speed equation, resulting in the rotor speed dynamics, which constitutes the fifth differential equation of the system, namely the rotor speed derivative ($\dot{\omega_r}$).

However, since this equation involves products of state variables (e.g., $x_3x_2 - x_4x_1$), it is inherently nonlinear. Therefore, for the sake of control design and analytical tractability, the rotor speed equation is linearized around a selected operating point, and its effect is incorporated into the state-space model as a constant coefficient within the matrix A.

$$T_e = \frac{3}{2}P(\psi_{ds}i_{qs} - \psi_{qs}i_{ds}) \quad (61)$$

$$T_e = \frac{3}{2}P[(L_s i_{ds} + L_m i_{dr})i_{qs} - \left(L_s i_{qs} + L_m i_{qr}\right)i_{ds}] \quad (62)$$

$$T_e = \frac{3}{2}P(L_s i_{ds}i_{qs} + L_m i_{dr}i_{qs} - L_s i_{qs}i_{ds} - L_m i_{qr}i_{ds}) \quad (63)$$

Since the terms $L_s i_{ds} i_{qs}\ ve\ L_s i_{qs} i_{ds}$ in Eq. (63) are equal in magnitude and opposite in sign, they cancel each other out, leading to Eq. (64).

$$L_s i_{ds}i_{qs} - L_s i_{qs}i_{ds} = 0 \quad (64)$$

$$T_e = \frac{3}{2}PL_m(i_{dr}i_{qs} - i_{qr}i_{ds}) \quad (65)$$

$$T_e = \frac{3}{2}PL_m(x_3x_2 - x_4x_1) \quad (66)$$

2.7.2 Mechanical Equation of the Motor

$$J\frac{d\omega_r}{dt} = T_e - T_L - Bw_r \quad (67)$$

In the state variable $x_5 = \omega_r$ (68)

$$\dot{x_5} = \frac{1}{J}(\frac{3}{2}PL_m(x_3x_2 - x_4x_1) - T_L - Bx_5) \quad (69)$$

Since the term $PL_m(x_3x_2 - x_4x_1)$ in Eq. (69) is nonlinear, a linearization is performed by assuming the operating point as $x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 0$.

$$\frac{\partial \dot{x_5}}{\partial x_1} = \frac{3}{2}PL_m x_4 \quad (70)$$

2.8 State-Space Model

$$\text{A} = \begin{bmatrix} -\frac{R_s}{\sigma L_s} & \omega_e & 0 & -\frac{\omega_e L_m}{\sigma L_s} & 0 \\ -\omega_e & -\frac{R_s}{\sigma L_s} & \frac{\omega_e L_m}{\sigma L_s} & 0 & 0 \\ 0 & -\frac{\omega_r L_m}{\sigma L_r} & -\frac{R_r}{\sigma L_r} & (\omega_e - \omega_r) & 0 \\ \frac{\omega_r L_m}{\sigma L_r} & 0 & -(\omega_e - \omega_r) & -\frac{R_r}{\sigma L_r} & 0 \\ 0 & 0 & 0 & \frac{3}{2}\frac{P}{J}L_m & -\frac{B}{J} \end{bmatrix} \quad (71)$$

$$B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{72}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{73}$$

$$D = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{74}$$

Expressions in the Table

L_m: Mutual (magnetic coupling) inductance

L_s: Stator winding inductance

L_r: Rotor winding inductance

R_s: Stator winding resistance

R_r: Rotor winding resistance

ω_r: Actual angular speed of the rotor (rad/s)

ω_e: Angular speed of the synchronous reference frame (rad/s)

σ: Leakage coefficient, $\sigma = 1 - \frac{L_m^2}{L_s L_r}$

P : Number of pole pairs of the motor

L_m : Mutual (magnetic coupling) inductance

J : Moment of inertia

B : Viscous friction coefficient

State Variables:

$$x = \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \\ \omega_r \end{bmatrix} \tag{75}$$

Inputs:

$$u = \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix} \tag{76}$$

Output:

y = $[\omega_r]$ (77)

The complete set of derived equations enables the system to be represented in state-space form. This formulation allows modern control techniques to be applied directly to the system.

SLIDING MODE CONTROL

Sliding Mode Control (SMC) is a variable-structure control technique that enables stable control of nonlinear systems. The fundamental principle of sliding mode control is to force the system dynamics onto a predefined sliding surface, thereby ensuring the desired behavior. Once the system trajectory reaches the sliding surface, it remains constrained to move along this surface, and the error dynamics acquire specific stability properties.

Moreover, sliding mode control is a robust control approach designed to handle systems affected by measurement uncertainties and modeling inaccuracies, including unmodeled dynamics and parameter variations.

If the function f in the system model $\dot{x} = f(x, u)$ is not exactly known

Since controlling first-order dynamics is considerably simpler than controlling higher-order systems (n>1), a first-order dynamic structure, referred to as the sliding surface, is constructed within an n-th order system. A sliding condition is then established to ensure that the system trajectories reach this surface and remain on it. Subsequently, a control law is designed to satisfy these conditions.

The relationship between the system state and the sliding surface—specifically, the distance to the surface and the direction of motion—is characterized by first-order information.

Figure 3.1 illustrates a sliding surface (s=0) defined in the state-variable plane, along with the system trajectories converging toward this surface. ($s < 0$) and ($s > 0$), the control signal changes its direction to rapidly drive the system trajectories toward the sliding surface. Once the sliding surface is reached, the system motion is constrained to this surface, and the system subsequently evolves along it.

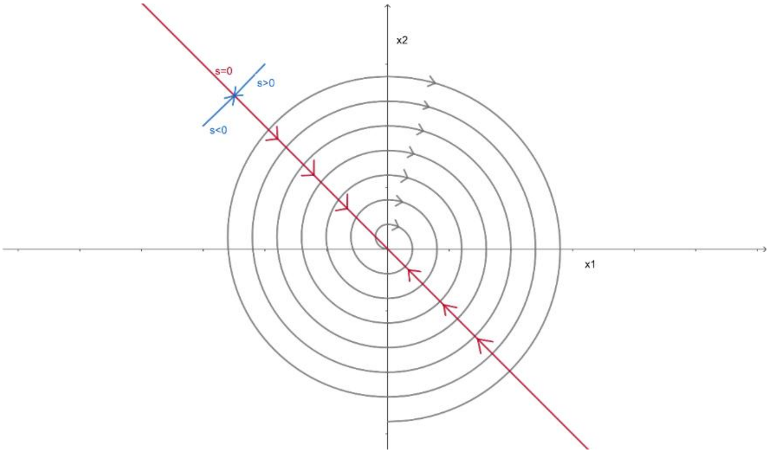


Figure 3.1 Orientation of Trajectories Toward the Sliding Surface in Sliding Mode Control

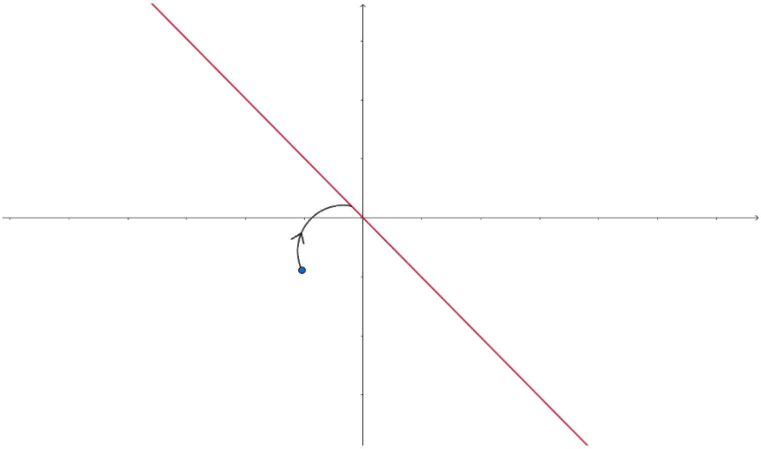


Figure 3.2 Orientation of the System Trajectory Toward the Sliding Surface in a Noise-Free Environment

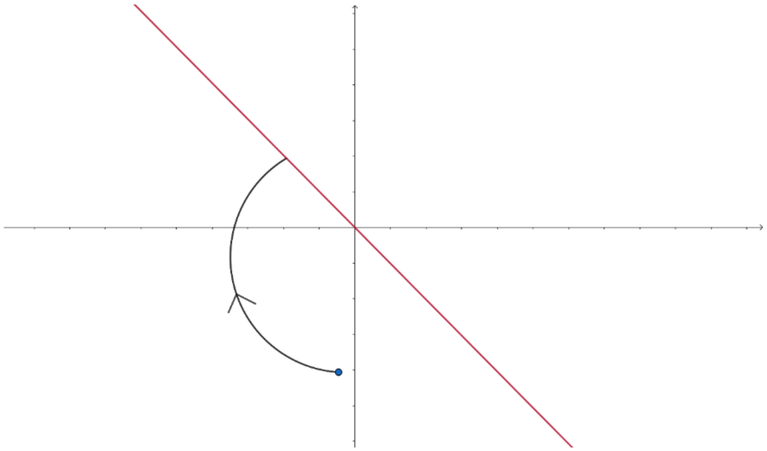


Figure 3.3 Orientation of the System Trajectory Toward the Sliding Surface in a Noisy Environment

Figures 3.2 and 3.3 illustrate the behavior of the sliding mode control scheme for the same system under different levels of disturbance. In Fig. 3.2, the system operates under nominal (noise-free) conditions, whereas in Fig. 3.3 an external disturbance is applied to the system, representing a noisy operating environment. Despite disturbances, both systems successfully reach the sliding surface defined by $s = 0$ and subsequently evolve toward the desired target point along this surface. This behavior clearly demonstrates the robustness property of sliding mode control. With appropriately designed control gains, the system maintains stability in the presence of external disturbances and achieves finite-time convergence to the sliding surface, thereby entering the sliding mode.

For a single-input system whose dynamics are described by $x^{(n)} = f(x) + b(x)u$

$$x = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ . \\ . \\ x^{(n-1)} \end{bmatrix} \qquad (78)$$

$$|f - \hat{f}| < F \qquad (79)$$

f: The exact value of f is unknown

$\hat{f}$: Estimated value of f

Although the exact magnitude of b may be unknown, knowledge of its sign is essential to ensure that the control input acts in the correct direction. Eliminating b from the system dynamics would result in the loss of the control input u.

Our objective is to determine a control input u u that ensures the output y=x tracks a desired reference x_d, despite uncertainties in $f(x)$ and $b(x)$.

$$e = x - x_d = \omega_r - \omega_{ref} \qquad (80)$$

The tracking error vector is defined as

$$S = \{s(x,t) = 0\} \qquad (81)$$

Based on the state-space formulation, a scalar time-varying sliding surface is defined as given in Eq. (81).

s : Sliding surface distance parameter

$$s = (x,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \qquad (82)$$

λ: It is a positive gain parameter that determines the dynamic response of the sliding surface.

For example, in the case of a third-order system, the sliding surface is defined as $\boldsymbol{s} = (x,t) = \left(\frac{d}{dt} + \lambda\right)^{3-1} e = \ddot{e} + 2\lambda\dot{e} + \lambda e$ where e denotes the tracking error and λ > 0 is a design parameter. When s=0, the system dynamics are constrained to the sliding surface, and the corresponding behavior on the sliding manifold is obtained.

$\dot{e} + \lambda e = 0 \Rightarrow \dot{e} = -\lambda e$ (Since the roots of the characteristic equation lie in the left half-plane at –λ, the system is stable.) It is observed that the tracking error e converges to zero. ($x \Rightarrow x_d$)

After enforcing the condition s=0, the control design is carried out by examining the sliding surface dynamics $\dot{s} = \ldots u$ In this manner, instead of directly addressing the tracking of an n-dimensional reference vector x_d , the problem is reduced to the analysis of a first-order stability problem.

In the sliding surface definition $s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e$ taking the time derivative of s yields the *n-th* derivative of the state variable, $x^{(n)}$. Since the system dynamics are given by $x^{(n)} = f(x) + b(x)u$

the control input (u) explicitly appears in the expression for ($\dot{s} = \ldots u$). s, x^{n-1}.

Since the sliding surface is defined as $s = \dot{e} + \lambda e$, knowing the bounds of s allows the corresponding bounds on the tracking error e to be determined.

$$|s| \leq \emptyset \Rightarrow \ |e| \leq \frac{\emptyset}{\lambda^{n-1}} \qquad (83)$$

Inequality (83) indicates that when the absolute value of the sliding surface is bounded within a specified tolerance ϕ, the system tracking error is also directly bounded. As the gain λ increases, the tracking error $|e|$ is forced into a smaller neighborhood around zero. While this improves tracking accuracy and system sensitivity, it may also lead to faster variations in the control signal. Therefore, the selection of the parameter λ should carefully consider this trade-off between tracking precision and control effort.

In summary, if the distance to the sliding surface is bounded by ϕ, then the magnitude of the tracking error vector is bounded by $\frac{\emptyset}{\lambda^{n-1}}$

$$\tau = \frac{n-1}{\lambda} \qquad (84)$$

The sliding surface formulation given in Eq. (82) attenuates the system error dynamics in a manner analogous to a filter. The approximate time constant of this structure can be expressed as $\tau \approx 1/\lambda$. In general, the error response reaches steady state within approximately 4τ to 5τ. In this context, increasing the gain parameter λ results in a faster system response; however, it also increases the sharpness of the control signal. Therefore, the choice of λ involves a trade-off between response speed and control smoothness.

The sliding surface function, as defined in Equation (82), represents a measure that incorporates the system error magnitude and its derivatives. Accordingly, the distance of the sliding variable s from the sliding surface quantitatively indicates the extent to which the system deviates from the reference state. Mathematically, when the condition $s| \leq \phi$ is satisfied, the tracking error is bounded as $|e| \leq \frac{\emptyset}{\lambda^{n-1}}$. This relationship clearly demonstrates the influence of sliding-mode dynamics on the system's error dynamics by ensuring boundedness and robustness.

3.1 Sliding Condition

In sliding mode control design, the primary objective is to force the system behavior onto a predefined sliding surface. Accordingly, the control input $u(t)$ must drive the system to this surface in the shortest possible time and ensure that it remains on the surface thereafter. The condition that guarantees this requirement is commonly expressed by the inequality $s.\dot{s} < 0$

Equation (85) indicates that as the system state approaches the sliding surface, the absolute value of *s(t)* decreases and the trajectory is directed toward the surface.

To first bring the system into the reaching phase and subsequently ensure the transition to the sliding phase, the control law *u(t)* must be appropriately designed. In this design process, Lyapunov stability theory is frequently employed.

Lyapunov Stability Analysis

In sliding mode control design, Lyapunov stability theory is often used to ensure system stability. This theory defines an energy-like function to analyze whether a linear or nonlinear system is stable around an equilibrium point.

3.2.1 Definition of the Lyapunov Function

A general Lyapunov function is defined as follows:

$$V(s) = \frac{1}{2}s^2 \qquad (86)$$

Here, $s\ (t)$ is a variable representing the sliding surface, and the condition $s(t)$=0 indicates the point at which the system achieves the desired dynamic behavior.

3.2.2 Stability Condition

According to Lyapunov theory, for a system to be stable, the time derivative of the Lyapunov function must be negative. This condition indicates that the system's energy decreases over time and that the system trajectories evolve toward the equilibrium point.

$$\dot{V} = s\dot{s} \leq -p|s|\,, \qquad p < 0 \qquad (87)$$

This inequality demonstrates that the system is asymptotically stable. In other words, the system behavior converges toward the sliding surface over time and remains on it thereafter.

3.2.3 Relationship Between the Control Law and Lyapunov Stability

In sliding mode control design, the control law is defined as follows:

$$u = \widehat{u} - \eta sign(s) \qquad (88)$$

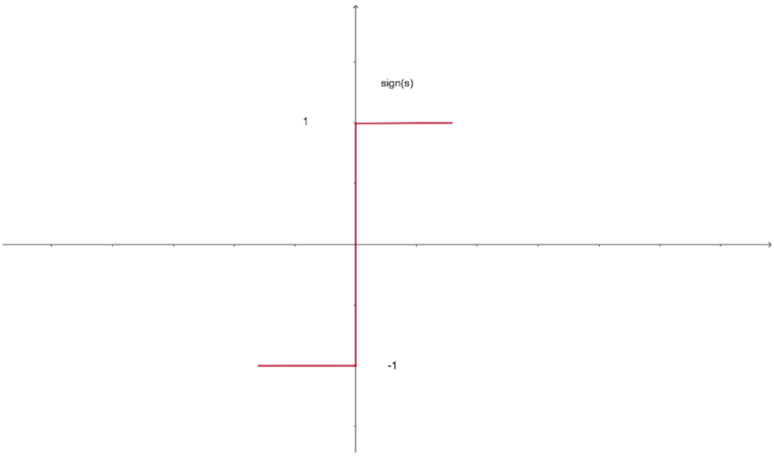


Figure 3.4 Sign Function Used in Sliding Mode Control

This control law regulates $\dot{s}$ to ensure that s decreases toward zero. Under this condition, the Lyapunov derivative becomes:

$$\dot{V}(s) = s\dot{s} = -\eta sign(s) = -\eta sign|s| \qquad (89)$$

Equation (89) satisfies the Lyapunov stability condition and guarantees that the system will reach the sliding surface and remain on it.

Lyapunov analysis enables the mathematical proof of stability for sliding mode control. The negativity of the derivative of the defined Lyapunov function indicates that the system remains stable despite disturbances and model uncertainties. Through this approach, the controller is designed robustly and systematically.

$$V = \frac{1}{2}\frac{d}{dt}s^2 \qquad (90)$$

$$\frac{1}{2}\frac{d}{dt}s^2 \leq -p|s| \qquad (91)$$

Under normal conditions, convergence to the sliding surface is asymptotic; however, by employing the Lyapunov equation given above, this convergence is ensured to occur within a finite time.

$$s\dot{s} = \frac{1}{2}\frac{d}{dt}s^2 \leq -p|s| \qquad (92)$$

Since the square of s s is taken in Equation (92), its value is always positive. If the sliding condition is satisfied, s^2 decreases continuously and eventually converges to zero, thereby ensuring that the sliding surface is reached.

$$s\dot{s} \leq -p|s| \qquad (93)$$

An input (u) must be determined such that Equation (93) is satisfied.

Here,

p denotes the convergence rate to the sliding surface.

The condition p>0 must be satisfied. The larger the value of p is selected, the faster the system reaches the sliding surface.

Once s reaches zero, the conditions $\dot{s} = \frac{1}{2}\frac{d}{dt}s^2 \leq -p|s|$ formülünde $0 \leq 0$, which implies the formation of an invariant set. Consequently, the system remains on the sliding surface. Since the system dynamics are constrained to the sliding surface, their dependence on the original system dynamics is effectively eliminated. As a result, robustness against disturbances and model uncertainties is achieved.

3.3 Reaching Time to the Sliding Surface

In sliding mode control, the control input $u(t)$ is designed to drive the system state toward the sliding surface defined by $s(x,t) = 0$. Accordingly, the system reaches the sliding surface in finite time, and this duration is referred to as the reaching time t_r.

If the control law satisfies the following inequality:

$$s\dot{s} \leq -p|s| \qquad (94)$$

Here, p>0 is a design parameter (which can be interpreted as a control gain). In this case, it is guaranteed that the system will reach the sliding surface within the following maximum time:

$$t_r \leq \frac{|s|}{p} \qquad (95)$$

$|s|$: the value of s at t=0

t_r: the reaching time to the sliding surface

p: the reaching gain (or convergence rate gain)

Inequality (95) guarantees that the error dynamics of the system are not only stable but also converge to zero in finite time, thereby ensuring the transition to the sliding phase. This property constitutes one of the most significant advantages that distinguishes sliding mode control from classical control methods.

The sliding phase starts after the reaching time. That is, for $t > t_r$, the system evolves on the sliding surface, and this interval is referred to as the sliding phase. Therefore, if the parameter p is chosen too small, the reaching time increases; if it is chosen too large, the chattering phenomenon may increase.

In classical control systems, asymptotic convergence of the system to an equilibrium point is generally considered sufficient. However, if the following situation is considered:

If the system asymptotically converges from $x \rightarrow y$ and subsequently from $y \rightarrow z$, the combined effect of these two separate asymptotic convergences does not necessarily guarantee overall asymptotic convergence from This situation may lead to divergence in the system behavior.

To eliminate this issue, the sliding mode control approach offers a two-stage strategy:

1. Reaching phase: The system reaches the predefined sliding surface in finite time. This duration is typically bounded as $t_r \leq \frac{|s|}{p}$ and is often approximated as $t_r \approx 4\tau, 5\tau$.

2. Sliding phase: Once on the sliding surface, the system dynamics evolve along this surface and asymptotically converge to the desired equilibrium point.

Thus, the system reaches the sliding surface within a finite time at the initial stage, avoiding transitions that would otherwise require infinite time at intermediate points. Once on the sliding surface, only the reduced-order dynamics govern the system behavior, thereby preventing divergence from the desired state. This feature represents one of the most significant advantages of sliding mode control.

3.4 Sliding Mode Control Design

1-) $s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e$ (96)

In the sliding mode control (SMC) approach, the first step is to define a sliding surface that includes the error signal and its derivatives. This definition is given in Equation (96).

2-) $s\dot{s} \leq -p|s|$ (97)

In order for the system to reach the sliding surface and remain on it, the condition given in Equation (97) must be satisfied. This condition guarantees that the system trajectories are directed toward the sliding surface and do not move away from it. By taking the derivative of s, the appropriate selection of the control input u is determined.

3-) $u = \hat{u} - \eta \text{sign}(s)$ (98)

To ensure that the system remains on the sliding surface in the presence of disturbances and uncertainties, the term $-\eta \text{sign}(s)$ given in Equation (98) is added to the derived control law.

Here, η is a positive gain selected with sufficient magnitude to suppress disturbances and uncertainties. However, if η is chosen too large, the system exhibits high-frequency switching behavior, which leads to undesirable oscillations known as chattering.

To mitigate the chattering problem, it is often preferred to decompose the system into first-order subsystems. This is because the switching action implemented through the sign function can be directly applied only to first-order systems. Therefore, higher-order systems are typically modeled as n first-order subsystems, and sliding mode control is implemented on this structured representation.

The function sign(s) represents the sign of s s. The product sign (s) s corresponds to multiplying a number by its sign, which results in $|s|$.

3.5 Continuous Approximations for Discontinuous Controllers

As shown in Figures 3.5 and 3.6, if the axes are defined in terms of the error variables instead of the state variables, the sliding surface is located at the origin. Figures 3.5 and 3.6 illustrate how the sliding surface is positioned in different coordinate planes. When the axes are defined using the system state variables (x_1, x_2), the sliding surface expressed as $s = x_2 + \lambda x_1$ may be located at different positions in the state plane depending on the parameter values.

However, when the axes are defined in terms of the error variables (e1,e2), that is, when the plot is constructed based on the difference between the system output and the reference, the sliding surface equation s(e)=0 directly passes through the origin. This follows from the fundamental control objective that the error must converge to zero. Therefore, in such phase-plane representations, the sliding surface is always located around the origin. This approach allows a more meaningful and direct observation of how the system behavior deviates from the reference.

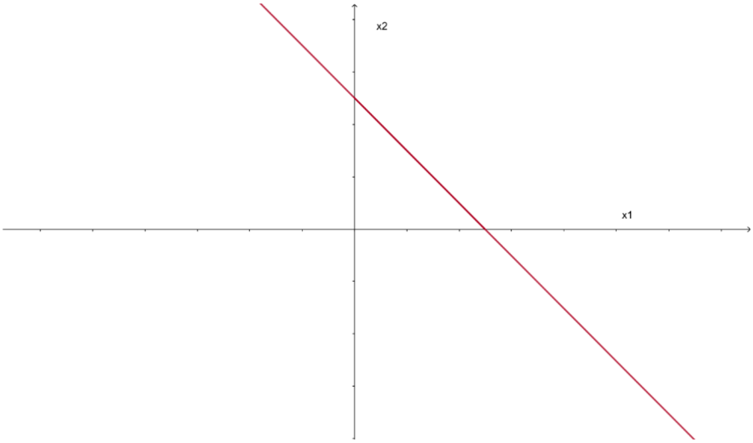


Figure 3.5 Sliding Surface Defined in the State Space

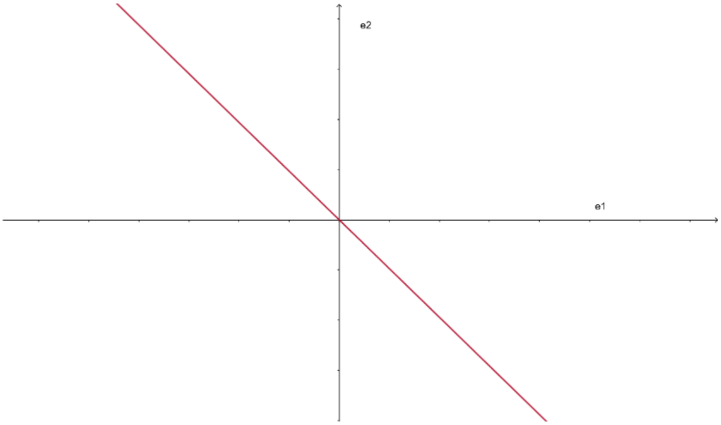


Figure 3.6 Sliding Surface Defined in the Error Space

In real-time control systems, delays in measurement data transmission or the time required for the control input to take effect may cause the system to cross the sliding surface (s=0) before fully settling on it. In such a case, a control law of the form $-\eta \cdot sign(s)$ forces the system to react in the opposite direction. However, since measurement and control delays persist, the system overshoots s=0 again. This process repeats continuously, resulting in high-frequency back-and-forth oscillations. This behavior is referred to as chattering, and it not only degrades control performance but may also lead to adverse effects such as mechanical wear, energy losses, and undesirable noise. Figure 3.7 illustrates the oscillatory behavior that occurs after the sliding surface is reached. The system rapidly switches between the two sides of the s=0 surface and, instead of exhibiting ideal sliding mode behavior, enters a chattering regime.

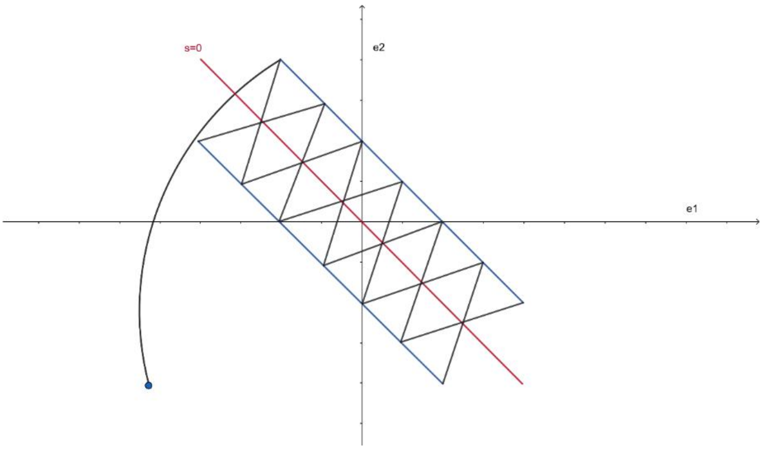


Figure 3.7 Chattering Effect During the Reaching Phase of the Sliding Surface

3.6 Chattering Suppression

Classical Control Law in Sliding Mode Control:

$$u = \hat{u} - \eta sign(s) \tag{99}$$

The sign(s) function in the control law causes very high-frequency switching around the sliding surface (s=0), leading to an undesirable oscillatory behavior known as chattering. This phenomenon may result in adverse effects such as mechanical wear, acoustic noise, increased energy consumption, and degradation of control performance.

To reduce this effect, a continuous function such as the saturation (sat) function is used in practice instead of the sign(s) function. With this approach, the control law is redefined as follows:

$$u = \hat{u} - \eta sat\left(\frac{s}{\emptyset}\right) \tag{100}$$

Here:
$\emptyset$: a positive parameter that determines the width of the boundary layer.
sat(x): the saturation function, defined as follows:

$$sat(x) = \begin{cases} 1 & if\ x > 1 \\ x & if\ |x| \le 1 \\ -1 & if\ x < -1 \end{cases} \tag{101}$$

With this approach, instead of remaining exactly on the sliding surface (i.e., s=0), the system is confined within the boundary layer defined by $-\emptyset \le s \le \emptyset$.

This represents a deliberate deviation from perfect surface tracking. In return, the chattering effect is significantly reduced.

If ∅ is chosen to be small, the system remains very close to s=0, resulting in a low tracking error; however, chattering may increase. If ∅ is selected to be large, the system stays within a wider band around the sliding surface, reducing chattering but leading to a higher tracking error. Therefore, the value of ∅ should be optimized according to the system dynamics and application requirements.

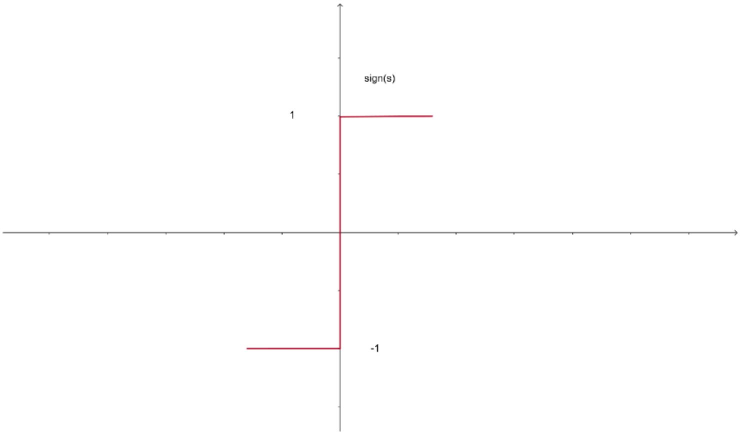


Figure 3.8 Sign(s) Function

Figure 3.8 illustrates the discontinuous nature of the *sign(s)* function. This abrupt switching behavior causes high-frequency switching, resulting in chattering.

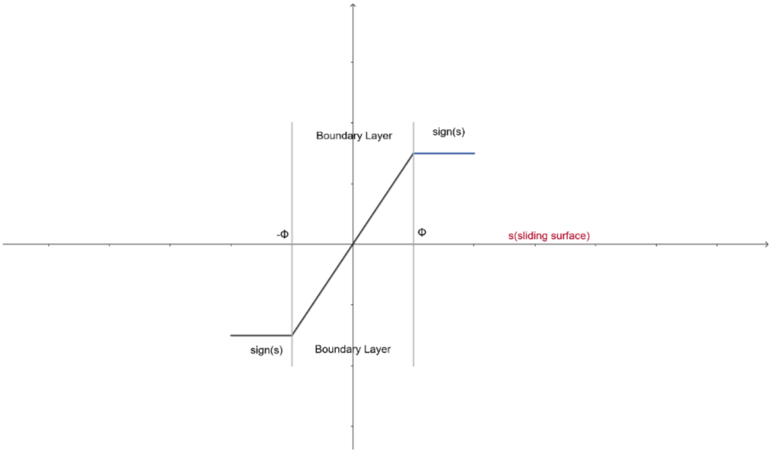


Figure 3.9 Saturation Function

Figure 3.9 illustrates how the $sat\left(\frac{s}{\emptyset}\right)$ function provides a linear transition within the boundary layer ($\pm\emptyset$), thereby smoothing the chattering effect. This structure ensures continuity in the control signal.

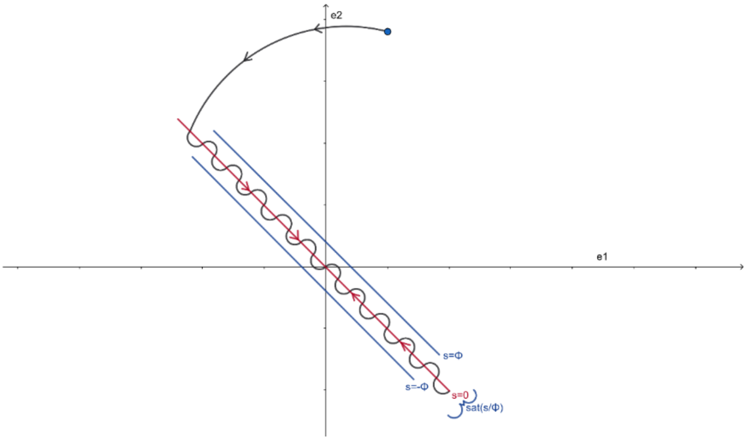


Figure 3.10 Suppression of Chattering Within the Boundary Layer Using the Saturation Function

Figure 3.10 shows that when the saturation function is applied, the system does not hit the sliding surface directly; instead, it oscillates within a boundary layer around the surface. In this way, the chattering effect is reduced.

V/F Control Method

Induction motors are among the most widely used motor types in industrial applications. Among the control methods developed for speed regulation of these motors, V/f control (voltage-to-frequency control), also known as scalar control, is widely preferred due to its simplicity and ease of implementation. This method enables speed control by maintaining the motor's magnetic flux at an approximately constant level.

The magnetic flux (ϕ) in the stator windings of the induction motor can be approximately defined by the following expression:

$$\phi \approx \frac{V}{f} \tag{102}$$

V: The phase voltage applied to the stator (RMS)

f: Frequency (Hz)

This ratio is referred to as the V/F ratio. To maintain a constant magnetic flux, this ratio must be kept constant.

When this condition is satisfied:

1. The motor magnetizing flux remains constant.
2. Torque production becomes stable.
3. The motor does not enter saturation or weak-flux regions.

The synchronous speed of the induction motor is given by:

$$n_s = \frac{120*f}{p} \tag{103}$$

n_s: Synchronous speed (rpm)

f: Frequency (Hz)

p: Number of poles

As shown in Equation (103), increasing the frequency increases the motor's synchronous speed. If the speed is increased while maintaining a constant V/F ratio, the motor operates at a higher speed and remains stable.

MATLAB Implementation

Design of the Sliding Mode Control Block:

$$\dot{x_1} = x_2 \tag{104}$$

$$\dot{x_2} = f(\overline{x_2}) + u + \omega_2 \tag{105}$$

$$e_1 = x_1 - x_1^d \tag{106}$$

$$e_2 = x_2 - x_2^d \tag{107}$$

$$\dot{e}_1 = x_2 - \dot{x}_1^d = e_2 \tag{108}$$

$$\dot{e}_2 = f(\overline{x_2}) + u + \omega_2 - \dot{x}_2^d \tag{109}$$

Sliding surface:

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \tag{110}$$

$$s = \frac{de_1}{dt} + \lambda e_1 = \dot{e_1} + \lambda e_1 = e_2 + \lambda e_1 \tag{111}$$

$$s = 0; \tag{112}$$

$$\dot{e_1} = -\lambda e \tag{113}$$

Lyapunov:

$$V = \frac{1}{2}s^2 \tag{114}$$

$$\dot{V} \leq -p|s| \tag{115}$$

$$s\dot{s} \leq -p|s| \tag{116}$$

$$|s| = sign(s)s \tag{117}$$

$$s\dot{s} \leq -psign(s)s \tag{118}$$

$$\dot{s} \leq -psign(s) \tag{119}$$

$s(0)$: Initial value of the sliding surface and t_r: Reaching time to the sliding surface (120)

$$s(t_r) = 0 = \dot{e_1} + \lambda e_1 \Rightarrow \dot{e_1} = -\lambda e \tag{121}$$

$$s(t_r) - s(0) \leq -psign(s)t_r \tag{122}$$

$$t_r \leq \frac{|s(0)|}{p} \text{ (finite time convergence)} \tag{123}$$

Equation (123) shows that the system reaches the sliding surface in a time shorter than $\frac{|s(0)|,}{p}$.

$$\dot{V} = s\dot{s} = s(\ddot{e_1} + \lambda\dot{e_1}) \tag{124}$$

$$\Rightarrow s(f(\overline{x_2}) + u + \omega_2 - \dot{x}_2^d + \lambda e_2) \tag{125}$$

To eliminate the undesired terms, the control input u is selected as $(-f(\overline{x_2}) + \dot{x}_2^d - \lambda e_2)$

and the term için **-$\boldsymbol{\eta sign(s)}$** is added to maintain the system on the sliding surface.

$$s\dot{s} = s(f(\overline{x_2}) - \dot{x}_2^d + \lambda e_2 - f(\overline{x_2}) + \dot{x}_2^d - \lambda e_2 - \eta sign(s) + \omega_2) \tag{126}$$

$$s\dot{s} = s(-\eta sign(s) + \omega_2) \tag{127}$$

$$s\dot{s} = s\omega_2 - s\eta sign(s) \quad (128)$$

$$s\omega_2 \leq |s||\omega_2| \quad (129)$$

Equation (129) shows that the product of the absolute values of two numbers is greater than or equal to the product of the numbers themselves.

$$s\dot{s} \leq |s||\omega_2| - \eta|s| \quad (130)$$

$$s\dot{s} \leq |s|(|\omega_2| - \eta) \quad (131)$$

In Equation (131), if the parameter η is chosen to be greater than ω_2, which represents the disturbance effects, Equation (133) is obtained. Consequently, the system converges to the sliding surface. This property is referred to as robustness.

$$\dot{V} = s\dot{s} \leq -p|s| \quad (132)$$

$$|s|(|\omega_2| - \eta) \leq -p|s| \quad (133)$$

$$|\omega_2| + p < \eta \quad (134)$$

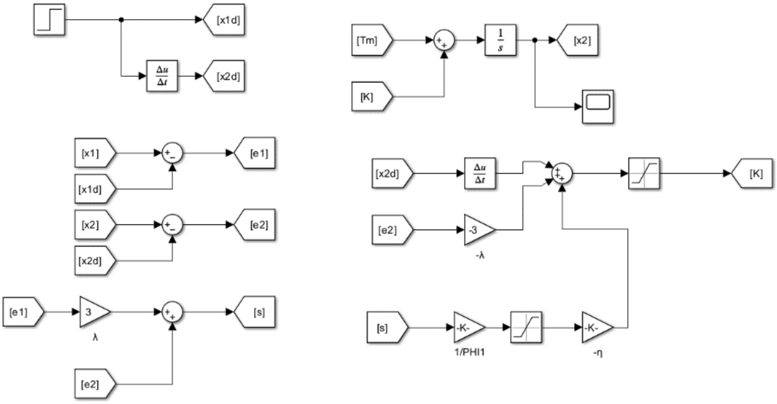


Figure 5.1. Sliding Mode Control Block Implemented in MATLAB/Simulink

Based on the calculations presented in Figure 5.1, the corresponding MATLAB/Simulink blocks have been constructed. The parameter K represents the control input (u), T_m denotes the disturbance term (ω_2)

$$\dot{x_1} = x_2 \quad (135)$$

$$\dot{x_2} = u + \omega_2 \quad (136)$$

$$e_1 = x_1 - x_1^d \tag{137}$$

$$e_2 = x_2 - x_2^d \tag{138}$$

$$\dot{e}_1 = x_2 - \dot{x}_1^d = e_2 \tag{139}$$

$$s = e_2 + \lambda e_1 \tag{140}$$

$$K = \dot{x}_2^d - \lambda e_2 - \eta sat\left(\frac{s}{\emptyset}\right) \tag{141}$$

The operations performed in Simulink are given in Equations (135–141).

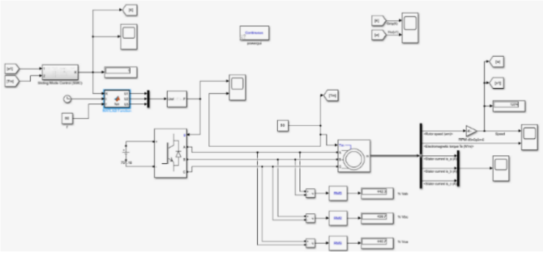


Figure 5.2 Speed Control Scheme of a Three-Phase Induction Motor Implemented in MATLAB/Simulink

Figure 5.2 shows a three-phase induction motor operating under V/F control and sliding mode control.

Three-Phase Voltage Generation

The value of K obtained from the sliding mode control algorithm is used as the amplitude of the three-phase sinusoidal voltages. The sinusoidal reference signals for each phase are given in Equations (142), (143), and (144).

$$U_1 = K * \sin(2 * \pi * f * t) \tag{142}$$

$$U_2 = K * \sin(2 * \pi * f * t - 2 * \frac{\pi}{3}) \tag{143}$$

$$U_3 = K * \sin(2 * \pi * f * t + 2 * \frac{\pi}{3}) \tag{144}$$

K: $\frac{V}{f}$ the variable that maintains the ratio constant

f: Constant stator frequency (60 Hz)

t: Time

With this structure, three-phase sinusoidal reference signals are generated with equal amplitudes and a 120° phase shift, resulting in a configuration suitable for supplying the motor.

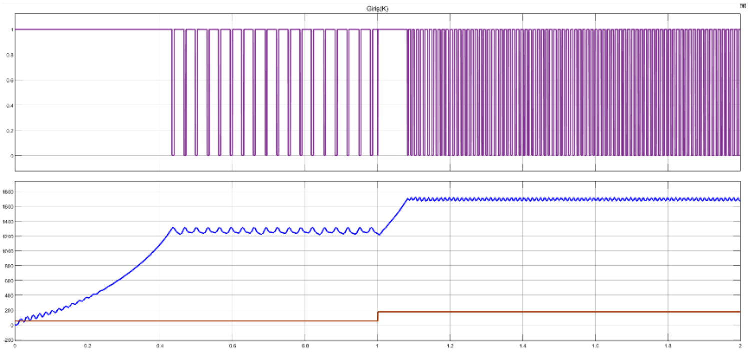


Figure 5.3 Motor Speed Responses and Control Signal Generated by Sliding Mode Control

Figure 5.3 shows the time-dependent behavior of the dynamically determined V/f ratio (K) obtained from the sliding mode control algorithm (purple) and the corresponding motor speed (blue). Initially, the system accelerates gradually toward the 1300 rpm reference. After a certain period, the sliding mode control algorithm becomes active and adjusts the value of K according to the system dynamics. At t= s, the reference speed is changed to 1700 rpm. In response to this change, the value of K abruptly approaches zero and then increases again, steering the motor toward the new reference.

This temporary drop to nearly zero originates from the high-gain nature of the sliding mode control structure and causes a short-term decrease in speed while the system re-establishes equilibrium. In general, the motor reaches both reference values within finite time and maintains stable operation thereafter.

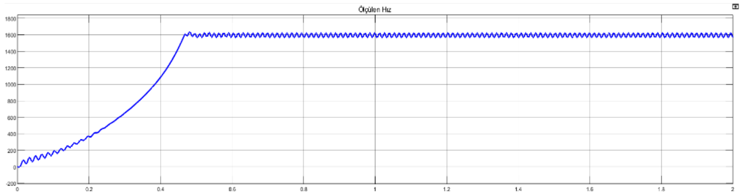


Figure 5.4 50 Nm yük torkunda motorun hız grafiği

Figure 5.4 presents the speed–time response of the three-phase induction motor operating under an applied load torque of , $T_m = 50\ Nm$. As observed from the speed curve, the motor accelerates from low speeds and reaches approximately 1600 rpm in about 0.47 seconds, after which it stabilizes at that level.

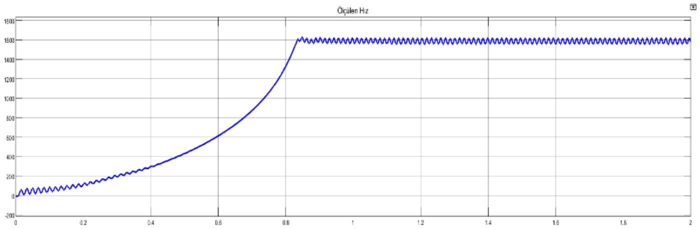


Figure 5.5. Motor Speed Response Under a 100 Nm Load Torque

Figure 5.5 illustrates the time-dependent variation of the motor speed under a constant load torque of $T_m = 100\ Nm$ applied to the system. As observed from the speed curve, the motor begins to accelerate from t=0 s and reaches approximately 1600 rpm at around 0.85 seconds.

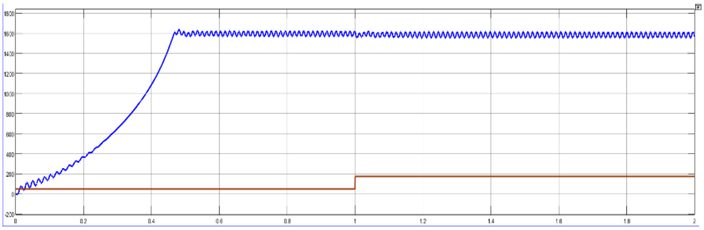


Figure 5.6. Motor Speed Response to Variable Load Torque Under Constant Reference Speed

As shown in Figure 5.6, the initial load torque (brown) is set to $Tm = 50$ Nm, at which the motor reaches approximately 1600 rpm (blue) in about 0.44 seconds and enters steady-state operation. At t=1s of the simulation, the load torque is suddenly increased to *Tm=175 Nm* using a step block. Despite this significant increase, the sliding mode control structure maintains the motor speed at the desired level. Although small-amplitude transient oscillations are observed, the system quickly recovers and preserves the reference speed. This behavior demonstrates the high robustness of the sliding mode control algorithm against load variations.

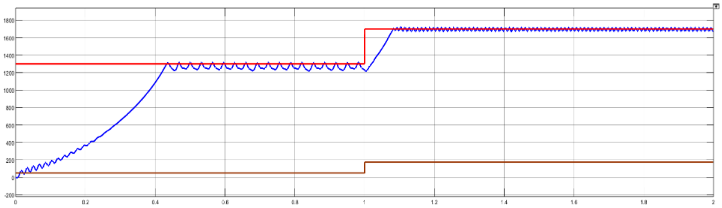


Figure 5.7. Motor Speed Response Under Simultaneous Changes in Reference Speed and Load Torque

Figure 5.7 shows the system's speed-tracking performance (blue) when both the desired speed (red) and the load torque (brown) are simultaneously changed. At the beginning of the simulation, the reference speed is set to hız $x_1^d = 1300\, rpm$ and the load torque to *Tm=50 Nm*. The motor successfully reaches the desired value within approximately 0.44 seconds and enters the sliding mode region. At this stage, low-amplitude chattering, which is characteristic of classical sliding mode control, can be observed.

At t=1 s, the reference speed is increased to $x_1^d = 1700\, rpm$ and the load torque is simultaneously stepped up to *Tm=175 Nm*. This abrupt change tests the control system's robustness. As shown in the graph, despite simultaneous variations, the motor does not lose stability and converges to the new reference value. Within approximately 0.1 seconds, the system re-enters the sliding mode. These results clearly demonstrate that sliding mode control provides both high robustness and rapid adaptation capability under sudden and concurrent disturbances.

CONCLUSION

In this study, a dynamic and robust speed control structure for a three-phase induction motor has been developed by integrating the conventional voltage/frequency (V/F) control method with a Sliding Mode Control (SMC) algorithm. By continuously updating the V/F ratio based on the instantaneous speed error through sliding mode control, the proposed approach effectively overcomes several limitations of the classical V/F scheme, particularly under dynamic reference variations and load disturbances.

Simulation studies were conducted under both steady-state and transient operating conditions to evaluate the performance of the proposed control strategy. The motor successfully reached the initial reference speed of 1300 rpm with a fast transient response, and when the reference was increased to 1700 rpm, the system achieved the new operating point with a short settling time and stable behavior. Moreover, even under abrupt load torque variations from 50 Nm to 175 Nm, the motor speed remained close to the reference value, demonstrating the robustness of the control structure.

The simulation results indicate that the adaptive V/F ratio (i.e., the gain K) generated by the sliding mode controller varies smoothly with the system dynamics, resulting in improved speed tracking performance and enhanced stability. However, the inherent chattering phenomenon associated with classical sliding mode control may introduce high-frequency oscillations in the control signal, potentially affecting system performance. To address this issue, the discontinuous sign(s) function was replaced with a continuous saturation function sat(s/ϕ), introducing a boundary layer around the sliding surface and effectively suppressing chattering to a significant extent.

Despite these improvements, one of the well-known limitations of conventional V/F control—namely, insufficient torque production in low-speed (low-frequency) operating regions—was also observed in this study. This behavior arises due to the dominance of stator impedance at low frequencies, which weakens the magnetic flux and may lead to unstable motor operation. Although the proposed sliding mode–based V/F control structure yields better performance than the classical V/F method in this region, the limitation could not be completely eliminated.

In conclusion, the proposed sliding mode control–based V/F strategy provides enhanced dynamic performance, strong robustness against load and reference variations, and improved adaptability compared to the conventional V/F control approach. Nevertheless, for achieving more effective torque control at low speeds, future research may focus on advanced control techniques such as Field-Oriented Control (FOC) or higher-order sliding mode algorithms (e.g., Super-Twisting), which are expected to further improve performance and robustness.

REFERENCES

Bayramoglu, H., & Kömürcügil, H., (2012). Time-Varying Sliding-Coefficient-Based Approach to the Decoupled Terminal Sliding Mode Controller Design. Proceedings of the 17th International Conference on Methods and Models in Automation and Robotics (MMAR2012), pp. 481-486.

Biricik, S., Kömürcügil, H., & Basu, M., (2015). Sliding Mode Control Strategy for Three-Phase DVR Employing Twelve-Switch Voltage Source Converter. *Proceedings of the 41st annual conference of the IEEE Industrial Electronics Society (IECON'15),* pp. 921-926.

Cömert, C., & Kasnakoğlu C. (2016). Design of a Sliding Mode Controller for Quadcopter. Electrical – Electronics and Biomedical Engineering Conference, Bursa, Türkiye.

Cömert C., Kasnakoğlu C., (2017). Comparing and Developing PID and Sliding Mode Controllers for Quadrotor*,* International Conference on Mechatronics Systems and Control Engineering (ICMSCE 2017), Kayseri, Türkiye.

Efe, M.Ö., & Kasnakoğlu, C., *A.* (2008). Fractional Adaptation Law for Sliding Mode Control, *International Journal of Adaptive Control and Signal Processing*, 22 (10), 968–986. doi: 10.1002/acs.1062.

Kürkçü, B., & Kasnakoğlu, C., (2015). Estimation of Unknown Disturbances in Gimbal Systems, *2015 International Conference on Automatic Control (ICOAC 2015)*, April 28-29, Ankara, Türkiye.

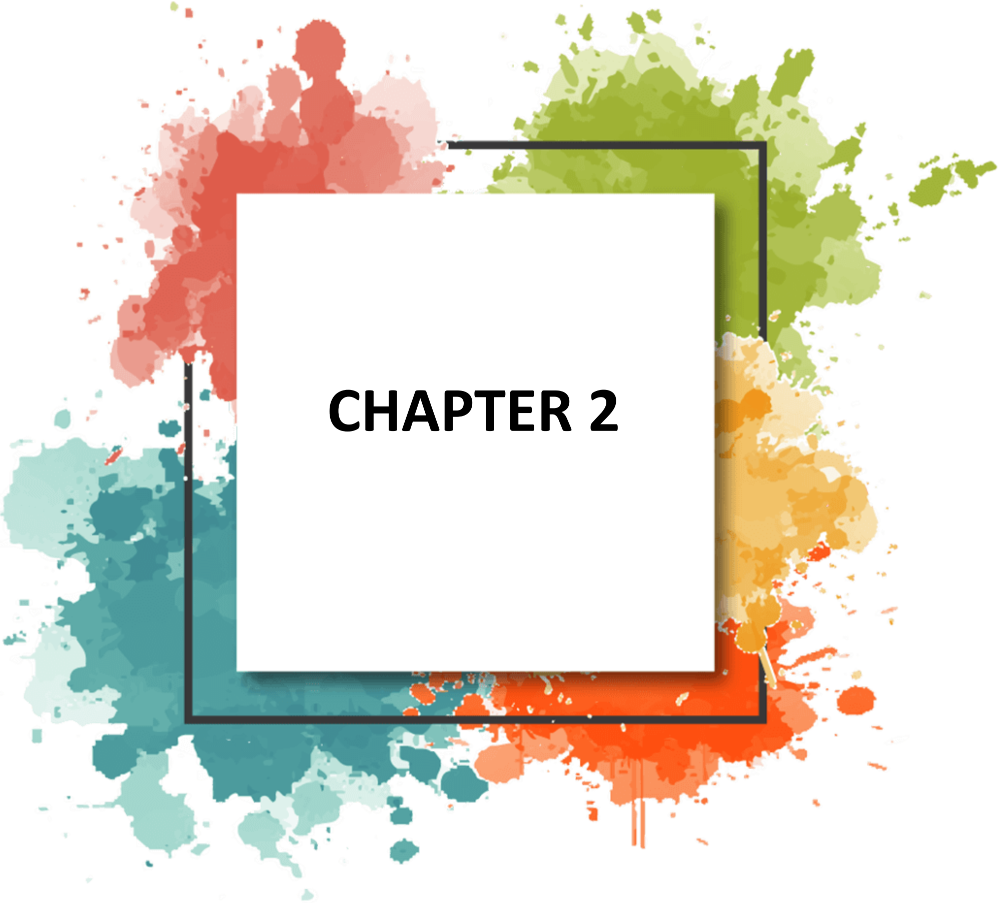
Nurettin, A., & Inanç, N.,(2022). Performance Study of Sliding Mode Speed Controller for Induction Motor Vector Control Using MATLAB/SIMULINK", *IJESA*, 6(3), pp. 65–72.

Shtessel, Y., Edwards, C., Fridman L., & Levant A. (2014). *Sliding Mode Control and Observation*. Boston, MA: Birkhäuser.

Yaylacı, E. K., (2011). Sliding Mode Speed Control Of Induction Motor, Master's Thesis, Electrical-Electronics Engineering, Sakarya University.

Young, K. D., Uzguner, U., & Hedrick, J. K. (1999). A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, **7**(3), 328–342.

CHAPTER 2



Throughput Prediction on Automated Guided Vehicles

Tugba Kara[1] & Ahmet Gökçen[2]

1. Introduction

In recent years, with the progress made by artificial intelligence, the importance of data has increased. This data is produced by many sources such as users, sensors, etc. The information obtained by interpreting data has formed the building blocks of today's technology. Thanks to the interpretation of data, systems are allowed to operate in a more agile manner. Prediction systems provide a support to quality of services. The most remarkable key element in this context is the throughput. Throughput can be defined as the amount of data successfully transmitted over the network in a certain period of time. However, throughput does not remain stable for many reasons (He, Dovrolis, & Ammar, 2007) like location, time, wireless medium etc. Also this is valid for vehicular radio channels; its statistical data tends to change in different physical environments (Bernadó, Zemen, Tufvesson, Molisch, & Mecklenbräuker, 2012). These disruptions in data flow have a great impact on the performance of real systems. Modern systems require high throughput along with low latency. When looking at this perspective from the industrial field, it is seen that throughput prediction is used as much as possible. Reliable and stable throughput is the indispensable key of automation systems. One of the biggest examples in this field is industrial mobile robots.

Mobile robots like Automated Guided Vehicles (AGVs) minimize human labor in many fields like logistics, production (Zhang, Yang, Wang, Guan, Ding, & Lee, 2023a). They accelerate production, and reduce the margin of error caused by humans. Therefore, mobile robots also need to operate efficiently and accurately. The ability to exchange data correctly and effectively with other hardware around them affects the performance of mobile robots. This data exchange requires a stable network connection and high throughput values when high mobility is available. Throughput prediction becomes important when the mobile robots with high mobility are important in critical business tasks. In addition, based on the studies about important topics such as communication of mobile robots, maneuver control (Molina-Masegosa et al., 2023), and latency

[1] Affiliation: Department of Computer Engineering, Iskenderun Technical University, Iskenderun 31200, Turkey, ORCID: 0009-0007-7038-1995

[2] Affiliation: Department of Computer Engineering, Iskenderun Technical University, Iskenderun 31200, Turkey ORCID: 0000-0002-7569-5447

(Lau, Ekpo, Zafar, Ijaz, & Gibson, 2023), the importance of developments in this field is emphasized once again. Some methods have been tried with machine and deep learning techniques for throughput prediction (Ghasemi, 2018; Raca et al., 2020; Minovski, Ögren, Mitra, & Åhlund, 2023). It has been learned that when the throughput value is predicted correctly in advance, a decrease in productivity can be prevented. It is an undeniable fact that with the increased reliability of these predictions, the data flow between mobile robots and the infrastructure can be directly planned.

In this study, we investigate the dataset published in (Hernangómez et al., 2024). Compared to the work in (Prokop, Połap, & Srivastava, 2023), we take additional improvement by creating a simple ANN model with residual connections, a hybrid deep learning model, and a Bidirectional LSTM (BiLSTM) model. Due to the inadequacy of linear prediction of traditional methods, the lack of correlation between throughput and other features (He et al., 2007), and the successful results of machine learning and deep learning in throughput prediction (Zhang, Patras, & Haddadi, 2019), previous studies were examined and comparisons were made with methods developed and tested. First, deep learning methods were chosen due to their advantage of not requiring feature selection and their success with time series data. Then, the data was divided into train, test, and validation sets using KFold to prevent overfitting. Finally, ANN, BiLSTM and hybrid model with multi-head attention are trained and compared. To evaluate the learned models, Mean Square Error (MSE), Root Mean Square Error (RMSE), Absolute Relative Error at 95th Percentile (ARE95), Percentage of Absolute Relative Errors below 10% (PARE10) metrics are used, and R^2 was calculated to understand the model's ability to explain the data. According to the results of these metrics, the ANN model performed much better than models like BiLSTM and hybrid model, and in the uplink direction, ANN performed better than the proposed method in (Prokop et al., 2023) with a simpler model architecture.

1.1. Related Works

Many studies have been done so far for throughput prediction. In this part, studies related to throughput prediction have been examined, and based on that, the general logic behind the approaches to the prediction problem has been tried to be explained. Among these studies, vehicular network has gained importance. The main aim of these studies is to predict throughput, to prevent problems in tasks such as communication and navigation of mobile robots like AGVs, to improve user experience quality, to increase the stream quality of video conferences that need high bandwidth, and to improve experience quality in user-focused situations like gaming.

These studies include areas such as multimedia streaming services, mobile 5G, LTE and other internet bandwidth prediction, and autonomous driving support with V2X in AGV mobile robots, including scenarios where moving users use mobile networks. Although deep learning models are mostly used in these studies, machine learning methods like Support Vector Machine (SVM) and Random Forest (RF) have also been tried. In general, RMSE and MSE are the main metrics considered, while ARE95 and PARE10, which allow a clearer interpretation in regression analysis, have also been used in a few studies.

ML model-based TCP throughput prediction is proposed by (Wei, Kawakami, Kanai, Katto, & Wang, 2018) for mobile networks. In the study, because the user is in movement (walking, static, or in a bus), as a first step of TRUST they identified the pattern of the movement with sensor data and communication factors using ML. When the identification of movement is done, an LSTM model is used for throughput prediction. As a metric, NRMSE is used in all cases. In the moving bus case, they observed that throughput prediction errors are decreased by 44%.

For real-time mobile bandwidth prediction (Mei et al., 2020) to improve QoE, authors worked on different movement types like subway and bus rides with different times of day to analyze the time influence in bandwidth prediction. They used two LSTM models to predict bandwidth for one-second-ahead and multi-second-head predictions; additionally, for different movement types they developed Multi Scale Entropy to set logic between bandwidth prediction and patterns. At the end, for real-time model inference they developed Model Switching to select the best performing LSTM model and Bayes Model Fusion to switch mobility types. As a result, they achieved RMSE with 12% and Mean Absolute Error (MAE) with 17%.

When the topic becomes 5G, (Azmin, Ahmadinejad, & Shahriar, 2022) created a transformer-based model named Informer to predict future bandwidth data, and 95% error decrease is observed in prediction. They applied RF to select features with Informer to obtain the best prediction model.

The well-known LSTM model for time series datasets is applied with an attention mechanism (Na, Shin, Lee, & Lee, 2023). They collect TCP data and then convert them to throughput data for prediction. In the training phase they use a sliding window approach with 4 time steps, and RMSE values are lower than those of Seq2Seq (Sutskever, Vinyals, & Le, 2014) and LSTM without attention models.

In the case of low-delay video streaming, to make smooth bitrate switching because of throughput decrease, (Liu, Zhang, Du, & Qiu, 2022) proposed TP-Reinforcement Learning which has three steps. Firstly, to make the main decision it inherits Reinforcement Learning; then Mogrifier LSTM is used to predict throughput; and finally bitrate is decided for further processes. The TP-RL model is evaluated in different environments and QoE is increased up to 20.7%.

For dynamic wireless networks (4G/5G), because of their changing and unstable nature, (Zhang, Jiang, Jin, Li, & Chen, 2023b) proposed CapRadar bandwidth prediction method to overcome various problems, highlighting classification of bandwidths in different scenarios. Their study has two main stages. Firstly, they classify the bandwidth scenario using a pretrained SVM model, and based on the classification, the prediction model is switched from a list of models. Their model reduced RMSE by 18.9% and MAE by 21.5%.

In (Prokop et al., 2023), a different version of the same dataset was used. To manage resources in a throughput manner, Quality of Service (QoS) within vehicles like AGVs is considered. For throughput prediction, Gated Recurrent Unit (GRU) with attention mechanism was chosen because of its simpler architecture than LSTM. Two separate fully connected neural networks are used. While GRU with attention is built for throughput historical data, the first Neural Network (NN) is built for sensor data. The third NN is responsible for the outputs of both models to predict final throughput. Because of the dataset structure they tested their models in both uplink and downlink throughput sides and achieved RMSE for uplink of 1.6 Mbps and for downlink 0.1 Mbps. However, they did not test the results with other application-centric accuracy metrics such as ARE95 and PARE10.

Mixed LTE/5G cellular network data with 100 ms granularity throughput prediction is studied by (Basit et al., 2024). Through the study, data was collected for 4 days. As the focus of our study, the authors also attached importance to ARE95, PARE10 metrics and evaluated pretrained models from previous studies like Multilayer Perceptron (MLP), Gradient Boosting Decision Trees (GDBT), Multivariate Time Series (MVTS) with new collected data. Even with low RMSE values, certain models cannot achieve reasonable ARE95, PARE10 values because of the structure of the dataset. There are many inputs that look very similar, but their true labels are very different.

(Raca et al., 2024) delve into limitations of throughput prediction in case of PHY API operation and labelling of throughput in cellular systems for ML-based models. They also highlight feature engineering importance in the case of prediction. They extend their work to transfer learning to improve throughput prediction inference in real-world networks, especially in QoE for video streaming.

Short-term bandwidth prediction is focused to improve QoE by (Afshar, Razavi, & Moshirpour, 2025). Authors used a real-world dataset to overcome prediction problems by creating closed-loop clustering-based Global Forecasting Models. They also focused on data preprocessing to test the reliability of predicting, a dynamic feedback system to make robust prediction, and how to select model architecture to evaluate in the prediction phase between global and local models.

2. Methodology

2.1. Performance Metrics

In our study, due to the importance of application-centric accuracy metrics such as ARE95 and PARE10, these were included along with the RMSE metric. Besides, the R² value, which can be understood as the model's ability to explain the data, was calculated. The RMSE, ARE95, and PARE10 calculations are as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{y_i} - y_i| \tag{1}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (\hat{y_i} - y_i)^2 \tag{2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{y_i} - y_i)^2} \tag{3}$$

$$\mathrm{PARE10} = \frac{100}{n}\sum_{i=1}^{n} \mathbb{1}\left(\frac{|\hat{y_i} - y_i|}{|y_i|} \leq 0.1\right) \tag{4}$$

$$\text{ARE95} = P_{95}\left(\left\{\frac{|\hat{y}_i - y_i|}{|y_i|} \times 100\right\}_{i=1}^{n}\right) \tag{5}$$

where y_i and $\hat{y}_i$ represent the actual and predicted values, respectively, n is the total number of data points, $\mathbb{1}(\cdot)$ is the indicator function, and $P_{95}(\cdot)$ denotes the 95th percentile operation.

2.2. Dataset

The dataset (Hernangómez et al., 2024) used in this study includes vehicle-to-infrastructure plus sensor (V2I+) data between an autonomous cleaning robot used in industrial areas and a private cellular network. As the authors also mentioned, the dataset can be used in studies like prediction of QoS, line-of-sight, fingerprinting, etc. This dataset was chosen because such needs still continue and are open to development for indoor robots.

If we go into the details of the dataset, besides high-resolution LTE data, it also includes position data and sensor data processed with the Robotic Operating System (ROS), and is enriched with throughput and delay data calculated from QoS measurements, which also expands its usage area for pQoS.

The AGV can be controlled remotely when necessary, thanks to its ability to connect to headquarters and report problems in unexpected situations. For this, having a strong and efficient internet connection is very important. The AGV movement path and the base station's location in the room are shown in Figure 1 (Fraunhofer HHI, 2023). In addition, the table of features in the dataset is also given roughly in Table 1. Detailed information such as how the data was collected and what the hardware was can be found in (Hernangómez et al., 2024).

Features	**Value Range (min – max)**	**Description**
RSRP	−79.4, −65.666	Reference signal received power in dB for DMRS
RSRQ	−37.666, −4.6	Reference signal received quality in dB
RSSI	−89, −50.333	Received signal strength indicator in dB
SNR	−10, 23.8	Signal to noise ratio in dB
throughput	–	Throughput value for uplink and downlink
ping	–	Ping reply received time in ms
odom	–	ROS odom data for position, speed and orientation

Features	Value Range (min – max)	Description
line of sight	0, 1	Direct sight indication from AGV to base station
obstacle sum	0, 97561	Amount of obstacle between AGV and base station

Table 1. V2I+ dataset features and their descriptions.

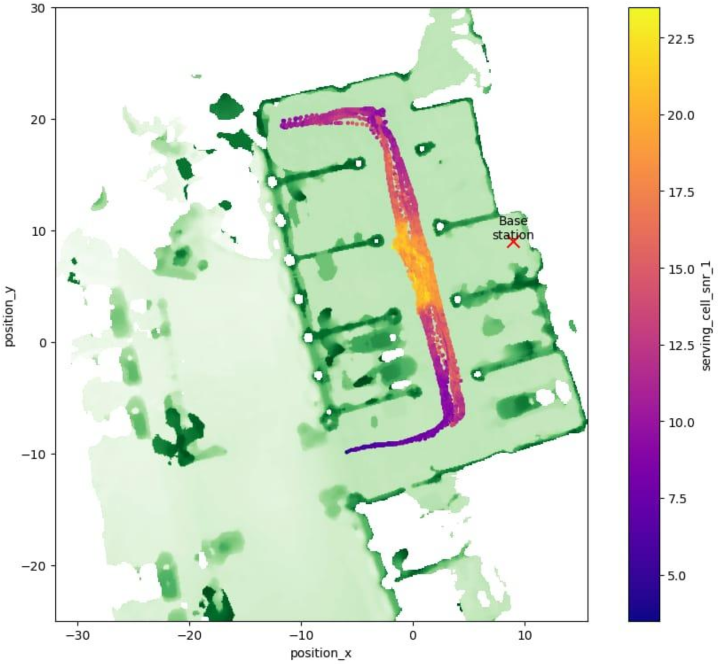


Figure 1. AGV routing description in a room with base station location and SNR information.

2.3. Data Preparation

Because of the NaNs (Not a Number) in the dataset, after dropping these columns and rows, uplink and downlink throughput prediction were processed separately with the remaining data. For this reason, the total amount of data was separated as 16,192 for uplink and 14,239 for downlink. From this data, 10% was separated for validation and test each, and in total, 80% was used for training. To prevent overfitting, cross-validation was done with 5 folds (the default value commonly used in machine learning practices) using KFold while training the model.

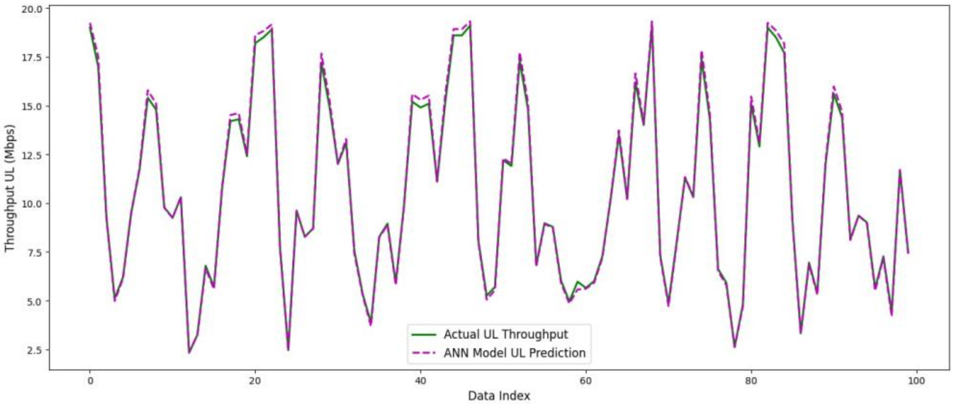


Figure 2. Prediction results of ANN model vs actual data for the uplink direction for index 0 to 100.

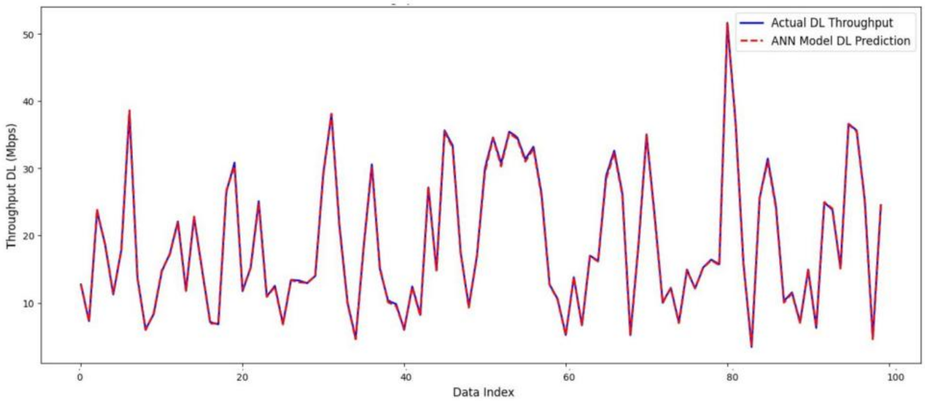


Figure 3. Prediction results of ANN model vs actual data for the downlink direction for index 0 to 100.

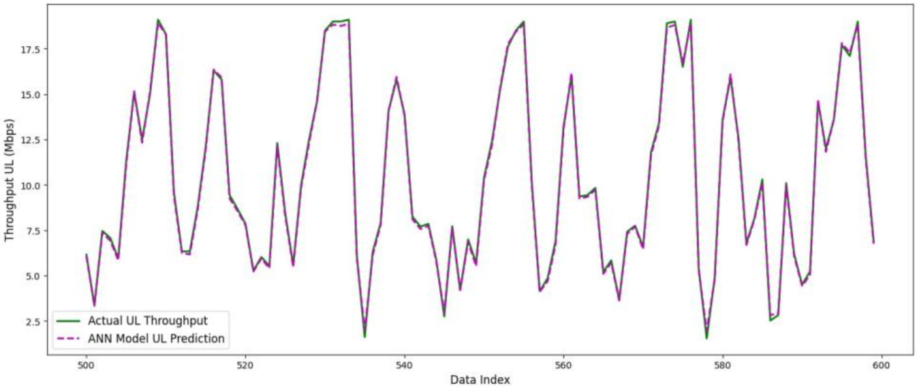


Figure 4. Prediction results of ANN model vs actual data for the uplink direction for index 500 to 600.

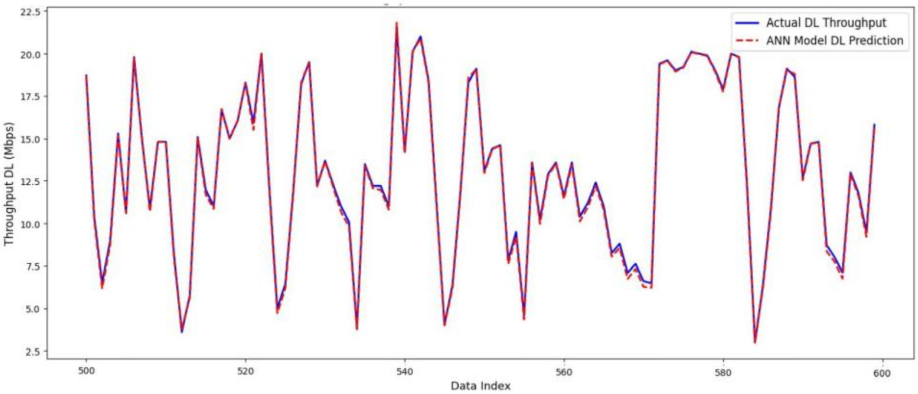


Figure 5. Prediction results of ANN model vs actual data for the downlink direction for index 500 to 600.

2.4. Throughput Prediction Models

As seen in studies conducted on similar datasets, model trials that started with machine learning in the field of pQoS have continued with deep learning model training. The use of models like RF, SVM, MLP, GDBT, and Recurrent Neural Network (RNN)-based models (such as LSTM) has accelerated developments in this area. Thanks to the good performance of deep learning models on time series data, in this study three models were tested (ANN, LSTM, and a hybrid model), and among them the ANN model performed the best in both real-time throughput prediction and one-step-ahead with current one throughput prediction. The model training and testing here were divided into two parts: real-time throughput prediction and one-step-ahead throughput prediction. The model architecture was designed with a limited number of layers. Increasing layers led to overfitting during initial experiments. Hyperparameter optimization was conducted using grid search. Deep learning models like ANN automatically extract features from raw data, explicit feature engineering was not required. However, an analysis was conducted using SHapley Additive exPlanations (SHAP) to see whether the model could be simplified by identifying its insignificant features.

An LSTM-based design is used in the first model. Bidirectional LSTM layers with residual connections are part of it. To function better in recurrent networks, each block has layer normalization (Ba, Kiros, & Hinton, 2016). Dropout is included for consistency. The structure is made up of four stacked LSTM blocks, with the later layers gradually losing units and the earlier layers having higher ones. A projection layer is used to match the dimensions in the residual connections. After the recurrent blocks, fully connected dense layers are added. These also use residual links to improve learning. The model is trained with the Adam optimizer (learning rate 0.0001). L1L2 regularization is applied to both recurrent and dense layers.

The second model is a feed-forward neural network. Input data is first flattened and then passed through a series of dense layers using Gaussian Error Linear Unit (GELU) (Hendrycks & Gimpel, 2016) activation, chosen by trial and error. Each dense block is followed by layer normalization and dropout. A residual connection is added by matching the flattened input size to an intermediate dense layer output. This model learns static patterns without temporal memory. L1L2 regularization is applied again. It is trained using the same loss function and optimizer as the LSTM model. It was trained over 32 epochs based on experiments.

Convolutional layers, Bi-LSTM blocks, and multi-head attention are all combined in the third model's hybrid architecture. Convolutional and recurrent blocks in this model process the input simultaneously. Residual connections are again used in both paths. After these stages, an attention layer is applied to capture global dependencies with 6 heads. The outputs from the CNN and attention-enhanced Bi-LSTM branches are merged and passed through additional convolutional and dense layers. Layer normalization and dropout are included throughout. This model benefits from local feature extraction, temporal learning, and attention mechanisms together. It was trained for 16 epochs. Training is performed on GPU using the same configuration as the other models. Architecture of the hybrid model demonstration can be seen in Figure 6.

All three models are evaluated using MSE loss and standard regression metrics: MAE, RMSE, and R². EarlyStopping and ReduceLROnPlateau callbacks are used in all cases.

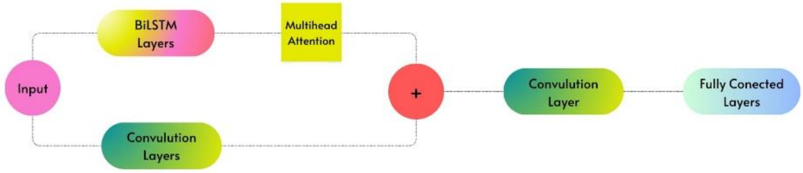


Figure 6. Architecture demonstration of hybrid model.

3. Results and Discussion

The uplink (UL) and downlink (DL) predictions were trained using the same models, their performance metrics are also reported independently. For this reason, two separate result tables are presented: Table 2 and Table 4 for uplink direction model metrics, and Table 3 and Table 5 for downlink direction model metrics.

In addition, Figure 2 and Figure 4 present the actual and predicted UL direction throughput values, and Figure 3 and Figure 5 present the actual and predicted DL direction throughput values of the ANN model over an index of the dataset with previous historical measurement together with the current one case.

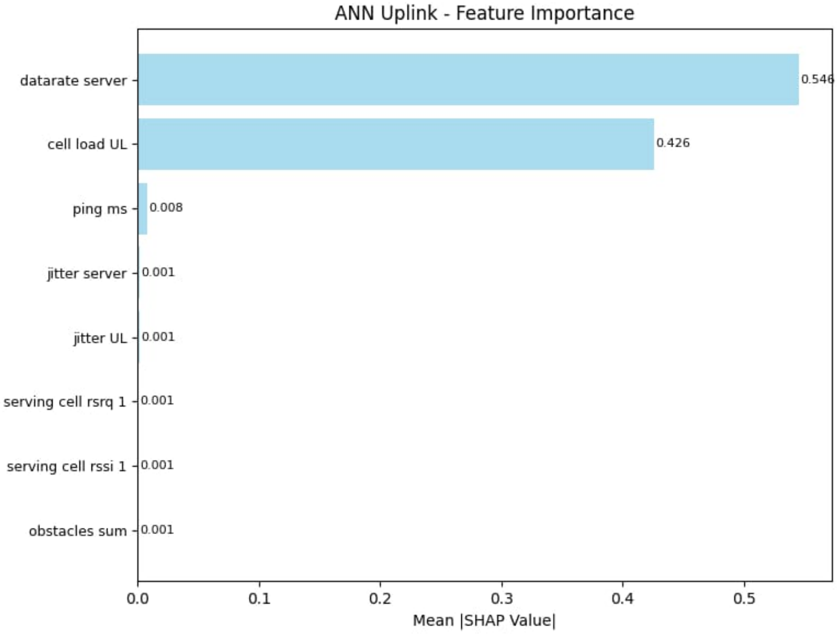


Figure 7. Feature importance from SHAP analysis for ANN uplink throughput prediction model.

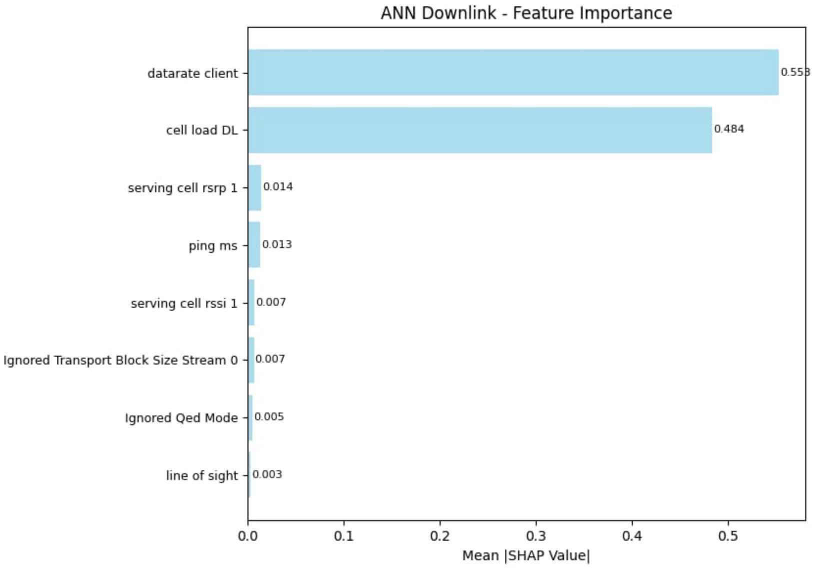


Figure 8. Feature importance from SHAP analysis for ANN downlink throughput prediction model.

In a study conducted on the same dataset (Prokop et al., 2023), the RMSE values obtained through training with previous and current data were reported as 0.1 for the downlink and 1.6 for the uplink. Their model has a more complex architecture than our proposed ANN model. The RMSE value of 1.6 on the uplink indicates that further improvement is needed, especially when compared to our ANN model which achieved an RMSE of 0.18 for uplink. Additionally, since the authors did not report ARE95 and PARE10 metrics, a direct comparison in that area was not possible. However, based on the survey paper by (Omar Basit, 2024), it is evident that the models discussed there show relatively poor performance on ARE95 and PARE10, whereas our proposed model demonstrates strong performance on these metrics.

Also, our proposed ANN model performs better in the uplink direction and has a simple architecture in contrast to the study by (Prokop et al., 2023). In the conducted model tests, the ANN model provided the highest accuracy and ranked first in terms of inference speed, with the hybrid model coming in second place and the BiLSTM model last. The ANN model's higher accuracy highlights the advantage of a single architecture rather than more than one model for throughput prediction. Comparison between models in RMSE values can be seen in **Hata! Başvuru kaynağı bulunamadı.**. Due to the dataset being created in a single environment, testing on different environments and scenarios was not possible. This represents a limitation for generalizability.

ANNs and other deep learning models can recognize intricate patterns in unprocessed data without the need for feature engineering. However, network operators must comprehend which network parameters impact throughput in order to use industrial AGVs. For this reason, SHAP analysis was applied to the best performing ANN model. As shown in Figure 7 and Figure 8, datarate parameters have the highest individual importance, followed by cell load parameters. Individual contributions of other features were low. However, performance drastically decreased when the ANN model was retrained with just these top features. RMSE increased from 0.43 to 2.87 for downlink and from 0.16 to 1.71 for uplink. This result shows that AGV network features work together in complex ways rather than separately. SHAP analysis assumes features are independent, but this is not true for AGV networks. Signal quality, network load, etc. all connect to each other. This explains why ANN models are needed for AGV throughput prediction; they can detect complex feature interactions that SHAP misses. The results show that all network parameters together improve throughput prediction, meaning full monitoring systems are better for AGV networks than tracking single metrics.

4. Conclusion

Throughput prediction plays a critical role in optimizing data transmission in vehicle-to-infrastructure communication. Accurate estimation of uplink and downlink throughput enables better decision-making for AGVs, such as remote control. In this context, we evaluated the performance of three different models---LSTM, ANN, and a hybrid CNN-BiLSTM model---on throughput prediction using historical and current measurement data and additionally instantaneous prediction only. Results show that the ANN model outperforms both LSTM and hybrid across all evaluated metrics and has a simpler architecture than the model in (Prokop et al., 2023). For uplink prediction, the ANN model achieved an RMSE of 0.1825 Mbps, ARE95 of 3.8%, and a PARE10 score of 98.76% for the case of previous data with current one. In downlink prediction, it reached 0.2734 Mbps RMSE, ARE95 of 3.32%, and 99.57% PARE10. Compared to the previous study on the same dataset, our model delivers significantly lower RMSE values, particularly in the uplink case where earlier work reported an RMSE of 1.6. Moreover, since the prior study did not include metrics such as ARE95 and PARE10, our model's strong performance on these metrics provides additional evidence of its reliability and generalization. In addition to these results, real-time throughput prediction also achieved good performance. As seen, the RMSE value obtained for downlink with a simpler model structure is close to that of the more complex GRU model.

As future work, prediction of throughput can be extended to forecast with a large future throughput range, allowing systems to adapt to bandwidth changes swiftly. Additionally, predicted throughput values can be used within ROS-based systems as indicators of communication quality, especially for components that rely on internet-based information flow or location-based tasks like fingerprinting. Also, digital twin technology can be added to this study. A virtual model can test different network conditions, and digital twin enables continuous model updates with real-time data. The system can also be extended to predict throughput for multiple AGVs working together in the same network.

References

Afshar, S., Razavi, R., & Moshirpour, M. (2025). Closed-loop clustering-based global bandwidth prediction in real-time video streaming. IEEE Transactions on Machine Learning in Communications and Networking, 3, 448–462. https://doi.org/10.1109/TMLCN.2025.3551689

Azmin, T., Ahmadinejad, M., & Shahriar, N. (2022). Bandwidth prediction in 5G mobile networks using Informer. In 2022 13th International Conference on Network of the Future (NoF) (pp. 1–9). IEEE. https://doi.org/10.1109/NoF55974.2022.9942521

Ba, J., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv. https://arxiv.org/abs/1607.06450

Basit, O., Dinh, P., Khan, I., Kong, Z. J., Hu, Y. C., Koutsonikolas, D., Lee, M., & Liu, C. (2024). On the predictability of fine-grained cellular network throughput using machine learning models. In 2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS) (pp. 47–56). IEEE. https://doi.org/10.1109/MASS62177.2024.00018

Bernadó, L., Zemen, T., Tufvesson, F., Molisch, A. F., & Mecklenbräuker, C. F. (2012). The (in-)validity of the WSSUS assumption in vehicular radio channels. In 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (pp. 1757–1762). IEEE. https://doi.org/10.1109/PIMRC.2012.6362634

Fraunhofer HHI. (2023). AI4Mobile industrial dataset [GitHub repository]. https://github.com/fraunhoferhhi/ai4mobile-industrial

Ghasemi, A. (2018). Predictive modeling of LTE user throughput via crowd-sourced mobile spectrum data. In 2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN) (pp. 1–5). IEEE. https://doi.org/10.1109/DySPAN.2018.8610464

He, Q., Dovrolis, C., & Ammar, M. (2007). On the predictability of large transfer TCP throughput. Computer Networks, 51(14), 3959–3977. https://doi.org/10.1016/j.comnet.2007.04.013

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (GELUs). arXiv. https://arxiv.org/abs/1606.08415

Hernangómez, R., Palaios, A., Watermann, C., Schäufele, D., Geuer, P., Ismayilov, R., Parvini, M., Krause, A., Kasparick, M., Neugebauer, T., Ramos-Cantor, O. D., Tchouankem, H., Calvo, J. L., Chen, B., Fettweis, G., & Stańczak, S. (2024). Toward an AI-enabled connected industry: AGV communication and sensor measurement datasets. IEEE Communications Magazine, 62(4), 90–95. https://doi.org/10.1109/MCOM.001.2300494

Lau, I., Ekpo, S., Zafar, M., Ijaz, M., & Gibson, A. (2023). Hybrid mmWave-Li-Fi 5G architecture for reconfigurable variable latency and data rate communications. IEEE Access, 11, 42850–42861. https://doi.org/10.1109/ACCESS.2023.3270777

Liu, Y., Zhang, C., Du, J., & Qiu, T. (2022). Throughput prediction-enhanced RL for low-delay video application. In 2022 18th International Conference on Mobility, Sensing and Networking (MSN) (pp. 728–735). IEEE. https://doi.org/10.1109/MSN57253.2022.00119

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (pp. 4765–4774).

Mei, L., Hu, R., Cao, H., Liu, Y., Han, Z., Li, F., & Li, J. (2020). Realtime mobile bandwidth prediction using LSTM neural network and Bayesian fusion. Computer Networks, 182, 107515. https://doi.org/10.1016/j.comnet.2020.107515

Minovski, D., Ögren, N., Mitra, K., & Åhlund, C. (2023). Throughput prediction using machine learning in LTE and 5G networks. IEEE Transactions on Mobile Computing, 22(3), 1825–1840. https://doi.org/10.1109/TMC.2021.3099397

Molina-Masegosa, R., Avedisov, S. S., Sepulcre, M., Farid, Y. Z., Gozalvez, J., & Altintas, O. (2023). V2X communications for maneuver coordination in connected automated driving: Message generation rules. IEEE Vehicular Technology Magazine, 18(3), 91–100. https://doi.org/10.1109/MVT.2023.3284562

Na, H., Shin, Y., Lee, D., & Lee, J. (2023). LSTM-based throughput prediction for LTE networks. ICT Express, 9(2), 247–252. https://doi.org/10.1016/j.icte.2021.12.001

Prokop, K., Połap, D., & Srivastava, G. (2023). AGV quality of service throughput prediction via neural networks. In 2023 IEEE International Conference on Big Data (BigData) (pp. 2493–2498). IEEE. https://doi.org/10.1109/BigData59044.2023.10386241

Raca, D., Zahran, A. H., Sreenan, C. J., Sinha, R. K., Halepovic, E., & Gopalakrishnan, V. (2024). Device-based cellular throughput prediction for video streaming: Lessons from a real-world evaluation. IEEE Transactions on Machine Learning in Communications and Networking, 2, 318–334. https://doi.org/10.1109/TMLCN.2024.3352541

Raca, D., Zahran, A. H., Sreenan, C. J., Sinha, R. K., Halepovic, E., Jana, R., & Gopalakrishnan, V. (2020). On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges. IEEE Communications Magazine, 58(3), 11–17. https://doi.org/10.1109/MCOM.001.1900394

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. arXiv. https://arxiv.org/abs/1409.3215

Wei, B., Kawakami, W., Kanai, K., Katto, J., & Wang, S. (2018). TRUST: A TCP throughput prediction method in mobile networks. In 2018 IEEE Global Communications Conference (GLOBECOM) (pp. 1–6). IEEE. https://doi.org/10.1109/GLOCOM.2018.8647390

Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. IEEE Communications Surveys & Tutorials, 21(3), 2224–2287. https://doi.org/10.1109/COMST.2019.2904897

Zhang, J., Yang, X., Wang, W., Guan, J., Ding, L., & Lee, V. C. S. (2023a). Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering. Automation in Construction, 146, 104699. https://doi.org/10.1016/j.autcon.2022.104699

Zhang, M., Jiang, X., Jin, G., Li, P., & Chen, H. (2023b). CapRadar: Real-time adaptive bandwidth prediction for dynamic wireless networks. Computer Networks, 233, 109865. https://doi.org/10.1016/j.comnet.2023.109865