

Article

Physics-Based Hybrid Control of Mobile Robot Drives with Adaptive Neural Network Compensation

Alina Fazylova ¹, Kuanysh Alipbayev ¹, Teodor Iliev ², Fariza Oraz ¹ and Kenzhebek Myrzabekov ^{1,*}

¹ Department of Aerospace and Electronic Engineering, Almaty University of Power Engineering and Telecommunications Named After Gumarbek Daukeyev, Almaty 050013, Kazakhstan; a.fazylova@aes.kz (A.F.); k.alipbayev@aes.kz (K.A.); f.oraz@aes.kz (F.O.)

² Department of Telecommunications, University of Ruse, 8 Studentska Str., 7004 Ruse, Bulgaria; tiliev@uni-ruse.bg

* Correspondence: k.myrzabekov@aes.kz

Abstract

This paper proposes a physically based hybrid architecture for controlling mobile robot drives. It combines a model-based controller, an adaptive neural network compensator for residual dynamics, and a Lyapunov-based stability supervision mechanism. Unlike existing hybrid control approaches, the proposed architecture implements a structured injection of neural network correction directly into the physical drive model with a controlled Lyapunov-based adaptation constraint. A mathematical model of the electromechanical drive of a differential mobile platform is developed, taking into account electrical and mechanical dynamics, wheel-to-surface contact interaction, and the system's energy characteristics. Numerical simulation results demonstrate that the hybrid approach improves tracking accuracy, improves transient response, and ensures stable operation of the control system under parametric uncertainty, adhesion changes, and external disturbances. The proposed architecture maintains the physical interpretability of the model while simultaneously enhancing the system's adaptability. The obtained results confirm the effectiveness of the developed method and its potential for application in control systems for mobile robotic platforms.

Keywords: differential-drive mobile robot; electromechanical drive modeling; residual dynamics compensation; Lyapunov-based stability supervision; wheel-ground interaction; split- μ disturbance rejection

1. Introduction

Modern mobile robots are increasingly used in industrial logistics, service robotics, healthcare, inspection systems, agriculture, and autonomous transport infrastructure. Their performance is determined not only by navigation and perception algorithms but also by the quality of low-level drive control, as it is the drive system that generates actual traction forces, transient processes, and motion stability. A review by Shin et al. demonstrates that autonomous mobile robots, particularly in tracking and transport applications, require reliable integration of sensor, navigation, and control subsystems; however, drive control issues often remain secondary to the navigation level [1].

For differential mobile platforms, motion accuracy is determined by the coordinated operation of the left and right drives, the dynamics of the electric motors, the limitations of the power electronics, changes in wheel traction, and the impact of external disturbances. In a review by Rybczak et al., machine learning methods for controlling mobile robots were



Academic Editor: Dan Zhang

Received: 9 April 2026

Revised: 7 June 2026

Accepted: 12 June 2026

Published: 15 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

systematized, demonstrating that the resulting algorithms are highly adaptive. However, issues of model stability, generalizability, and interpretability remain [2]. This is particularly important for the drive level, where adaptive correction errors can directly lead to actuator saturation or deterioration of the transient response.

Model-based control remains a fundamental approach, as it allows for a direct link between the control law and the physical parameters of the robot: the resistance and inductance of the motor armature circuit, torque and back-EMF coefficients, the reduced moment of inertia, wheel radius, platform base, and surface contact parameters. Guffanti et al. emphasize that correct identification of a differential mobile robot model is the basis for odometry, control design, and navigation [3]. However, model-based approaches become less effective with changes in mass, adhesion, rolling resistance, and unaccounted nonlinearities.

Robust control methods allow for partial compensation of uncertainties and external disturbances. Abadi et al. proposed a robust controller for a wheeled mobile robot, accounting for uncertainties, wind disturbances, and wheel slip [4]. This work confirms the relevance of robust control for mobile robots, but primarily considers the trajectory tracking level, while the electrical dynamics of the actuators, traction force generation, and adaptive neural network compensation of residual dynamics are not integrated into a single, physically interpretable framework.

Sliding mode and backstepping methods are also actively used for nonlinear control of wheeled mobile robots. Huang and Gao developed trajectory tracking controllers based on backstepping and a new sliding mode algorithm, demonstrating improved convergence and stability of motion [5]. However, such approaches are typically built around kinematic or dynamic tracking error and do not simultaneously consider the electromechanical structure of the actuator, voltage constraints, variable coupling, and a learnable residual dynamics compensator.

In recent years, much attention has been paid to adaptive and neural network control methods. Silaa et al. proposed indirect adaptive control of a wheeled mobile robot using a neural network and a discrete extended Kalman filter [6]. This approach improves the system's ability to compensate for uncertain parameters, but the neural network component remains insufficiently linked to a physically interpretable electromechanical model of the drive and does not fully address the issue of limiting the impact of neural network correction on closed-loop stability.

Physically informed neural networks and physics-guided learning are considered a way to combine the advantages of analytical modeling and trainable compensators. Liu et al. demonstrated that PINN models can be used to model and control robots while preserving theoretically based stability properties [7]. However, such approaches are more often applied to the general dynamics of robotic systems or manipulators, while their implementation directly into the drive loop of a differential mobile platform remains underdeveloped.

Sun et al. developed an approach by proposing a physically informed model for predicting robot dynamics, taking into account the coupling of the motor and external forces [8]. This work is important for the present study, as it demonstrates the need to consider not only the abstract dynamics of motion but also the interaction of the drive system with external forces. However, it does not provide a complete architecture for controlling mobile robot drives with actuator constraints, a contact wheel model, and Lyapunov-supervision adaptive neural network correction.

A review by Farea et al. shows that physically informed neural networks have significant potential for improving the physical consistency and generalization ability of trained models [9]. However, the authors also note open issues related to computational complexity, the selection of physical constraints, and the robustness of trained models. For

mobile robots, this means that neural network correction should not simply be added to the control loop, but rather structurally embedded in the physical model and constrained by a supervision mechanism.

Predictive control methods allow for explicit consideration of constraints and optimization of future control actions. Li and Liu proposed a PINN-based nonlinear MPC for trajectory tracking of automated guided vehicles [10]. This approach demonstrates that physically informed learning can improve predictive control performance; however, the focus remains on the trajectory level, rather than on the internal electromechanical structure of the left and right actuators, voltage saturation, contact traction force, and online residual compensation.

Wheel-to-surface contact is a critical factor for differential robots. Li et al. examined the trajectory control of wheeled mobile robots with an unknown slip ratio [11]. Their work demonstrates that ignoring the slip ratio degrades tracking performance. However, the contact model in such studies is more often used as part of the kinematic correction rather than as a link between the drive torque, adhesion coefficient, longitudinal force, and an adaptive compensator.

Thus, the current literature demonstrates the development of several areas: model-based control, robust control, adaptive neural network methods, physics-informed learning, and slip-aware control. However, an architecture that simultaneously integrates a physically interpretable electromechanical drive model, structured neural residual compensation, a contact wheel model, actuator saturation, and Lyapunov-type supervision remains underdeveloped. This gap defines the goal of this work: to develop a physics-based hybrid control architecture for differential mobile robot drives that preserves the physical interpretability of the model and improves adaptability under uncertainty, adhesion changes, and external disturbances [12].

2. State of the Art and Research Gap

Modern research in the field of mobile robot control can be divided into several areas: neural network control, hybrid intelligent controllers, robust and adaptive control, model predictive control, slip compensation, dynamics identification, and navigation and path planning methods. However, most existing studies address only specific aspects of the overall problem: either improving trajectory tracking, compensating for uncertainty, developing a neural network component, or considering motion planning. However, a comprehensive architecture integrating an electromechanical drive model, a physically interpretable base controller, neural network compensation for residual dynamics, actuator constraints, wheel-surface contact, and Lyapunov-type supervision remains underdeveloped.

Leal et al. examined the control of a line-following mobile robot using neural networks [13]. This work demonstrates the applicability of neural network control in simple navigation scenarios. However, the line-following problem is significantly simpler than controlling differential platform drives under variable adhesion conditions. It does not consider the electromechanical dynamics of the motor, the relationship between current, torque, and longitudinal force, the limitations of actuators, and the robustness of neural network correction. This paper addresses these limitations by moving from a simple navigation problem to a physically interpretable drive-level hybrid control architecture.

Szeremeta and Szuster proposed neural network tracking control for a four-wheeled mobile robot with Mecanum wheels [14]. A strength of this work is the application of neural tracking control to a mobile platform with complex kinematics. However, the Mecanum platform differs from a differential-drive robot in its mechanism for distributing velocities and contact forces. Furthermore, the primary focus is on the tracking problem, while the electromechanical structure of the drives, actuator saturation, and Lyapunov-supervised

residual compensation are not central elements. This paper focuses specifically on the differential-drive platform, where the adhesion asymmetry between the left and right wheels directly affects yaw deviation and control quality.

Razali et al. proposed a hybrid controller with genetic algorithm optimization for position and angular motion control of a mobile robot [15]. The study demonstrates that combining classical control and intelligent optimization can improve motion performance. However, this hybrid approach primarily involves selecting controller parameters and does not provide a physically interpretable actuator control architecture. It does not consider the residual dynamics of the electric drive, the neural network residual compensator, the wheel contact force, or the adaptive response constraint based on the stability criterion. In this paper, the hybrid structure is based not on empirical parameter optimization, but on the separation of a physical model-based controller and bounded neural residual correction.

Xu and Wu investigated adaptive learning control for robotic manipulators using an incremental hybrid neural network [16]. This work is important as an example of hybrid neural network learning in robotic systems. However, the object of the study is manipulators, not mobile platforms. Therefore, it lacks factors fundamental to differential-drive robots: wheel-ground interaction, longitudinal slip, left-right drive asymmetry, and split- μ disturbances. In this paper, the idea of hybrid neural learning is transferred to the drive level of a mobile platform, where uncertainty arises not only from model parameters but also from the changing contact of the wheels with the surface.

Nicodemus et al. proposed a physics-informed neural network-based model of predictive control for multi-link manipulators [17]. The study confirms the feasibility of using PINN models in predictive control. However, the scope of application is limited to manipulation systems that do not face the problems of wheel slip, adhesion coefficient, and asymmetric generation of longitudinal forces by the left and right wheels. In contrast, this paper applies the physics-guided idea to an electromechanical model of mobile robot drives and introduces neural network correction as residual compensation within the physical model.

Pinosky et al. considered hybrid control combining model-based and model-free reinforcement learning [18]. This work demonstrates the advantages of combining a physical model and a trainable component. However, model-free reinforcement learning can reduce interpretability and requires careful constraints on behavior in modes other than training ones. The present study uses a more engineering-controlled version of the hybrid architecture: the neural network unit does not replace the base controller, but is introduced as a bounded residual correction with Lyapunov-type supervision.

Chai et al. proposed an optimal robust trajectory tracking control strategy for a wheeled mobile robot [19]. This work aims to improve the robustness of trajectory tracking and convergence under uncertainties. However, robustness is considered primarily at the platform motion level, while the electromechanical dynamics of the motor, the relationship between current, torque, longitudinal force, and surface adhesion are not integrated into a single drive-level model. This paper extends this approach by considering not only trajectory tracking but also physically interpretable drive control with adaptive neural residual compensation.

Peng et al. developed a nonlinear disturbance observer incorporated model predictive strategy for trajectory tracking of a wheeled mobile robot [20]. The advantage of this work is that it explicitly accounts for external disturbances in the predictive control framework. However, the MPC approach typically focuses on the trajectory or vehicle-level control, and neural network residual compensation embedded in the physical drive model is not a central element. In this paper, adaptive correction is implemented in a more compact

hybrid architecture, where a model-based controller provides basic stability and a neural compensator compensates for residual dynamics.

Guffanti et al. proposed robust model predictive control with dynamic look-ahead distance for a four-wheel differential-drive mobile robot [21]. This work is strong in that it relates robust MPC to a practical implementation for a differential-drive platform. However, its focus remains on trajectory tracking, rather than on the internal structure of the electromechanical drive or the integration of neural residual compensation with the wheel-ground contact model. This paper complements this approach by focusing on the drive level and explaining how a physical control channel and a neural network correction channel can work together without losing interpretability.

Xu et al. examined dynamic model predictive control for a three-wheel independent drive and steering mobile robot [22]. This work demonstrates the effectiveness of MPC for mobile platforms with complex architectures of independent drives and steering. However, the object of study differs from the classical differential-drive platform, and neural residual correction and Lyapunov-type supervision are not considered. This article focuses on a more common differential motion scheme, which requires coordinated control of the left and right drives in response to adhesion changes and external disturbances.

Korayem et al. proposed a hybrid stable adaptive control approach for navigating nonholonomic wheeled mobile robots on slippery surfaces with obstacles [23]. This work is related to the topic under consideration, as it takes slippery surfaces and adaptive control into account. However, the primary focus is on navigation and stable motion in the presence of obstacles, rather than on structured neural network compensation for residual dynamics within an electromechanical drive model. In this paper, slippery and split- μ conditions are considered through a contact-aware drive model, and adaptive neural correction is limited to a Lyapunov-type supervision mechanism.

Boubaker et al. developed trajectory tracking of WMR with neural adaptive correction [24]. This study is important because it directly uses neural adaptive correction to improve the accuracy of a wheeled mobile robot. However, the neural network compensator in such studies is often considered as a separate adaptive unit, rather than as part of a physically interpretable electromechanical model. Furthermore, the question of the permissible influence of neural network correction on error energy and closed-loop stability remains insufficiently addressed. This work addresses this shortcoming by limiting neural network influence and inhibiting adaptation as Lyapunov-type error energy increases.

Tang et al. proposed motion/force-coordinated trajectory tracking control for a non-holonomic wheeled mobile robot [25]. This work is important in that it considers the coordination of motion and force, which is closer to the drive level than purely kinematic control. However, it does not develop a physics-based hybrid neural architecture, where residual dynamics are compensated by a neural network module integrated into the drive model. In this paper, force-related behavior is included through a wheel-ground contact model, and neural correction is used to compensate for unaccounted dynamic effects.

Teji et al. proposed an adaptive longitudinal slip compensation framework for wheeled mobile robots [26]. The study demonstrates that longitudinal slip significantly impacts tracking performance and motion efficiency. However, the method focuses primarily on slip compensation rather than integrating the slip model with electromechanical motor dynamics, actuator saturation, and neural residual compensation. In the present study, the slip ratio and adhesion coefficient are included in the contact layer of the model and are directly related to the actuator torque and longitudinal force.

Gameiro et al. evaluated PID-based algorithms for UGVs [27]. This work is useful as a baseline, as it demonstrates that PID-based algorithms are still applicable to the control of ground-based robotic platforms. However, the PID approach has limited adaptability

to nonlinear dynamics, adhesion variations, actuator saturation, and unaccounted disturbances. In this paper, classical control is considered only as a baseline stabilizing layer, which is enhanced by a neural residual compensator and supervision mechanism.

Hoseinnezhad et al. presented a comprehensive review of deep learning techniques in mobile robot path planning [28]. The review demonstrates the active development of deep learning in path planning problems. However, path planning is a high-level control technique and does not address the problem of physical trajectory execution at the actuator level. Even a well-planned trajectory can be poorly implemented due to actuator saturation, wheel slip, or adhesion changes. This paper complements path planning research by developing a low-level adaptive drive-control layer.

Zhu et al. performed a comparative analysis of deep reinforcement learning for mobile robot navigation in dynamic environments [29]. This work demonstrates the promise of DRL for navigation in complex dynamic environments. However, DRL approaches often require large amounts of data, have limited interpretability, and do not provide explicit guarantees of drive loop stability. In this paper, the learnable component is not used as a global policy, but is limited to the role of a residual compensator within a physically interpretable control model.

Ait Dahmad et al. proposed an adaptive model of predictive control for a 4WD-4WS mobile robot with trajectory tracking and obstacle avoidance [30]. This work demonstrates the effectiveness of advanced MPC for mobile platforms with all-wheel drive and steering. However, its focus is on trajectory tracking and obstacle avoidance, not on neural residual learning or differential-drive actuator-level uncertainty. This paper addresses a different challenge: creating a physically interpretable hybrid architecture for differential drives, where left–right torque and adhesion asymmetry are key factors.

Aremu et al. conducted a systematic review of autonomous mobile robot path planning techniques [31]. This work is important as an overview of modern path planning methods, including heuristic, learning-based, and hybrid approaches. However, planning-level research does not address the issue of executing control commands at the electric drive level with variable adhesion and actuator constraints. This paper complements this approach by developing a physically interpretable adaptive drive-control layer that can operate below the trajectory planning level.

Al Mahmud et al. presented a comprehensive review of advances and challenges in mobile robot navigation [32]. The work systematizes current achievements and unresolved problems in mobile navigation. However, the navigation layer does not eliminate the need for a robust drive circuit capable of executing motion commands under adhesion changes, actuator saturation, and the presence of unaccounted dynamics. This study proposes precisely such a lower-level physics-based hybrid drive control with neural residual compensation and Lyapunov-type supervision.

Thus, a critical analysis of previous works [13–32] shows that existing research addresses separate, but not unified, challenges. Some studies develop neural network control and adaptive correction, others robust control and MPC, and still others slip compensation, force coordination, PID baselines, navigation, or path planning. However, a unified architecture that combines electromechanical drive dynamics, contact-aware force generation, actuator saturation, structured neural residual compensation, and Lyapunov-type supervision into a single, reproducible framework remains underdeveloped. This research gap is closed in this paper by using a physics-based hybrid control architecture for differential mobile robot drives.

3. Materials and Methods

This paper uses a hybrid control methodology for a mobile robot’s electromechanical drive. This methodology combines a physically interpretable dynamics model, a classical stabilizing controller, a neural network adaptive compensator, a Lyapunov-based stability supervision mechanism, and hardware constraints on the actuators. The proposed method is designed to operate under conditions of parametric uncertainty, varying loads, varying surface contact characteristics, and the presence of unmodeled disturbances, while maintaining formal guarantees of closed-loop stability.

The physical structure of the control system and the placement of the computing modules are shown in Figure 1.

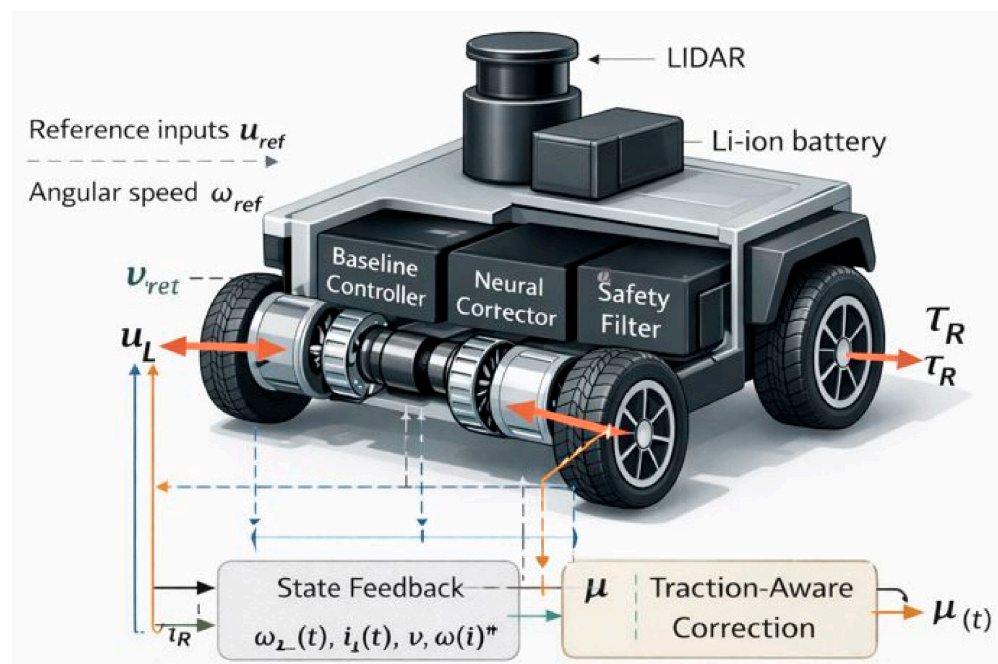


Figure 1. Structural diagram of a hybrid mobile robot drive control system.

The platform under consideration belongs to the class of differential mobile robots with independent drives for the left and right wheels. Each drive consists of a DC electric motor, a gearbox assembly, and a wheel module. The control architecture is implemented in a distributed manner and is logically divided into three computing units: a physically based base controller, a neural network compensator, and a safety filtering module. The system's control inputs are reference signals for the platform's linear and angular velocity $v_{ref}(t)$ and $\omega_{ref}(t)$, generated at the trajectory planning level. The output effects are the drive torques $\tau_L(t)$ and $\tau_R(t)$, generated at the engine power control level.

To account for interaction with the supporting surface, a friction correction module is used, generating an estimate of the effective friction coefficient $\mu(t)$ based on an analysis of the kinematic parameters of motion, drive currents, and wheel speeds; this assessment is used to adapt control actions and compensate for degradation of traction characteristics under changing contact conditions.

The overall architecture of the hybrid control loop is shown in Figure 2. The superscript * denotes the desired reference value. Thus, $v_{ref}^*(t)$ and $\omega_{ref}^*(t)$ represent the target linear and angular velocities generated by the trajectory-planning level and supplied to the hybrid control loop.

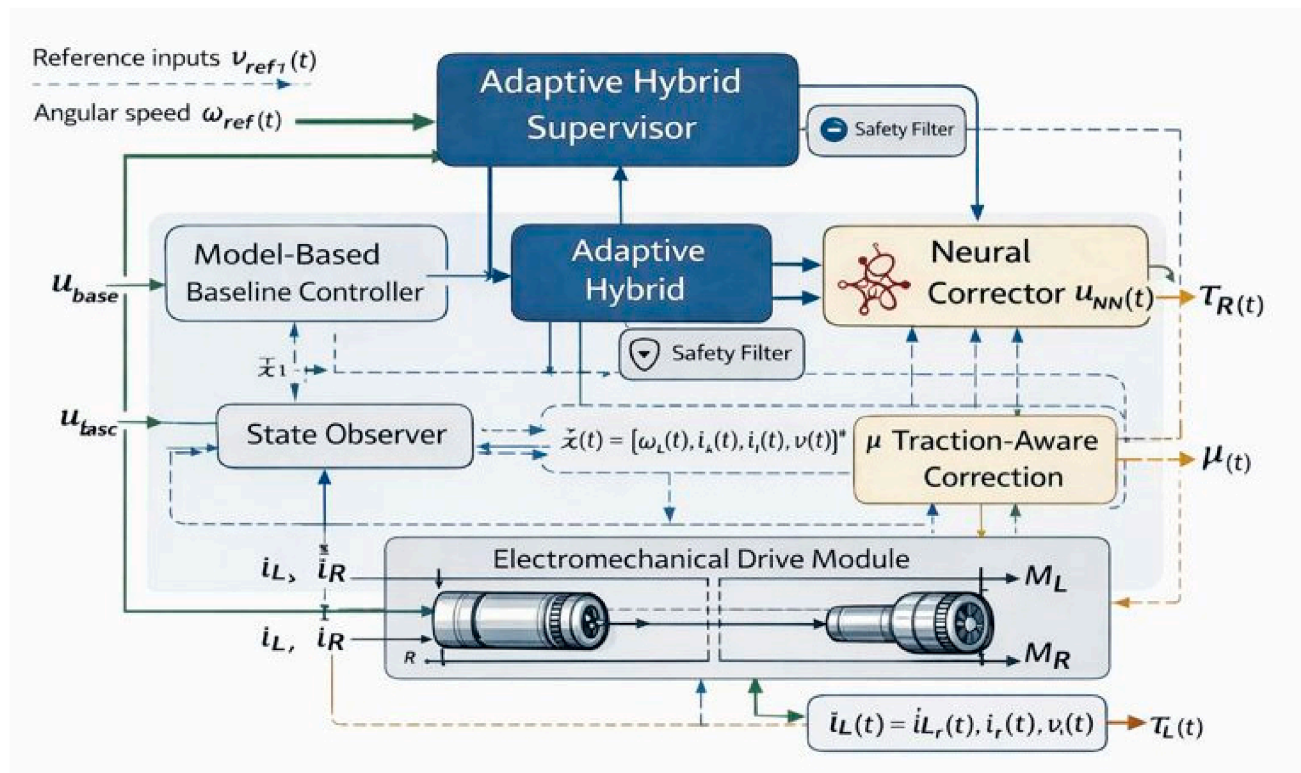


Figure 2. Hybrid control loop architecture with neural network compensation.

To clarify the relationship between the navigation layer, the control architecture, and the dynamic model used in the simulations, the complete workflow of the proposed navigation and drive-control process is introduced in Figure 3. The workflow starts from the reference trajectory or velocity command generated by the navigation layer. At each sampling instant, the observer forms the feedback signal vector from the measured or estimated wheel speeds, motor currents, and platform velocity. These signals are supplied simultaneously to the traction-estimation module, the physically based controller, and the neural network residual compensator.

The traction-estimation module evaluates the effective wheel-ground interaction conditions, including the slip ratio and adhesion-related variables. The model-based controller generates the nominal control command using the electromechanical drive model, while the neural-network compensator estimates the residual correction associated with unmodeled dynamics, parameter variations, and external disturbances. The Lyapunov-type supervision block evaluates whether the neural correction is admissible in terms of error-energy behavior. If the correction tends to increase the tracking-error energy, the neural update is temporarily blocked or attenuated.

The final control command is then formed by combining the nominal model-based control action and the admissible neural correction. Before being applied to the actuators, the command passes through the saturation and safety-filtering block to ensure compliance with voltage and torque constraints. The resulting left and right drive commands are applied to the mobile robot drive model. The updated robot state is then used by the navigation layer and by the feedback loop at the next time step. This workflow establishes a direct connection between the navigation process, the hybrid controller, and the mathematical model used for numerical simulation.

The control loop is organized as a multi-level system with functions separated between a physically interpreted stabilization loop and an adaptive neural network module. The base controller generates the nominal control action $u_{base}(t)$ based on a mathematical model

of the electromechanical drive and the current system state vector. The neural network compensator generates an additional correction signal $u_{NN}(t)$, designed to compensate for residual dynamics, unmodeled nonlinearities, variations in engine parameters, and external disturbances. The adaptive supervisor manages the interaction of these two channels, ensuring signal matching and preventing unstable operating conditions.

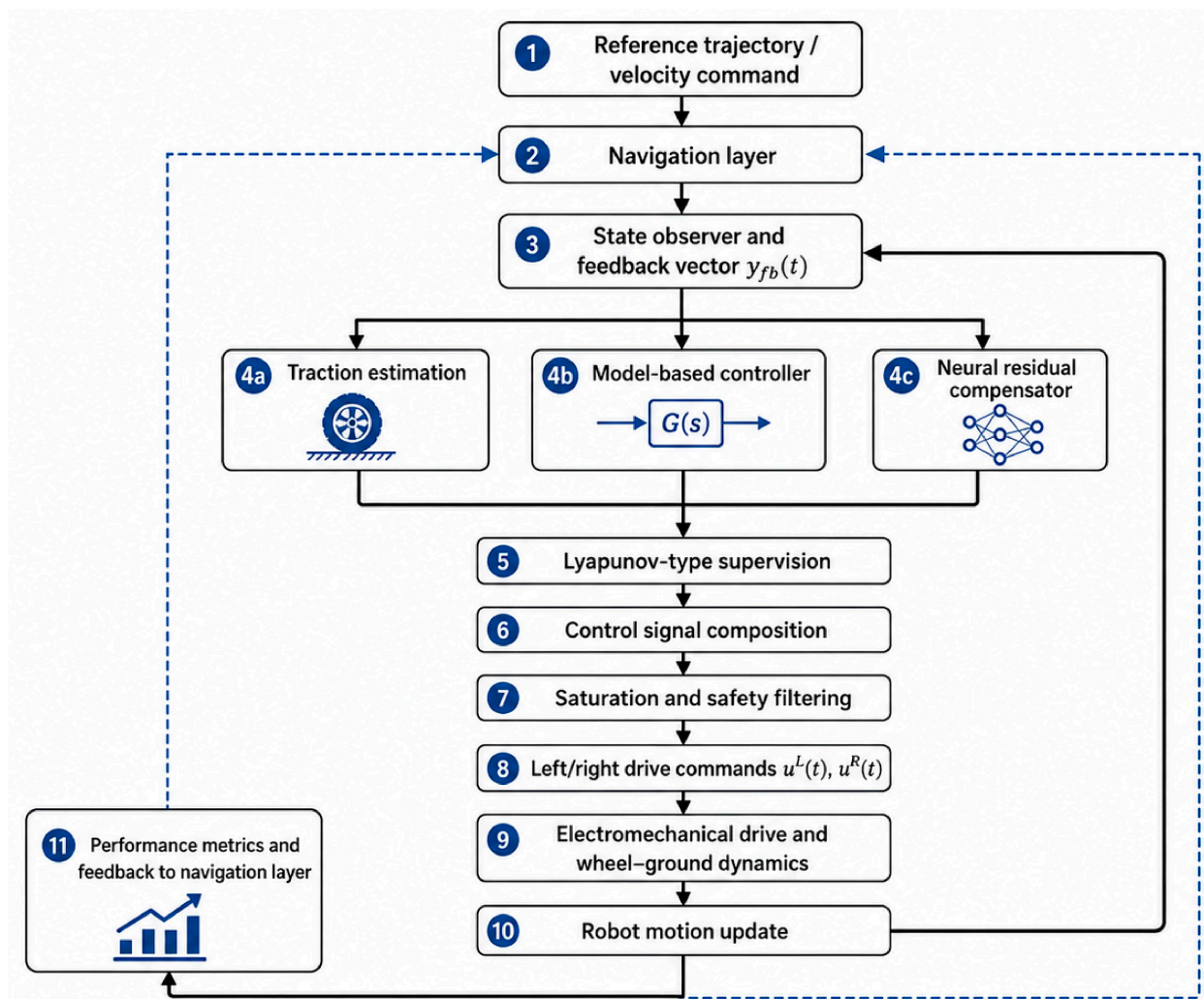


Figure 3. Detailed workflow of the proposed navigation and drive-control process.

To generate the input data for the control algorithm, a state observer is used. Its output is treated as a feedback signal vector rather than as the full state vector of the dynamic model:

$$y_{fb}(t) = [\omega_L(t), \omega_R(t), i_L(t), i_R(t), v(t)]^T \quad (1)$$

where $\omega_L(t)$ and $\omega_R(t)$ are the angular velocities of the left and right wheels, $i_L(t)$ and $i_R(t)$ are the left and right motor currents, and $v(t)$ is the linear velocity of the platform.

For consistency, the same left–right notation is used throughout the manuscript: $u_L(t)$ and $u_R(t)$ denote the left and right motor input voltages, $\tau_L(t)$ and $\tau_R(t)$ denote the generated drive torques, and $F_{xL}(t)$ and $F_{xR}(t)$ denote the longitudinal contact forces acting on the left and right wheels. All variables are functions of time unless otherwise stated.

The vector $y_{fb}(t)$ is used as the feedback input for the control loop, the traction-estimation module, and the neural network compensator. It is not used as the full state vector for

numerical integration. The complete state vector of the dynamic model is introduced separately in Section 3.

A mathematical model of the electromechanical drive is used as the physical basis of the algorithm. The electrical dynamics of the motor are described by the expression:

$$L_a \frac{di(t)}{dt} + R_a i(t) + k_e \omega(t) = u(t) \quad (2)$$

where L_a is the armature inductance, R_a is the armature resistance, k_e is the back-EMF coefficient, $\omega(t)$ is the rotor angular velocity, and $u(t)$ is the motor input voltage.

The mechanical dynamics are described by the equation:

$$J_{eq} \frac{d\omega(t)}{dt} = k_t i(t) - \tau_{load}(t) - b\omega(t), \quad (3)$$

where J_{eq} —the reduced moment of inertia, k_t —the electromagnetic torque coefficient, $\tau_{load}(t)$ —the external load torque, and b —the viscous friction coefficient.

This model is used both in the control loop and in generating training signals for the neural network module.

A cascade speed controller is used as the stabilizing base controller. The control error is defined as:

$$e(t) = \omega_{ref}(t) - \omega(t). \quad (4)$$

The control action of the base circuit is formed according to [33]:

$$u_{base}(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau. \quad (5)$$

The controller coefficients are selected based on the drive's dynamic characteristics, current and voltage limits, and performance and stability requirements. Tuning is performed to ensure aperiodic transient response and limited overshoot during stepwise changes in the setpoint. The neural network compensator implements an adaptive approximation of the system's residual dynamics and generates a corrective action:

$$u_{NN}(t) = f_{NN}(x(t)), \quad (6)$$

The function $f_{NN}(x)$ is assumed to be continuous and piecewise continuously differentiable with respect to its input arguments. This assumption follows from the adopted feedforward neural network structure with ReLU activation functions, which provides a continuous and piecewise smooth approximation. In addition, $f_{NN}(x)$ is considered locally bounded on the compact operating domain defined by the admissible ranges of wheel speeds, motor currents, tracking errors, and actuator limits. A fully connected architecture with two hidden layers, each containing 16 neurons, is used. This structure provides sufficient approximation power while maintaining computational efficiency and the ability to be implemented in real time. The parameters of the neural network compensator are updated online, synchronously with the numerical integration of the dynamic model at a fixed frequency of 200 Hz. The parameters of the neural network compensator are updated using the Adam algorithm with a learning coefficient of $\eta = 0.002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, ensuring robust online adaptation of the parameters [34]. The loss function is formed based on the squared tracking error for the platform's speed and position. The neural network compensator is trained online by minimizing the following tracking-error-based criterion:

$$J_{NN}(W, t) = 1/2 e^T(t) e(t) \quad (7)$$

where $J_{NN}(W, t)$ is the neural-network training criterion, W denotes the trainable weights of the neural compensator, and $e(t)$ is the tracking-error vector. This notation is used to avoid conflict with the armature inductance L_a in the motor model.

To prevent unstable adaptation modes, a supervision mechanism based on a Lyapunov-type error-energy function is used. In the revised formulation, the candidate function is written in a simple quadratic form without an additional weighting matrix:

$$V(t) = 1/2 e^T(t) e(t) \quad (8)$$

This form was selected because the supervision mechanism uses the scalar error energy only as an online admissibility criterion for neural network adaptation, rather than as a full analytical Lyapunov proof with a weighted quadratic metric. The time derivative $\dot{V}(t)$ is estimated numerically and used to decide whether the neural network weights may be updated. The update is allowed only when the adaptive correction does not increase the error energy, that is, when $\dot{V}(t) \leq 0$. If this condition is violated, the learning process is temporarily blocked to prevent the destabilizing growth of the tracking error.

The adaptation process and the stability supervision mechanism are shown in Figure 4.

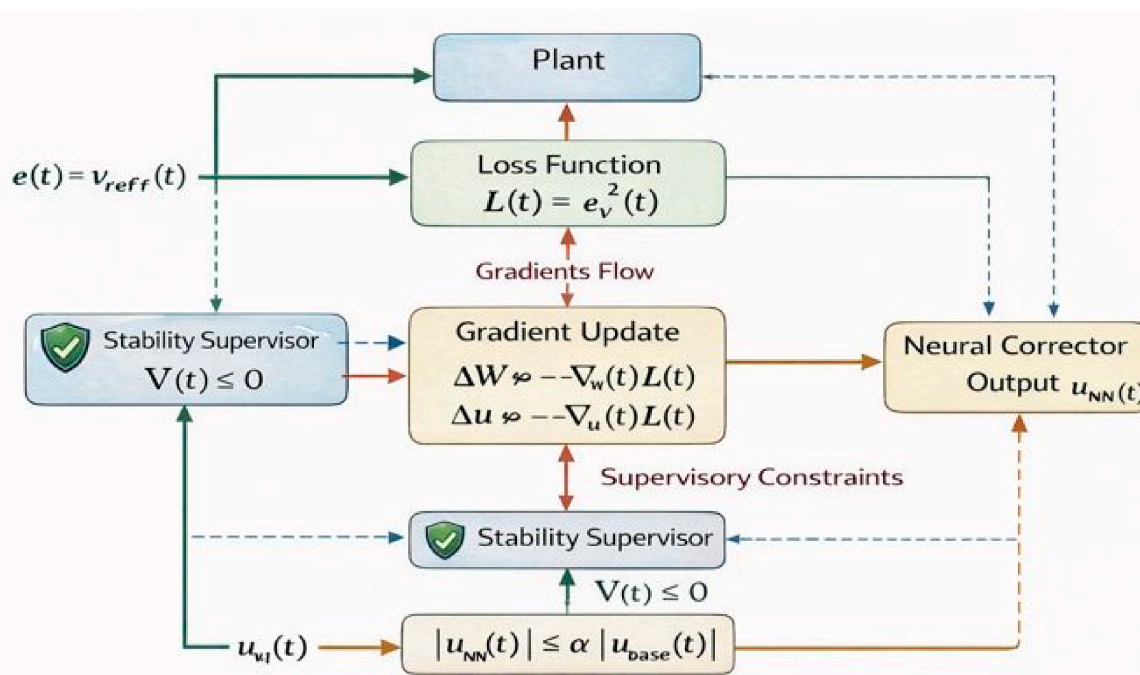


Figure 4. Adaptation mechanism of the neural network compensator and stability supervision.

The Lyapunov-type error-energy function used by the supervision mechanism has already been defined in Equation (8). Therefore, Figure 4 illustrates not a separate stability criterion, but the implementation of the same admissibility rule in the online adaptation loop. The safety factor was chosen equal to 0.4, which ensures the dominant role of the physically interpretable controller and prevents uncontrolled amplification of the neural network effect. The initial values of the neural network weights were initialized with small random variables from a uniform distribution $[-0.05; 0.05]$, which ensures a smooth start of adaptation without jumps in control actions [35]. Thus, learning is blocked in those modes where the neural network contribution can lead to an increase in the system error energy. This allows for the integration of adaptive learning without violating the stability of the closed loop. The composition of the control signal and the implementation of physical constraints are presented in Figure 5.

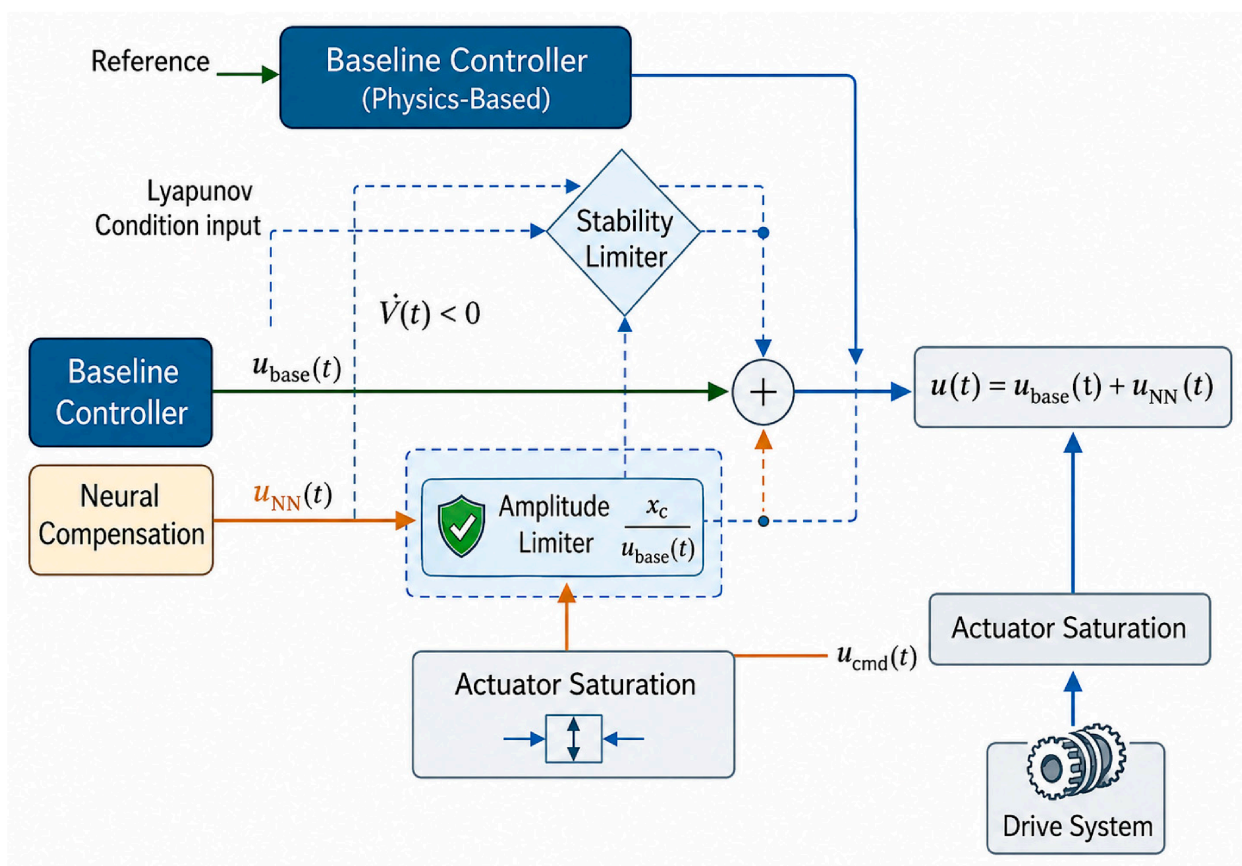


Figure 5. Generating a control signal taking into account the physical limitations of the drives.

The final control action is generated as follows:

$$u(t) = u_{base}(t) + u_{NN}(t). \quad (9)$$

To prevent excessive influence of the neural network channel, amplitude limiting is used:

$$|u_{NN}(t)| \leq \alpha |u_{base}(t)|, \quad (10)$$

where α is a safety factor that determines the permissible portion of adaptive correction.

After summing the signals, a hardware saturation block is applied:

$$u_{cmd}(t) = sat(u(t)), \quad (11)$$

This ensures that the control actions comply with the physical limitations of the actuators and prevents overload of the power electronics.

The numerical implementation of the proposed method was performed in Python 3.10 using the NumPy, SciPy, and PyTorch libraries. The differential equations are integrated using the fourth-order Runge–Kutta method with a 1 ms sampling step. The neural network weights are updated using the adaptive Adam algorithm with a limited learning step to prevent instability. To ensure reproducibility and comparability with existing research in the field of mobile robotics, the simulation environment parameters were set based on typical physical ranges presented in the literature for medium-sized differential mobile platforms. Since the primary goal of this study is to evaluate the control architecture rather than optimize a specific hardware implementation, nominal reference parameter values were adopted. The robot mass was set to 35 kg, which corresponds to widely used research mobile platforms for indoor and outdoor use. The wheel radius was

set to 0.10 m, which is within the standard range of wheel sizes for commercial mobile robots [36]. A hybrid component was additionally allocated. The input control limit of the drive was set at $u_{max} = \pm 12$ V, which corresponds to the typical limitations of low-voltage integrated electric motor drivers. The wheel–surface interaction parameters were varied to simulate different driving conditions. In the nominal mode, the adhesion coefficient was assumed to be $\mu = 0.8$, which corresponds to dry concrete or asphalt. For scenarios with disturbances, split- μ conditions were simulated with asymmetric adhesion values in the range $\mu \in [0.2, 0.4]$ for the side with reduced traction, which is typical for wet or dirty surfaces. The sampling period of the control loop was set equal to $T_s = 1$ ms. This ensures sufficient throughput for the hybrid neural network compensator and stability supervision mechanism [37]. All computational experiments were conducted with identical discretization and numerical integration settings, ensuring a fair comparison of the baseline and proposed control architectures.

The selected parameters enable systematic comparative modeling while maintaining physical realism and compliance with commonly used mobile robotics test configurations. The main physical and numerical parameters used in the study are listed in Table 1 [38]. All reference values were chosen to match a medium-scale mobile robot and provide realistic constraints on actuators and contact interactions.

Table 1. Simulation and physical benchmark parameters used in the numerical study.

Parameter	Symbol	Value	Unit	Description/Source Justification
Robot mass	m	35	kg	Representative medium-size differential mobile robot (indoor/outdoor class)
Wheel radius	r	0.10	m	Typical wheel size for research platforms
Track width (wheel separation)	L	0.42	m	Standard geometry for differential drive stability
Maximum control input	u_{max}	± 12	V	Low-voltage embedded motor driver constraint
Nominal friction coefficient	μ_{nom}	0.80	–	Dry concrete/asphalt surface
Low-adhesion friction coefficient	μ_{low}	0.20–0.40	–	Wet/contaminated surface (split- μ scenarios)
Sampling period	T_s	0.001	s	High-bandwidth hybrid control loop
Simulation step size	Δt	0.001	s	Numerical integration resolution
Rolling resistance force	F_{rr}	8	N	Typical rolling resistance approximation
Gravity acceleration	g	9.81	m/s ²	Standard gravitational constant
Neural correction gain bound	α	0.3	–	Stability-supervised NN output scaling
Baseline controller bandwidth	T_v	0.18–0.25	s	Typical velocity loop time constant
Monte-Carlo runs	N	30	–	Statistical robustness evaluation
Robot mass	m	35	kg	Representative medium-size differential mobile robot (indoor/outdoor class)
Low-speed regularization constant	ϵ	0.01	m/s	Prevents numerical singularity in slip calculation near zero velocity

As a result, a hybrid control methodology for the electromechanical drives of a mobile robot was developed. It combines a physically interpretable dynamic model, a stabilizing model-based controller, an adaptive neural network compensator for residual dynamics, and a stability supervision mechanism based on the Lyapunov function. The proposed approach ensures consistency between the physical correctness of the model and the adaptive capabilities of the trained algorithms, allows for the consideration of parametric uncertainty, variations in adhesion conditions, and external disturbances, and creates a reproducible computational basis for further numerical analysis of the performance of the mobile robotic platform’s control system.

The proposed navigation and drive-control workflow is directly connected with the mathematical model used for numerical simulation. Each functional block of the workflow corresponds to a specific group of model variables and equations. The reference trajectory and velocity commands define the desired linear and angular motion of the platform. These commands are converted by the control system into the left and right drive inputs $u_L(t)$ and $u_R(t)$, which are then applied to the electromechanical drive model.

The state observer provides the feedback signal vector $y_{fb}(t)$, which includes the wheel angular velocities, motor currents, and platform velocity. These variables are used simultaneously by the model-based controller, the traction-estimation module, and the neural residual compensator. The same variables also define the electrical motor dynamics, mechanical drive dynamics, slip-ratio calculation, wheel-ground contact model, and platform motion equations. Therefore, the navigation workflow is not treated as a separate high-level algorithm; instead, it is implemented through the same electromechanical and contact-dynamic variables that define the mathematical model.

In particular, the traction-estimation block corresponds to the slip-ratio and adhesion-model equations. The model-based controller uses the electromechanical drive equations to generate the nominal control signal. The neural residual compensator is connected with the residual dynamics term introduced in the mathematical model, while the Lyapunov-type supervision block evaluates the admissibility of neural adaptation using the tracking-error energy function. Finally, the saturation and safety-filtering block corresponds to the actuator constraints imposed on the left and right drive inputs.

Thus, the proposed workflow establishes a closed connection between the navigation command, feedback-state estimation, drive-level mathematical model, wheel-ground interaction, neural residual compensation, and actuator-constrained control implementation. This correlation ensures that all simulation scenarios are based on the same variables, constraints, and dynamic relationships.

4. Mathematical Modeling

To quantitatively evaluate the proposed hybrid control architecture, a physically interpretable dynamic model of the mobile robot's electromechanical drive was developed, supplemented by a hybrid "superstructure" mechanism (residual dynamics augmentation) and a wheel-surface interaction loop. The model is constructed using the principle of decomposition into interconnected subsystems: the electrical dynamics of the motor, the mechanical dynamics of the drive, the contact model of the traction, the dynamics of the robot body, and the computational pipeline for numerical integration. Additionally, a hybrid component (a neural network residual dynamics estimator) is introduced at a strictly defined point in the model, as well as energy and stability monitoring channels necessary for reproducible verification of stability properties and energy behavior in transient conditions. In this paper, residual dynamics are injected additively into the drive's mechanical dynamics equation via the correction torque τ_{nn} , which preserves the model's physical interpretability and prevents incorrect interference with the electrical subsystem.

Figure 6 shows the model structure, highlighting (i) the physical core (electrical and mechanical components, wheel contact, and body dynamics) and (ii) the proposed hybrid superstructure, which includes a neural network residual dynamics estimator and an adaptive correction unit $\Delta f(x)$ with a clearly defined injection point into the physical model. This approach allows us to formalize the novelty as the addition of structured residual dynamics, which does not destroy the interpretability of the physical model and provides a convenient basis for subsequent stability and energy exchange analysis.

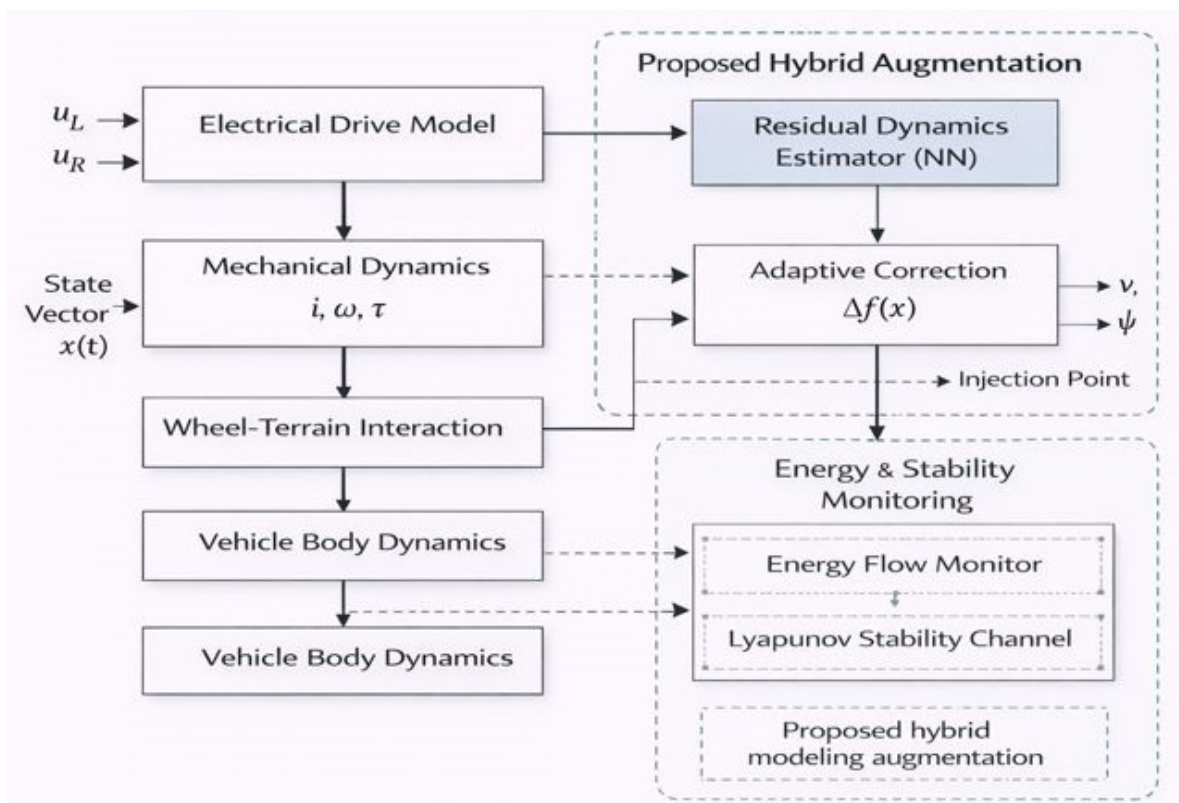


Figure 6. Physics-guided hybrid dynamic modeling structure.

A differential platform with two independent drives (left/right) is considered. Following the standard differential-drive robot representation, the input control signals at the power electronics level, or equivalent voltage/PWM commands, are defined as [39]:

$$u(t) = \begin{bmatrix} u_L(t) \\ u_R(t) \end{bmatrix}. \quad (12)$$

where $u_L(t)$ and $u_R(t)$ are the left and right motor input commands, respectively. This expression is not introduced as a novel model component; it is used only to define the input notation for the subsequent electromechanical drive equations.

The main variables measured (or assessed by the observer):

$$\omega(t) = \begin{bmatrix} \omega_L(t) \\ \omega_R(t) \end{bmatrix}, i(t) = \begin{bmatrix} i_L(t) \\ i_R(t) \end{bmatrix}, v(t), \psi(t), \quad (13)$$

where $\psi(t)$ is the yaw angle of the platform. The superscripts L and R are consistently used to denote the left and right drive channels, respectively.

For numerical integration of the complete electromechanical and platform dynamics, the full state vector is defined as follows:

$$x(t) = [i_L(t), i_R(t), \omega_L(t), \omega_R(t), v(t), \psi(t)]^T \quad (14)$$

The vector $x(t)$ is the only full state vector used for numerical integration of the dynamic model. In contrast, Equation (1) defines the feedback signal vector $y_{fb}(t)$, which is supplied to the controller and to the neural network compensator. Therefore, Equations (1) and (14) no longer represent two different orderings of the same state vector.

Output vector of the simulation (for comparison with target trajectories and quality metrics):

$$y(t) = \begin{bmatrix} v(t) & \psi(t) & \omega_L(t) & \omega_R(t) \end{bmatrix}^T. \quad (15)$$

The electrical part of each motor is described by the standard model of the armature chain:

$$L \dot{i}_j(t) = u_j(t) - R i_j(t) - k_e \omega_j(t), j \in L, R, \quad (16)$$

If it is necessary to take into account the limitations of power electronics, control action saturation is introduced:

$$u_j(t) = \text{sat}_{[u_{\min}, u_{\max}]}(u_{\text{cmd},j}(t)). \quad (17)$$

This block generates currents $i_{L,R}$, which determine electromagnetic moments:

$$\tau_{m,j}(t) = k_t i_j(t), \quad (18)$$

where k_t —the motor torque coefficient.

The mechanical part of the drive takes into account the reduced inertia J_{eq} , viscous friction b , and the equivalent load torque $\tau_{load,j}(t)$, including contact resistance and external disturbances [40]:

$$J_{eq} \dot{\omega}_j(t) = \tau_{m,j}(t) - \tau_{load,j}(t) - b \omega_j(t). \quad (19)$$

To connect with the wheel part, the wheel radius r is introduced and, if there is a gearbox, the gear ratio G . In its simplest form (with ω_j as the wheel speed), the linear speed of the rim is:

$$v_{w,j}(t) = r \omega_j(t). \quad (20)$$

If ω_j refers to the engine shaft, then:

$$\omega_{w,j}(t) = \frac{\omega_j(t)}{G}, v_{w,j}(t) = r \omega_{w,j}(t). \quad (21)$$

An important element of the model is the formation of the longitudinal force $F_{x,j}(t)$ through the adhesion coefficient μ and the normal reaction $F_{z,j}(t)$. First, the longitudinal slip ratio of each wheel is determined:

$$s_j(t) = \frac{v_{w,j}(t) - v(t)}{\max(|v_{w,j}(t)|, |v(t)|) + \varepsilon}, \quad (22)$$

where $j \in L, R$, r is the wheel radius, $\omega_j(t)$ is the angular velocity of the corresponding wheel, $v(t)$ is the platform linear velocity, and $\varepsilon = 0.01$ m/s is a low-speed regularization constant introduced to avoid numerical singularity when $v(t)$ approaches zero.

Next, a linkage map/model $\mu = \mu(s)$ is introduced. In the context of reproducible modeling, the following can be used: (i) a saturable nonlinear function, (ii) a Burckhardt-type parametric approximation, (iii) a piecewise smooth approximation that preserves physical correctness. In general:

$$\mu_j(t) = \mu(s_j(t); \theta_\mu), \quad (23)$$

where θ_μ —Surface parameters (asphalt/soil/ice, etc.).

Then the traction force:

$$F_{x,j}(t) = \mu_j(t) F_{z,j}(t). \quad (24)$$

where $\mu_L(t)$ and $\mu_R(t)$ are the corresponding effective adhesion coefficients.

Figure 7 shows the computational flowchart of the contact model: computing $s(t)$ from $\omega(t)$ and $v(t)$, forming $\mu(s)$, estimating $\hat{\mu}(s)$ in the traction estimator, and generating $F_x(t)$ as input to the hull dynamics.

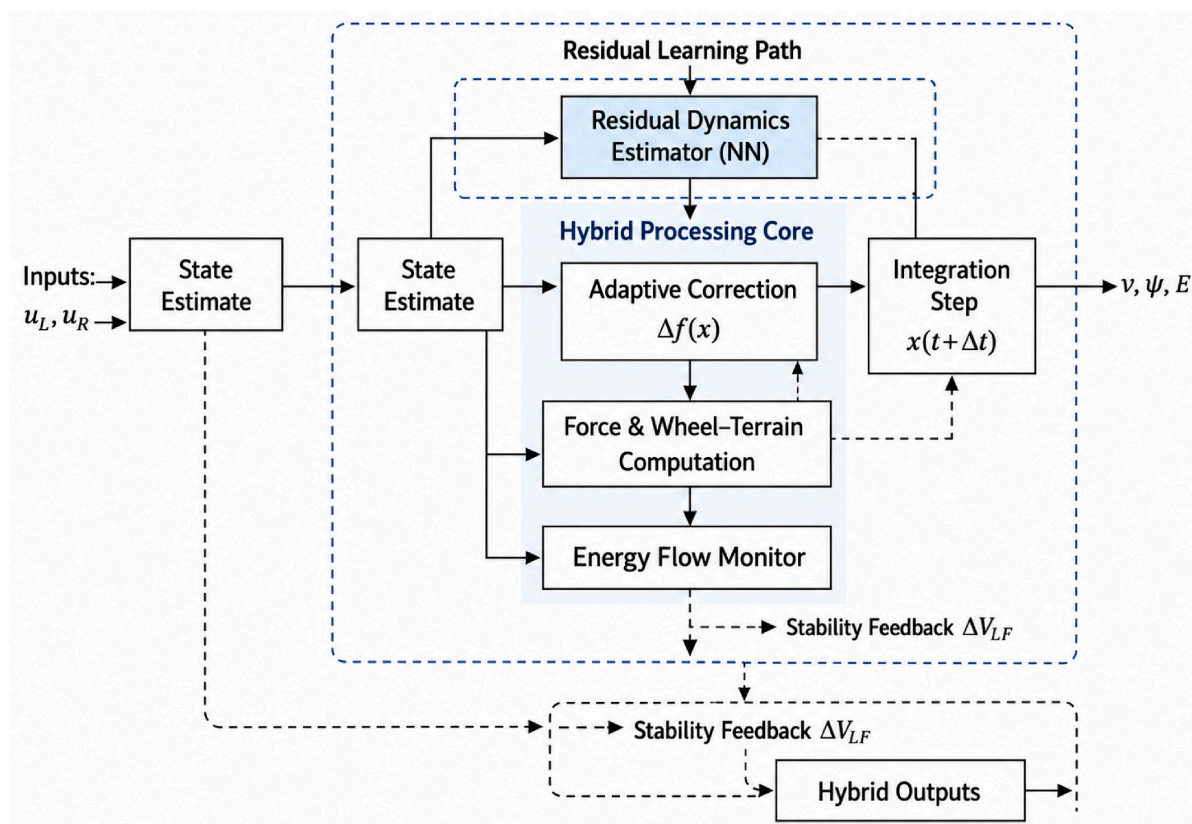


Figure 7. Contact interaction computational scheme.

The dynamics of translational motion (along the direction of motion) is written as:

$$m \dot{v}(t) = F_{x,L}(t) + F_{x,R}(t) - F_{res}(t), \quad (25)$$

where m is the robot's mass, and $F_{res}(t)$ is the total resistance to movement (aerodynamics, rolling, slope, etc.). In the basic formulation, the resistance can be written as:

$$F_{res}(t) = c_r v(t) + c_0 \operatorname{sgn}(v(t)), \quad (26)$$

The robot's rotation is determined by the differential platform kinematics. We introduce the base b (the distance between the wheels). Then the yaw rate is:

$$\dot{\psi}(t) = \frac{r}{b}(\omega_R(t) - \omega_L(t)), \quad (27)$$

Accordingly, the complete planar kinematic model used for trajectory-level simulation is written as:

$$\begin{aligned} \dot{x}(t) &= v(t) \cos \psi(t) \\ \dot{y}(t) &= v(t) \sin \psi(t) \\ \dot{\psi}(t) &= \omega(t) \end{aligned} \quad (28)$$

where $x(t)$ and $y(t)$ are the Cartesian coordinates of the platform center of mass, and $\omega(t)$ is the yaw rate defined in Equation (27). In the previous version, only the translational

part of this kinematic relation was shown after Equation (27). In the revised version, the yaw-angle equation is included explicitly to avoid ambiguity.

These equations are used to simulate scenarios involving trajectory tracking and lateral deviation.

The physical model is supplemented with a structured residual dynamics estimator. The following decomposition is introduced:

$$\dot{x}(t) = f_{phys}(x(t), u(t), \hat{\mu}(t)) + \Delta f(x(t)), \quad (29)$$

where $f_{phys}(\cdot)$ is defined by the equations of electrical, mechanical, contact, and body dynamics, and $\Delta f(x)$ is the residual term approximated by the neural network module (Residual Dynamics Estimator).

An important element is the injection point: the residual dynamics are not introduced into all equations at once, but rather into a selected subset corresponding to the most uncertain regions. In this formulation, the injection occurs in the mechanical and/or contact layer (for example, in the $\dot{\omega}$ equation and/or in the $\mu(s)$), which ensures interpretability and reduces the risk of incorrect physical effects:

$$\dot{\omega}_j(t) = \dot{\omega}_j^{phys}(t) + \Delta f_{\omega,j}(x), \mu_j(t) = \mu^{phys}(s_j(t)) + \Delta f_{\mu,j}(x). \quad (30)$$

This structuring allows for further correct analysis of stability and energy exchange and the explanation of improved results without resorting to a “black box”.

For reproducible verification of the physical correctness of the modeling, energy flow and stability monitoring are used. Electrical power at each drive:

$$P_{e,j}(t) = u_j(t) i_j(t), \quad (31)$$

Mechanical power on the shaft:

$$P_{m,j}(t) = \tau_{m,j}(t) \omega_j(t), \quad (32)$$

Power at the interaction level with the housing (in the longitudinal channel):

$$P_x(t) = (F_{x,L}(t) + F_{x,R}(t)) v(t). \quad (33)$$

Total kinetic energy (simplified) includes translational and rotational components:

$$E(t) = \frac{1}{2} m v^2(t) + \frac{1}{2} J_z \dot{\psi}^2(t) + \sum_{j \in \{L,R\}} \frac{1}{2} J_{eq} \omega_j^2(t), \quad (34)$$

where J_z is the moment of the platform's inertia relative to the vertical axis.

The difference in fluxes and losses is used in the subsequent section to verify the energy balance and interpret the effectiveness of the hybrid superstructure (e.g., reducing peak currents/power and smoothing transients). For stability analysis, Lyapunov-based monitoring, coordinated with training supervision blocks, is used in the control loop. At the modeling level, the same Lyapunov-type diagnostic function defined in Equation (8) is monitored and the sign of $\dot{V}(t)$ is tracked to assess the “safety” of the adaptive intervention under different driving conditions. This directly links the mathematical model, the hybrid superstructure, and the supervision mechanism previously demonstrated in the control methodology.

To eliminate ambiguities in reproducibility, the simulation computational cycle is fixed. At each discrete step $t_k \rightarrow t_{k+1} = t_k + \Delta t$, the following operations are performed:

1. Generate inputs $u_L(t_k)$ and $u_R(t_k)$ (from the control algorithm) and available state estimates $\tilde{x}(t_k)$;
2. Calculate electrical dynamics and update currents $i_{L,R}(t_k)$;
3. Calculate mechanical dynamics and update velocities $\omega_{L,R}(t_k)$;
4. Calculate slip $s_{L,R}(t_k)$ and contact parameters;
5. Estimation of $\hat{\mu}(t_k)$ and generation of forces $F_{x,L}, F_{x,R}$;
6. Updating the body dynamics $v(t_k), \psi(t_k)$;
7. Calculation of the residual dynamics $\Delta f(x)$ and its injection into the specified subsystems;
8. Monitoring of energy quantities and stability diagnostics;
9. Numerical integration of $x(t_{k+1})$ using the fourth-order Runge–Kutta method or an equivalent circuit.

Figure 8 illustrates the general structure of the proposed hybrid control system for the mobile robot drives. The diagram highlights a physically interpretable control loop based on a mathematical model of an electromechanical drive, an adaptive neural network compensator for residual dynamics, and a stability supervision module implemented using a Lyapunov function. The basic controller generates a nominal control response taking into account the drive dynamics and the current system state, while the neural network unit generates a correction signal to compensate for unmodeled nonlinearities, parameter variations, and external disturbances. A supervision mechanism limits the adaptive action of the neural network and maintains closed-loop stability. The presented architecture reflects the principle of integrating a physical model and trainable algorithms while maintaining the interpretability and robustness of the control system.

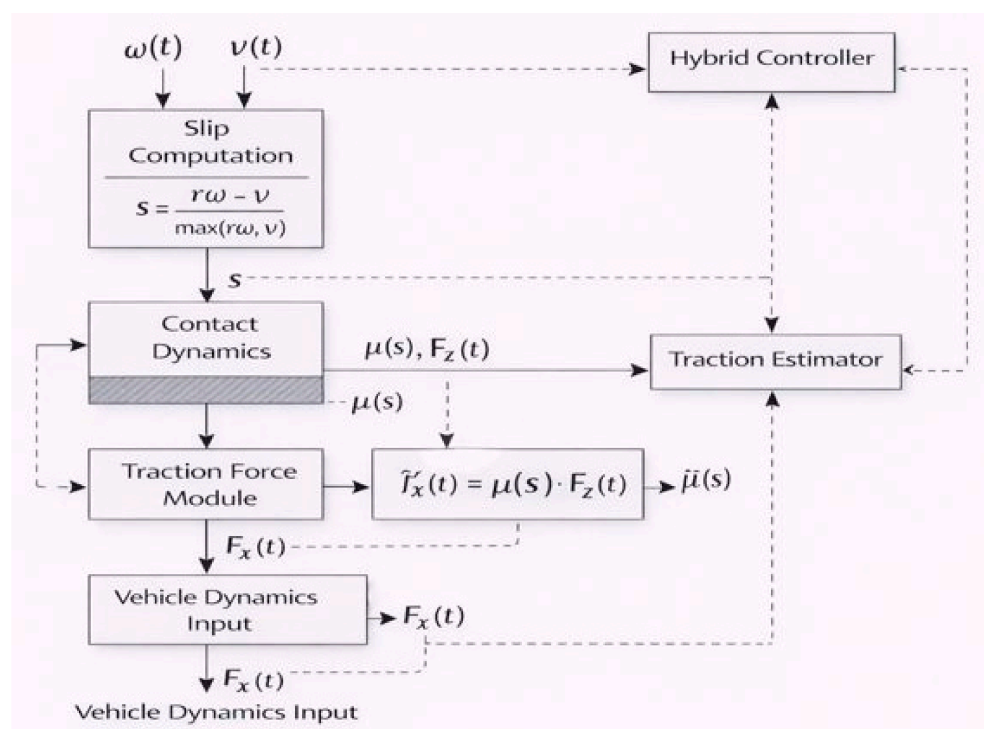


Figure 8. Computational pipeline for hybrid model.

The baseline parameter set is based on the values specified for a mobile platform with a PMDC motor (supply voltage, armature resistance/inductance, torque and EMF constants, equivalent inertia and viscous friction, wheel radius, and track). These values are used as a starting point, and their variations are then specified as scenario-specific (Table 2) [41].

Table 2. Nominal model parameters.

Parameter	Designation	Value	Units of Measurement	Description
Drive supply voltage	V_{in}	12	V	Electric drive supply voltage
Electromagnetic torque constant	K_t	1.188	N·m/A	Motor torque coefficient
Back-EMF constant	K_b	1.185	V·s/rad	Back EMF constant
Armature resistance	R_a	0.156	Ohm	Armature winding resistance
Armature inductance	L_a	0.82	mH	Armature circuit inductance (mH scale used for physical consistency)
Equivalent moment of inertia	J_{eq}	0.275	kg·m ²	Drive reduced moment of inertia
Equivalent viscous friction	b_{eq}	0.392	N·m·s/rad	Viscous friction coefficient
Wheel radius	r	0.075	m	Radius of the drive wheel of the mobile platform
Wheel track	$2b$	0.40	m	Distance between the centers of the drive wheels

For each side $j \in \{L, R\}$, the coupled electrical and mechanical dynamics of the PMDC drive are used:

$$\begin{aligned}\dot{i}_j(t) &= -\frac{R_a}{L_a} i_j(t) - \frac{K_b}{L_a} \omega_j(t) + \frac{1}{L_a} u_j(t), \\ \dot{\omega}_j(t) &= \frac{K_t}{J_{eq}} i_j(t) - \frac{b_{eq}}{J_{eq}} \omega_j(t) - \frac{1}{J_{eq}} \tau_{load,j}(t),\end{aligned}\quad (35)$$

where $u_j(t)$ —control voltage (or equivalent driver command signal), and $\tau_{load,j}(t)$ —the total load torque from the wheel–road contact and the resistances.

Substituting the nominal values from Table 2 yields a numerical form of the mathematical model:

$$\begin{aligned}\dot{i}_j(t) &= -189.88 i_j(t) - 1445.12 \omega_j(t) + 1219.51 u_j(t), \\ \dot{\omega}_j(t) &= 4.3207 i_j(t) - 1.4255 \omega_j(t) - 3.6364 \tau_{load,j}(t).\end{aligned}\quad (36)$$

Equivalently, as a matrix state model for each channel j :

$$\begin{aligned}x_j(t) &= \begin{bmatrix} \omega_j(t) \\ i_j(t) \end{bmatrix}, \dot{x}_j = Ax_j + Bu_j + G\tau_{load,j}, \\ A &= \begin{bmatrix} -1.4255 & 4.3207 \\ -1445.12 & -189.88 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1219.51 \end{bmatrix}, G = \begin{bmatrix} -3.6364 \\ 0 \end{bmatrix}.\end{aligned}\quad (37)$$

The linear velocity $v(t)$ and the angular yaw rate $\dot{\psi}(t)$ are expressed in terms of the angular velocities of the wheels:

$$v(t) = \frac{r}{2}(\omega_R(t) + \omega_L(t)), \dot{\psi}(t) = \frac{r}{2b}(\omega_R(t) - \omega_L(t)).\quad (38)$$

For $r = 0.075$ m and $b = 0.20$ m can be obtained:

$$v(t) = 0.0375(\omega_R + \omega_L), \dot{\psi}(t) = 0.1875(\omega_R - \omega_L).\quad (39)$$

The longitudinal slip calculation is introduced (in accordance with Figure 7):

$$s_j(t) = \frac{r\omega_j(t) - v(t)}{\max(r|\omega_j(t)|, |v(t)|, \varepsilon)}, j \in \{L, R\}\quad (40)$$

where r is the wheel radius, $\omega_j(t)$ is the time-dependent angular velocity of the corresponding wheel, $v(t)$ is the platform linear velocity, and ε is a low-speed regularization constant.

In Equation (40), $\omega_j(t)$ denotes the instantaneous angular velocity of the j -th wheel. The omission of the time argument in the previous notation was only a shorthand and has been corrected to avoid ambiguity. Therefore, $\omega_j(t)$ in Equation (40) is consistent with $\omega_R(t)$ and $\omega_L(t)$ used in Equation (39).

The adhesion coefficient $\mu(s)$ is given by the Burckhardt model:

$$\mu(s) = c_1(1 - e^{-c_2 s}) - c_3 s, \quad (41)$$

where c_1 , c_2 , and c_3 are empirical Burckhardt tire–road interaction coefficients that define the shape of the adhesion–slip curve $\mu(s)$. The coefficient c_1 determines the peak level of the adhesion curve, c_2 controls the rate at which the adhesion coefficient increases with slip, and c_3 characterizes the decrease in adhesion at higher slip values. These coefficients are selected according to the surface type in order to reproduce different wheel–ground contact conditions in the simulation scenarios. In this study, four representative surface conditions were used: dry asphalt, wet asphalt, snow, and ice.

To make the selection of the Burckhardt model parameters reproducible, the coefficient sets used for different surface conditions are summarized in Table 3. These coefficients were not treated as arbitrary fitting parameters; instead, each set was assigned to a specific representative wheel–ground contact condition and was used consistently in the corresponding simulation scenarios.

Table 3. Burckhardt model coefficients used for different surface conditions [42].

Surface Condition	c_1	c_2	c_3	Physical Interpretation
Dry asphalt	1.2801	23.99	0.52	High-adhesion surface
Wet asphalt	0.857	33.822	0.347	Reduced-adhesion paved surface
Snow	0.1946	94.129	0.0646	Low-adhesion deformable surface
Ice	0.05	306.39	0.0	Very-low-adhesion surface

The same coefficient sets were used consistently in all simulation scenarios involving variable adhesion and split- μ disturbances. This makes the surface-dependent contact model reproducible and avoids treating c_1 , c_2 , and c_3 as arbitrary fitting parameters.

The longitudinal contact force:

$$F_{x,j}(t) = \mu(s_j(t)) F_{z,j}(t), \quad (42)$$

and the wheel/shaft load torque:

$$\tau_{load,j}(t) = r F_{x,j}(t) + \tau_{rr,j}(t), \quad (43)$$

where $\tau_{rr,j}$ is the rolling resistance/loss torque.

Figure 9 shows the time dependencies of the input electrical power $P_e = u(t)i(t)$, the total mechanical power taking into account losses $\dot{E} + P_{cu} + P_{visc} + P_{load}$, and the residual balance error.

The obtained results demonstrate a high degree of consistency in energy flows: the balance error does not exceed a few percent in transient conditions and tends to zero in steady state. This confirms the correctness of the generated electromechanical model and its suitability for subsequent numerical analysis and control.

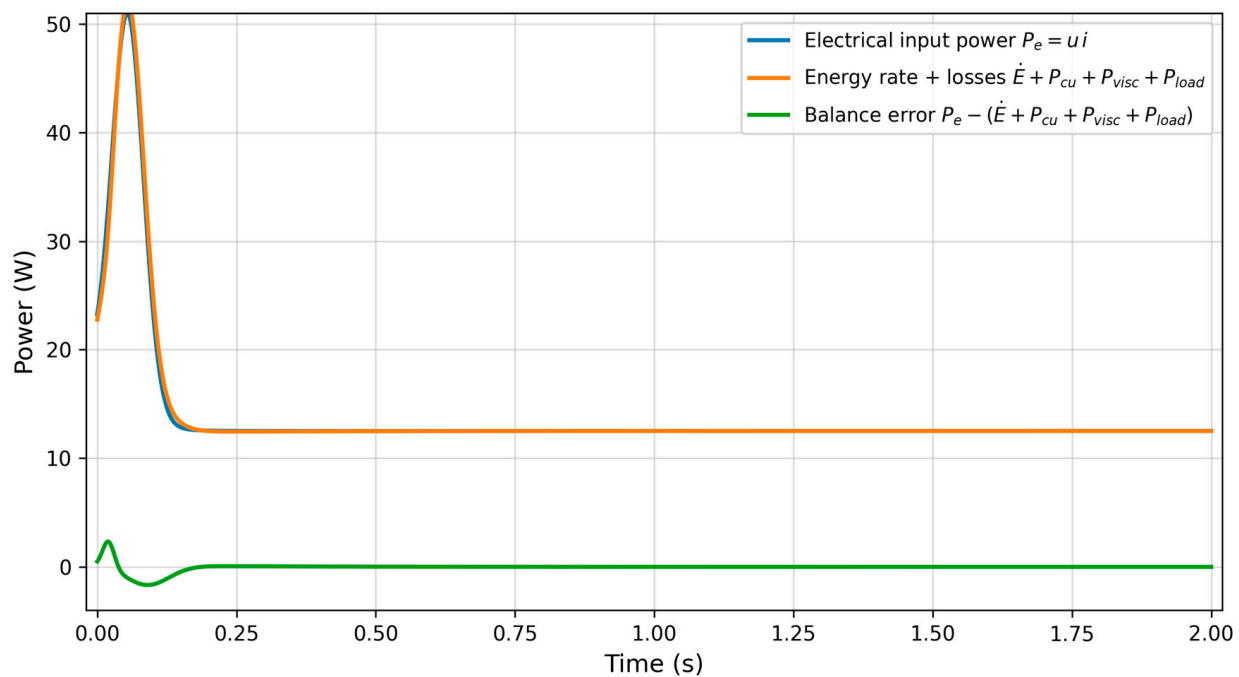


Figure 9. Power balance validation of the drive model.

Thus, a mathematical model of the electromechanical drive of a differential mobile platform has been developed, taking into account electrical and mechanical dynamics, wheel-to-surface contact interaction, and the dynamics of robot motion. The model is supplemented with a hybrid component in the form of a neural network residual dynamics estimator with a specific injection point into the mechanical subsystem. The resulting model ensures physical consistency in the description of motion processes and can be used for numerical analysis of the effectiveness of hybrid control algorithms for mobile robots.

5. Results and Discussion

This section presents the results of numerical simulations aimed at quantitatively assessing the effectiveness of the proposed hybrid control architecture. The main objectives of the experiments were to verify the contribution of the neural network adaptive compensator, analyze the stability of the closed-loop control system in various driving modes, and confirm the physical consistency of the developed mathematical model of the electromechanical drive and the wheel-surface contact interaction. All computational experiments were performed in Python version 3.10 using the NumPy 1.24, SciPy 1.10, and PyTorch 2.0 libraries [43,44]. Numerical integration of the dynamic equations was performed using the fourth-order Runge–Kutta method with a fixed step size of 1 ms. To ensure reproducibility, all simulation scenarios were run with identical discretization settings, initial conditions, and simulation parameters. To quantitatively evaluate the contribution of the neural network adaptive block, an ablation study was conducted comparing three control options: a basic physically based controller, a hybrid architecture with fixed neural network weights, and the proposed hybrid system with online parameter adaptation.

Figure 10a shows the transient response of the reference speed tracking. The basic controller exhibits the largest overshoot amplitude and a longer settling time. Using a hybrid architecture with fixed neural network weights partially improves the dynamic performance, but the best results are achieved with online adaptation enabled. The tracking error analysis presented in Figure 10b confirms this observation. The proposed adaptive hybrid algorithm reduces the maximum error by more than a factor of two compared to the physically based controller and also significantly reduces the oscillatory component of the

error in transient conditions. This demonstrates the ability of the neural network corrector to effectively compensate for unmodeled dynamic effects and parametric uncertainties of the system.

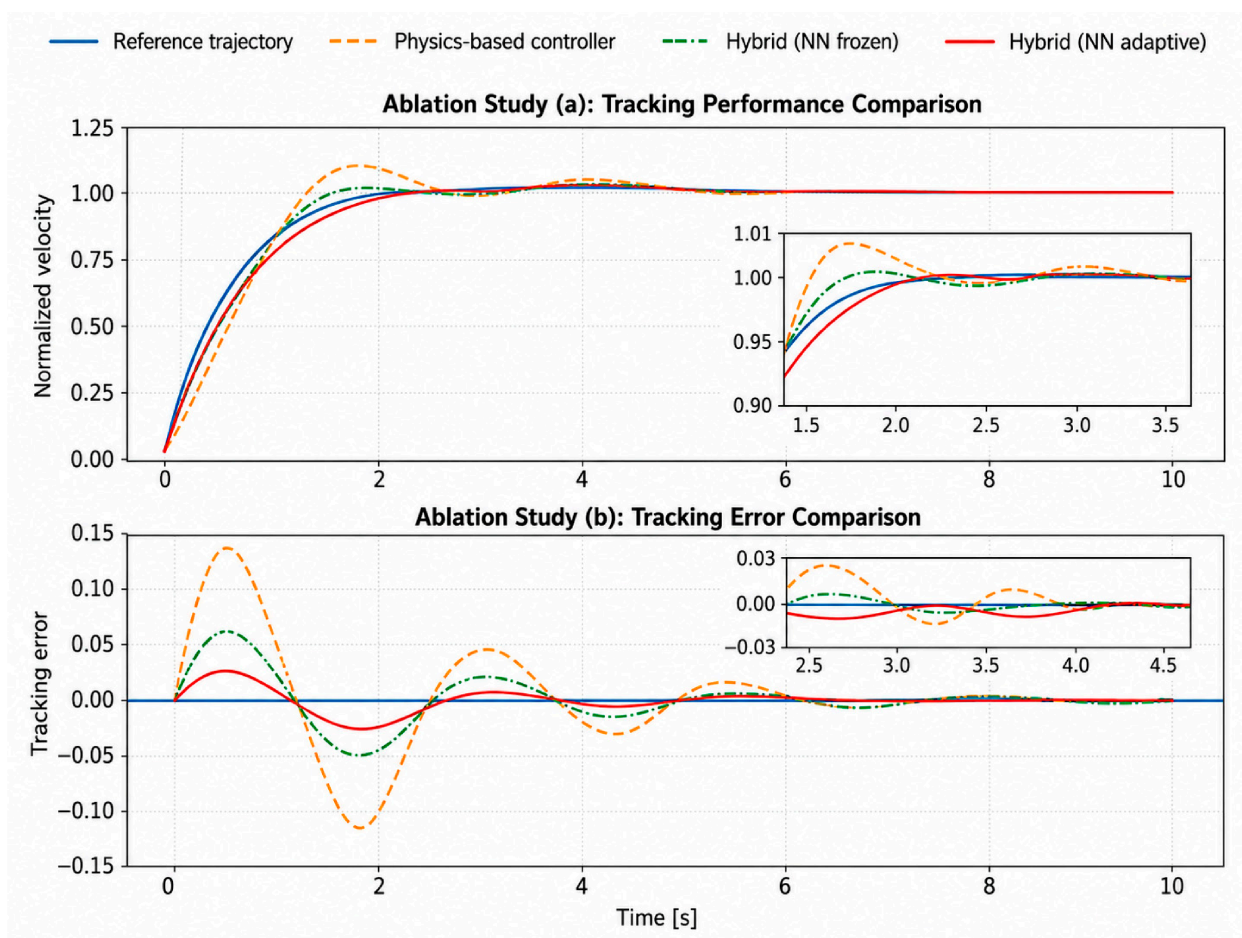


Figure 10. Ablation study of hybrid controller performance with magnified inset views: (a) Normalized velocity tracking response; (b) tracking error comparison. The inset windows highlight the settling region and the residual-error region.

Using the fixed-weight neural network model allows us to separate the effects of the architecture itself from those of online learning. The results demonstrate that the main improvement in dynamic performance is achieved through adaptive tuning of the neural network block's parameters, rather than solely through static model expansion. This confirms the feasibility of using online adaptation in mobile robot control tasks under uncertainty. Figure 11 illustrates the complete computational pipeline, including input scenario generation, uncertainty injection, robot dynamics simulation, and processing of output control performance metrics. The Monte-Carlo Simulation Input block generates sequences of reference trajectories and motion scenario parameters, including acceleration modes, split- μ conditions, and load surges.

The Disturbance and Uncertainty Injection subsystem implements the model's parametric uncertainty (Δm , the mass perturbation, and ΔI_Z , the perturbation of the yaw moment of inertia), variations in adhesion coefficients (μ_L and μ_R for the left and right wheels), and simulates sensor noise and delays (η_y and η_ψ denote additive measurement noise in the lateral-position and yaw-angle channels, respectively, while τ_d denotes the measurement or communication delay). This allows for the evaluation of system behavior under conditions similar to those encountered in real-life mobile robot operation. The hy-

brid control system includes a mode selector (FSM/Mode Selector), which switches control configurations depending on the current motion scenario, as well as a self-tuning controller combining a physically based LQR component and a neural network adaptive module.

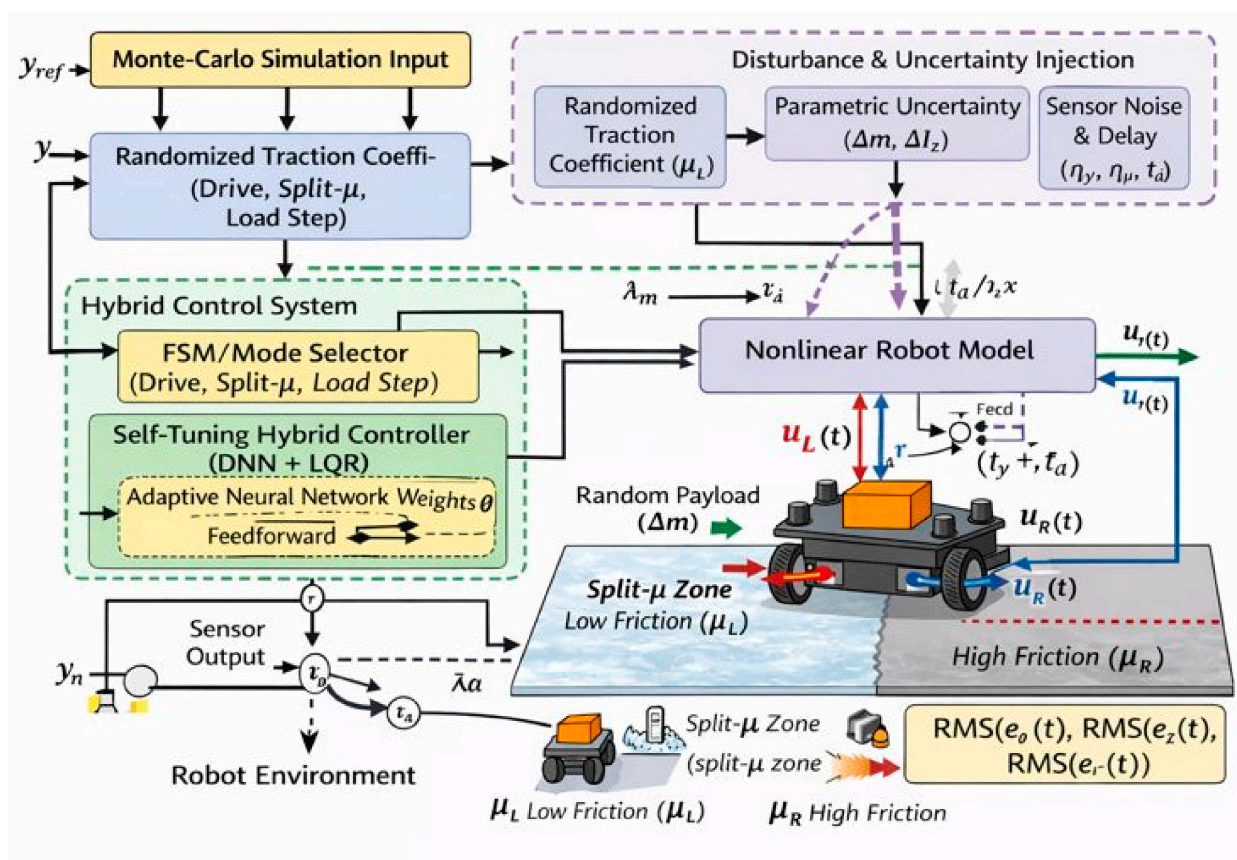


Figure 11. Statistical robustness assessment framework of the proposed hybrid neural control system under disturbed operating conditions.

The neural network corrector generates an online adaptation of the θ weights, providing compensation for residual dynamics and parametric uncertainties, while a feedforward channel is used to accelerate the system's response in transient conditions. The outputs of the nonlinear model of the mobile robot include control actions on the left and right actuators, $u_L(t)$ and $u_R(t)$, as well as measured sensor signals containing additive noise and delays. This data is used to calculate integral control quality metrics, including $\text{RMS}(e_\theta(t))$, $\text{RMS}(e_\psi(t))$, and $\text{RMS}(e_y(t))$.

Thus, Figure 11 demonstrates that the effectiveness of the proposed algorithm is assessed not in a single deterministic scenario but within the framework of a statistically representative set of disturbed conditions, which significantly increases the reliability of the results obtained and confirms the robustness of the developed control system.

A comprehensive verification of the drive interaction with the platform dynamics and the wheel–road contact model was performed. Figure 12 presents the results of the combined modeling of the electric drive, motion kinematics, and contact mechanics. The top panel illustrates the relationship between the control voltage and the electromagnetic torque of the motor; the middle panel demonstrates the consistency of the platform linear velocity $v(t)$ and the wheel peripheral velocity $r_\omega(t)$, n_d ; and the bottom panel reflects the evolution of the slip coefficient $s(t)$, the adhesion coefficient $\mu(t)$ and the longitudinal traction force $F_x(t)$.

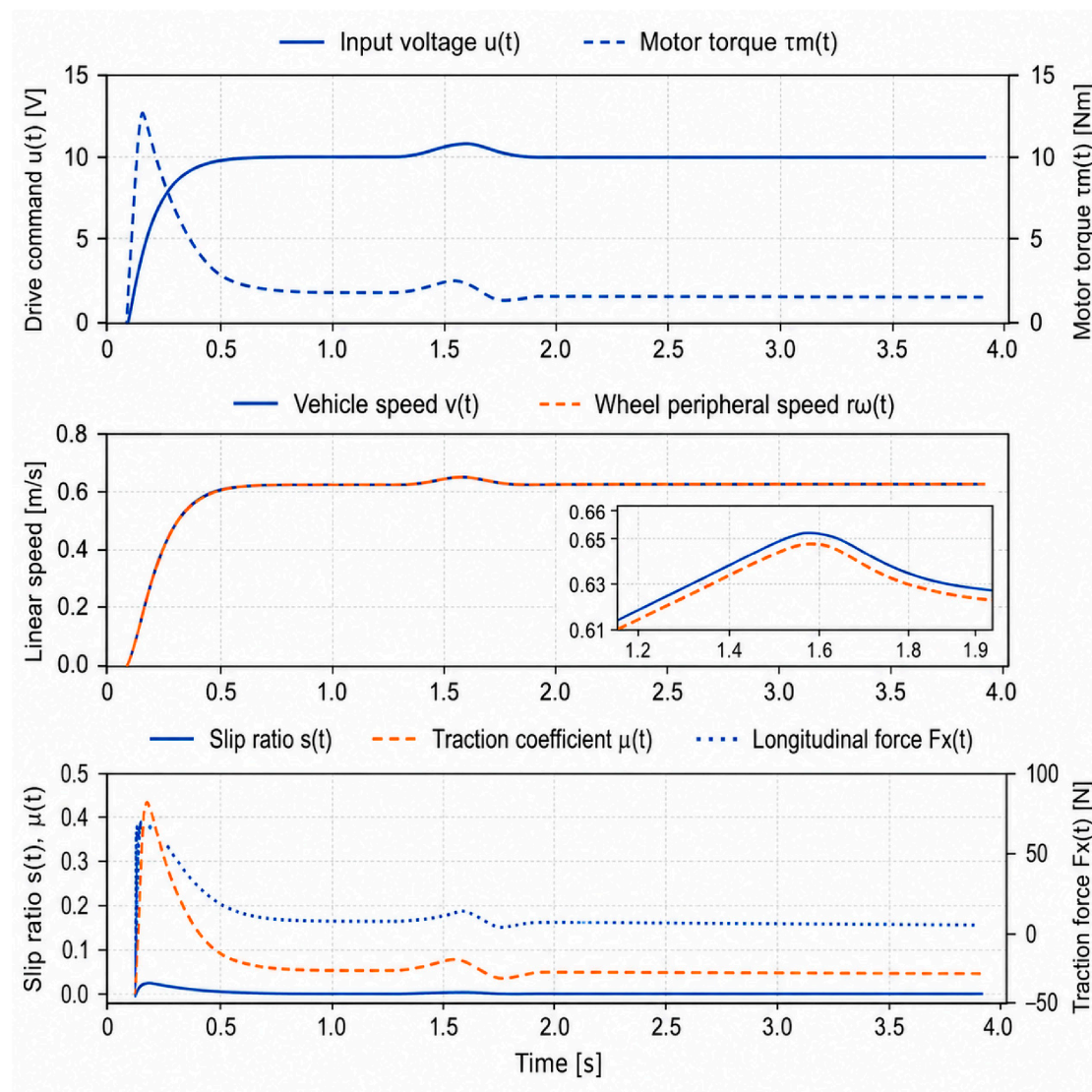
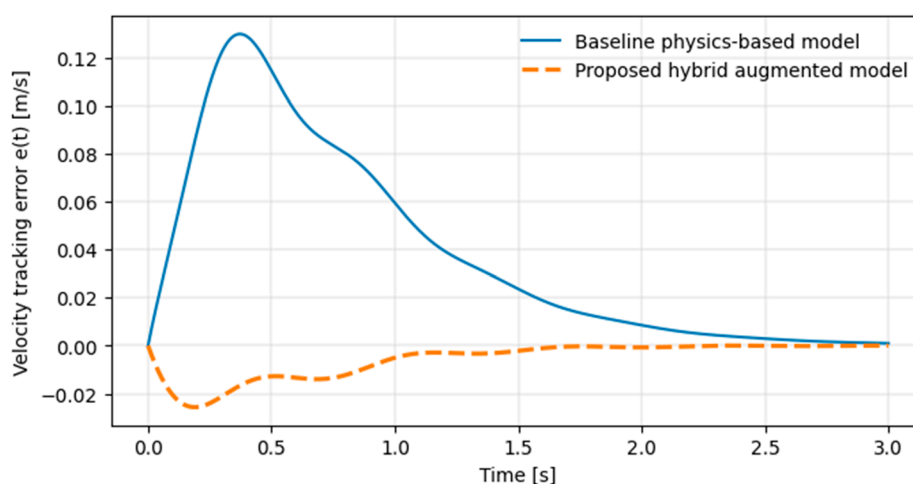


Figure 12. Coupled drive–kinematics–wheel–ground interaction validation with a magnified inset view.

The obtained dependencies confirm the physical consistency of the implemented model: kinematic velocities remain consistent, the contact force is correctly generated based on the nonlinear $\mu(s)$ dependence, and the traction force adequately responds to changes in the driving mode. Thus, the proposed structure of the mathematical model ensures the correct reproduction of traction processes and the dynamics of the mobile platform. Importantly, the consistency of the platform's linear velocity and the wheel's peripheral velocity in the middle graph of Figure 12 indicates the correct formation of the differential platform's kinematic connections, while the dynamics of the adhesion coefficient and longitudinal force in the lower graph confirm the physical feasibility of the used contact model. This allows the developed model to be used as a reliable basis for analyzing driving modes with variable contact conditions. To quantitatively assess the effect of the hybrid approximation of residual dynamics, a comparative test of the basic physical model and the extended hybrid model with neural network compensation was conducted. Figure 13 shows the dynamics of the speed tracking error for both approaches.



RMS error: baseline = 0.0561 m/s, hybrid = 0.0093 m/s | Peak error: baseline = 0.130 m/s, hybrid = 0.026 m/s

Figure 13. Hybrid augmentation effect on tracking error dynamics.

The results demonstrate a significant improvement in dynamic performance using the proposed hybrid structure. Specifically, a significant reduction in both the root mean square error (RMS) and peak deviations is observed compared to the classical physical model. This confirms the effectiveness of integrating the trainable compensating block into the mathematical model structure and justifies its use for improving the accuracy and stability of the control system.

It should also be emphasized that the reduction in RMS error and the peak deviations are directly related to the correct injection of residual dynamics into the most uncertain subsystems of the model. This confirms that the proposed hybrid augmentation structure not only improves tracking accuracy but also preserves the physical interpretability of the model, a fundamental advantage over purely neural network approaches.

The values presented in Table 4 confirm that the proposed hybrid architecture provides a more than sixfold reduction in RMS error and a more than fivefold reduction in peak deviations compared to the basic physically based controller.

Table 4. Comparative management quality metrics for different architectures.

Controller	RMS Speed Error (m/s)	Peak Error (m/s)	Settling Time (s)
Physical Controller	0.056	0.130	2.1
Hybrid (Frozen NN)	0.031	0.074	1.4
Proposed Hybrid (Adaptive NN)	0.009	0.026	0.9

Figure 14 shows a comparison of the linear velocity tracking dynamics for the basic physically based controller and the proposed hybrid control architecture.

As Figure 14 shows, the basic controller exhibits a slower transient response and increased settling time, which is due to the limited ability of the physical model to compensate for unmodeled dynamic effects and drive nonlinearities. At the same time, the proposed hybrid controller provides a faster speed reference and reduced transient deviations. It should also be noted that the speed trajectory using the hybrid controller practically coincides with the reference curve already at an early stage of the transient response, indicating the effective operation of the neural network compensator, which generates a corrective control action online. The reduction in dynamic tracking error is particularly pronounced in the acceleration range (0–1 s), where the influence of parametric uncertainties and inertial

effects is most significant. This confirms the ability of the proposed architecture to improve system performance without degrading stability.

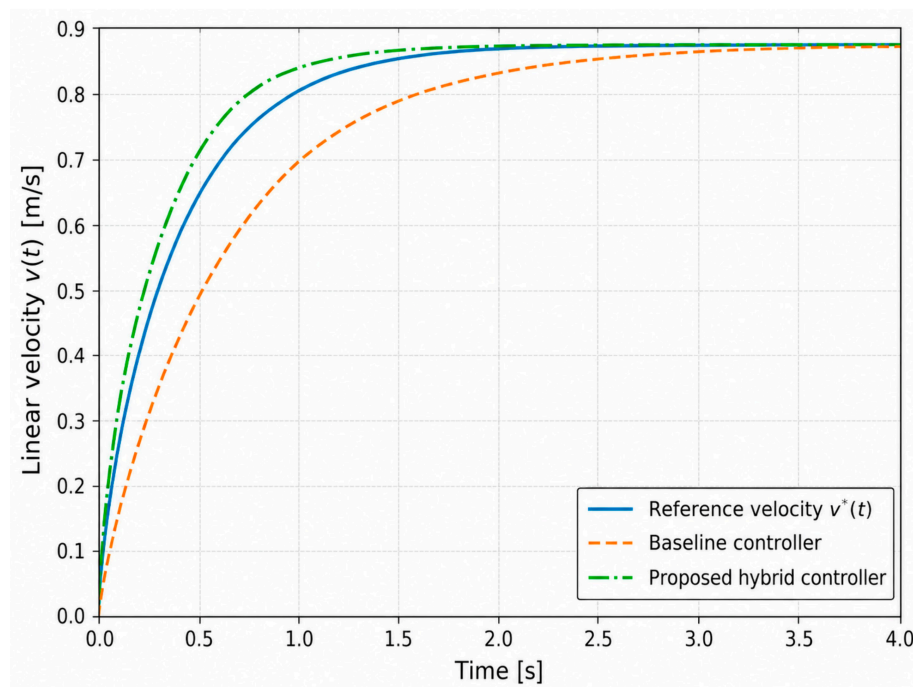


Figure 14. Linear velocity tracking dynamics under baseline and proposed hybrid control architectures.

Figure 15 shows a scenario of driving along an S-shaped trajectory under conditions of severe asymmetry of wheel grip (split- μ mode). The bottom graph illustrates the time evolution of the lateral tracking error. The vertical dotted line indicates the onset of the split- μ disturbance. After this point, the basic controller exhibits a sharp increase in error, while the hybrid architecture provides a smoother response and faster disturbance rejection. The vertical dotted line indicates the time instant at which the split- μ disturbance is introduced, corresponding to the onset point marked in the upper panel.

Quantitative analysis confirms the advantage of the proposed approach: the root mean square error after the split- μ condition decreases from 0.073 m to 0.052 m, and the peak deviation decreases from 0.176 m to 0.136 m. This corresponds to a 29% reduction in the RMS error and a 23% decrease in the maximum deviation. The obtained results demonstrate the ability of the hybrid algorithm to effectively compensate for asymmetric contact disturbances, which is critical for the operation of mobile robots on non-uniform surfaces.

Figure 16 presents the results of the tracking quality analysis for the reference linear velocity and angular velocity (yaw rate) in the nominal motion mode. The top graph shows the platform's linear velocity dynamics, where the hybrid controller achieves faster convergence to the reference value and less overshoot compared to the base controller. The middle graph illustrates angular velocity tracking, where the hybrid system also achieves more accurate alignment with the reference trajectory. The bottom graph demonstrates the tracking error dynamics.

Thus, the use of the hybrid architecture results in a reduction in the root mean square error of linear velocity from 0.0615 to 0.0464 m/s and the angular velocity error from 0.0529 to 0.0420 rad/s. The peak error of linear velocity decreases from 0.2005 to 0.1535 m/s, and that of angular velocity from 0.0851 to 0.0660 rad/s. These results confirm the improved tracking accuracy and dynamic performance of the system when using the adaptive neural network compensator.

Figure 17 shows the transient processes of the platform's linear velocity, actuator control action, and tracking error using the basic physically based controller and the proposed hybrid control architecture. The top graph shows the linear velocity dynamics with a step change in the reference. It is evident that the hybrid controller provides a smoother transient response and reduced overshoot compared to the basic controller, despite the presence of control action limitations. The middle graph illustrates the drive control action taking into account actuator saturation. It is observed that the hybrid algorithm generates smoother control signals and reduces the duration of drive operation in saturation mode, which reduces the dynamic load on the power electronics. The bottom graph shows the linear velocity tracking error. Using the hybrid architecture leads to faster error decay after transients and reduced residual deviations. The obtained results confirm that the integration of a neural network compensator for residual dynamics improves tracking quality and drive stability in the presence of control action limitations.

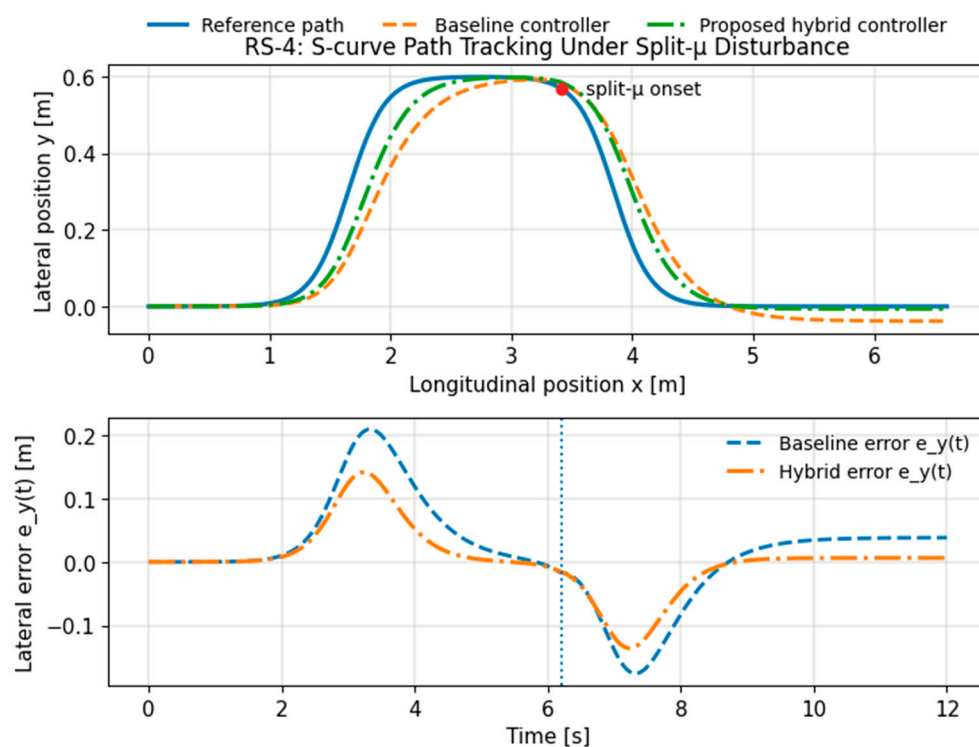


Figure 15. Mobile platform motion along an S-shaped trajectory under asymmetric wheel adhesion (split- μ mode).

The results presented in Figure 16 demonstrate that the use of a hybrid control architecture with a neural network compensator for residual dynamics improves the system's transient response compared to the baseline physically based controller. Despite the control constraints, the hybrid controller exhibits smoother control signal generation and a shorter drive saturation time. This leads to reduced dynamic overloads of the actuators and increased system stability. Tracking error analysis reveals faster attenuation after transient processes and reduced residual deviations, confirming the effectiveness of integrating adaptive neural network compensation into the physically based control structure of a mobile robot's drives. Figure 18 presents the simulation results for the mobile platform's motion under asymmetric wheel-to-surface adhesion conditions (split- μ mode), indicated by the shaded area. The top graph shows the dynamics of the platform's angular velocity relative to the reference trajectory. Under conditions of deteriorating adhesion, the base controller's deviation from the target trajectory increases, while the hybrid system demonstrates more stable adherence to the reference signal.

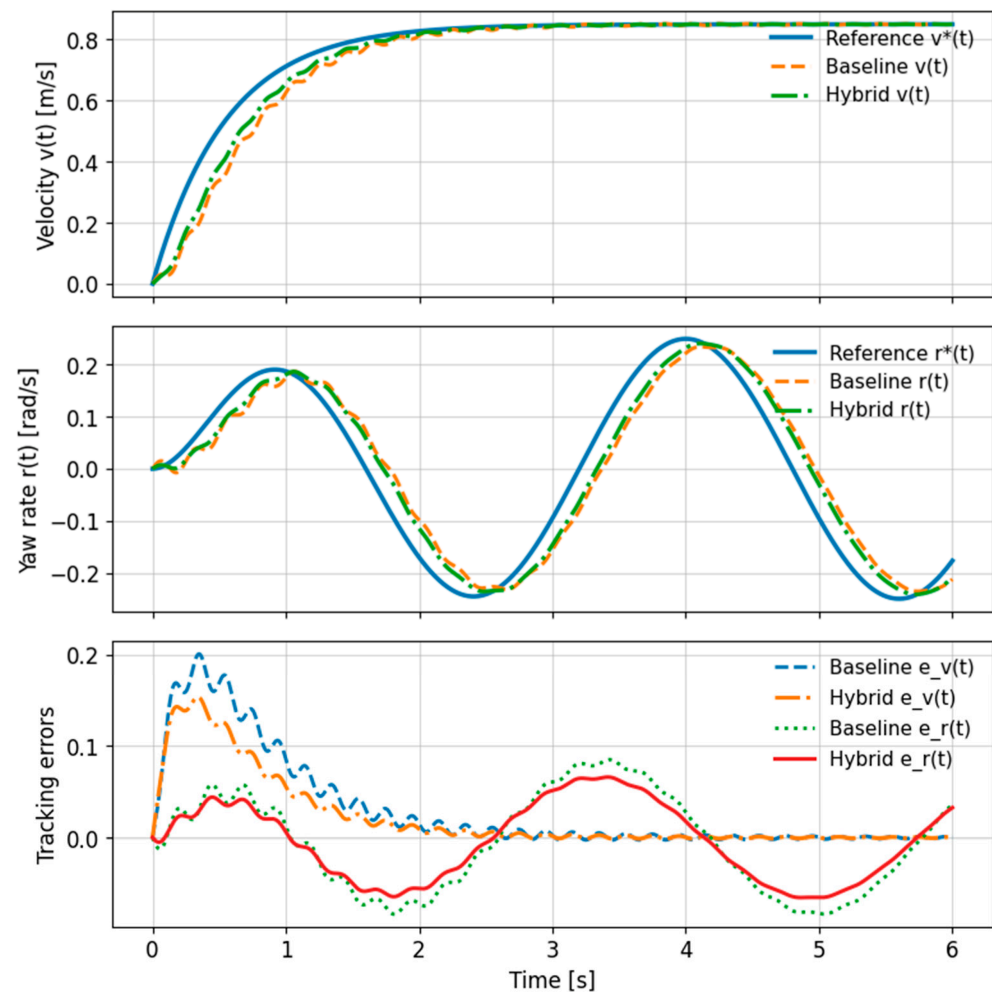


Figure 16. Nominal tracking characteristics for linear and angular velocity of a mobile robot, comparing the basic and hybrid controllers.

The middle graph illustrates the angular velocity error magnitude. In the zone of reduced adhesion, the base controller's error increases significantly, while the use of neural network compensation reduces the amplitude of these deviations and stabilizes the rotational dynamics. The bottom graph shows the dynamics of the platform's directional error.

The hybrid architecture ensures more stable system behavior under changing contact conditions and reduces the accumulation of directional error, confirming improved control robustness under adhesion uncertainty.

Figure 19 shows the time dependences of the control actions of the left and right wheel drives of a mobile platform under asymmetric adhesion conditions (split- μ mode), taking into account the limitations of the maximum control signal. The shaded area corresponds to the range of degraded surface adhesion.

Analysis of the control signals shows that under asymmetric adhesion, the hybrid control system generates somewhat more pronounced control signals compared to the base controller. This is due to additional compensation for unmodeled dynamic effects and contact condition variations, implemented by the neural network residual dynamics compensator. Despite the increase in control signal amplitude, the signals remain within the permissible saturation limits, and the improved tracking performance and motion stability confirm the effectiveness of the proposed control architecture.

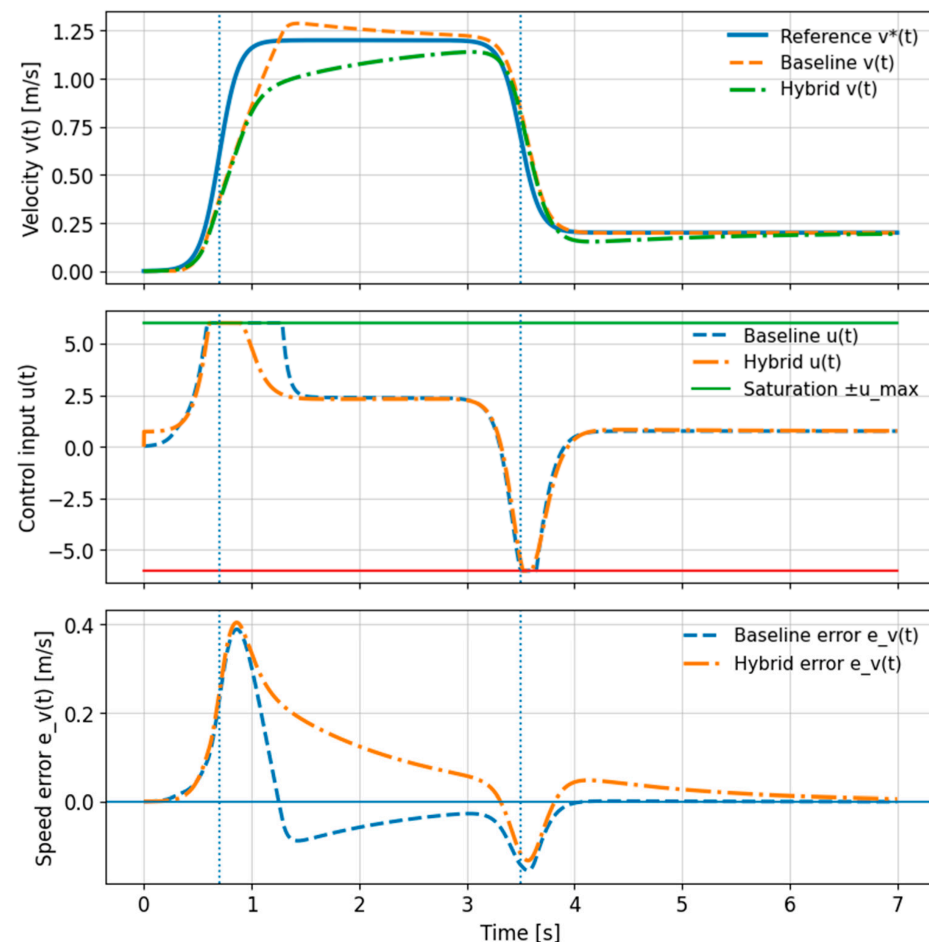


Figure 17. Transient processes of linear velocity, control action, and tracking error, comparing the baseline and hybrid controllers, taking into account actuator constraints.

Figure 20 shows the dynamics of the lateral tracking error of the mobile platform under asymmetric adhesion, corresponding to the split- μ mode (shaded area). During the initial period of motion, both control algorithms provide similar levels of tracking accuracy. However, as adhesion conditions deteriorate, a significant increase in lateral error is observed for the baseline controller, while the hybrid architecture demonstrates more stable system behavior and a significantly smaller increase in trajectory deviation. After exiting the low-adhesion zone, the hybrid controller stabilizes the error without significant drift accumulation, while the baseline controller exhibits a pronounced residual bias. The obtained results demonstrate improved trajectory control robustness when using a hybrid system with neural network compensation for residual dynamics.

Analysis of the lateral tracking error shows that the hybrid control architecture significantly reduces the mobile platform's sensitivity to asymmetric adhesion conditions and prevents the accumulation of trajectory deviation. This confirms the effectiveness of integrating neural network compensation to improve the robustness of mobile robotic platforms under uncertain contact conditions.

To summarize the scenario-based evaluation, the main simulation cases, their purpose, the observed control effect, and their practical interpretation are consolidated in Table 5. This table is intended to connect the numerical results with potential real-world operating conditions of differential-drive mobile robots. It also clarifies that the proposed hybrid architecture was not evaluated only in a single nominal case, but under several representative conditions, including model uncertainty, variable adhesion, asymmetric traction, actuator constraints, and adaptive neural compensation.

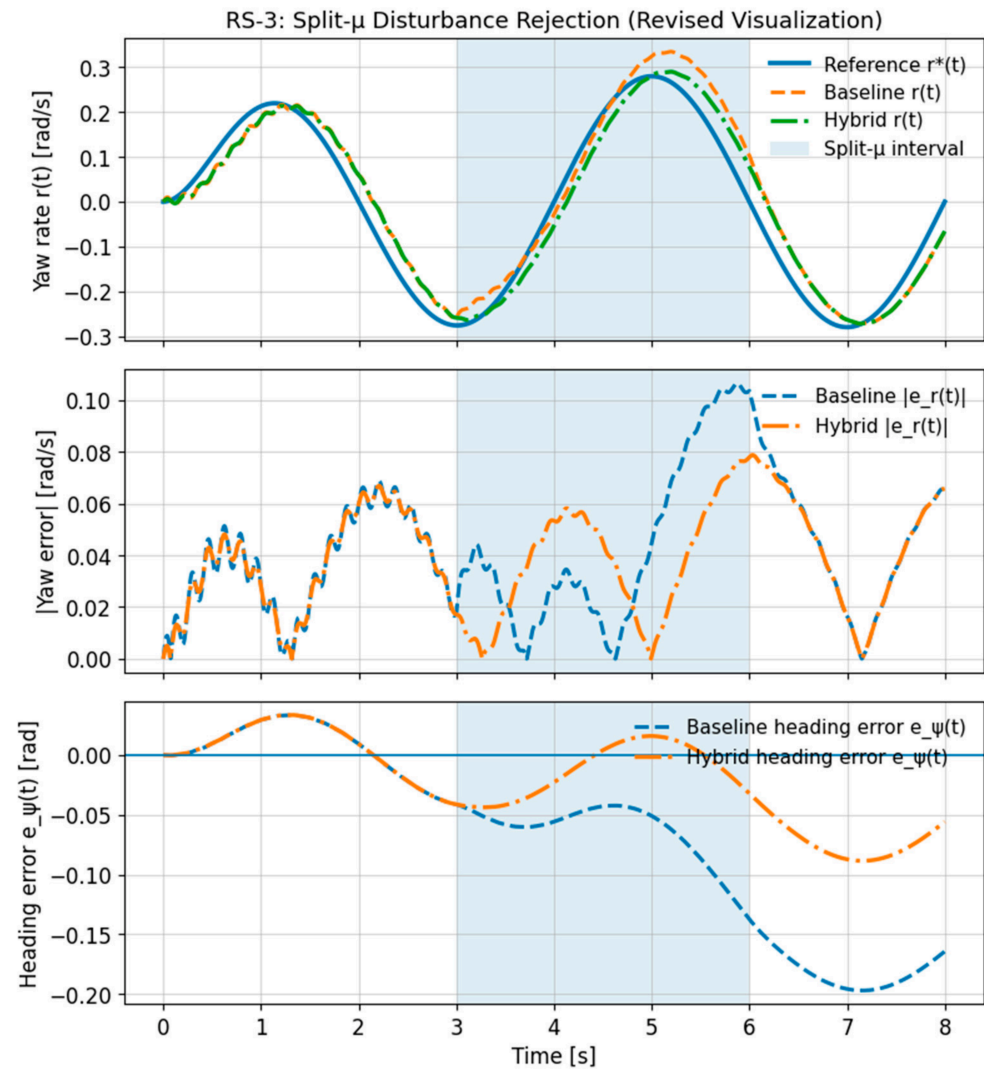


Figure 18. Tracking the angular velocity and directional error dynamics of a mobile platform under asymmetric adhesion conditions (split- μ mode).

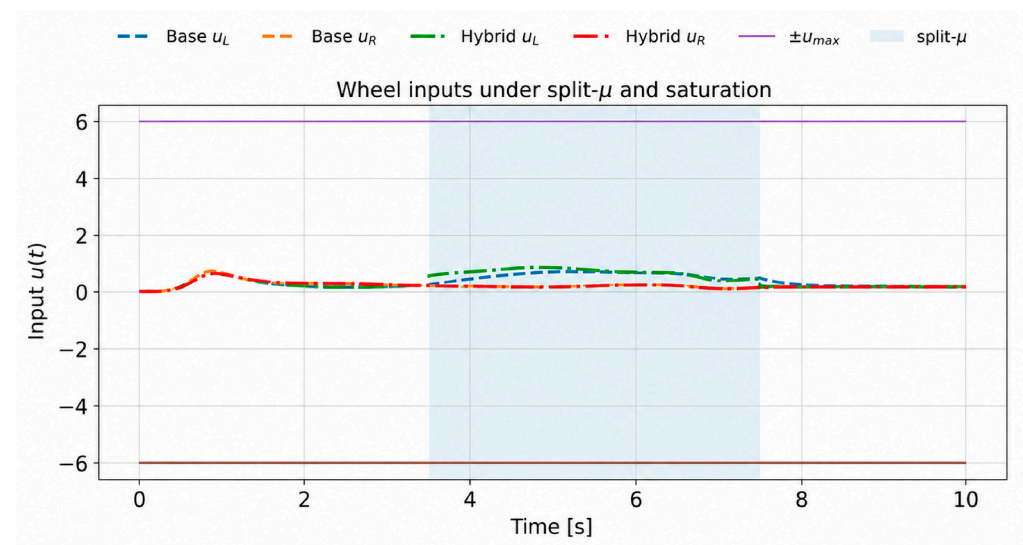


Figure 19. Left- and right-wheel-drive control signals under asymmetric adhesion and actuator signal limitations.

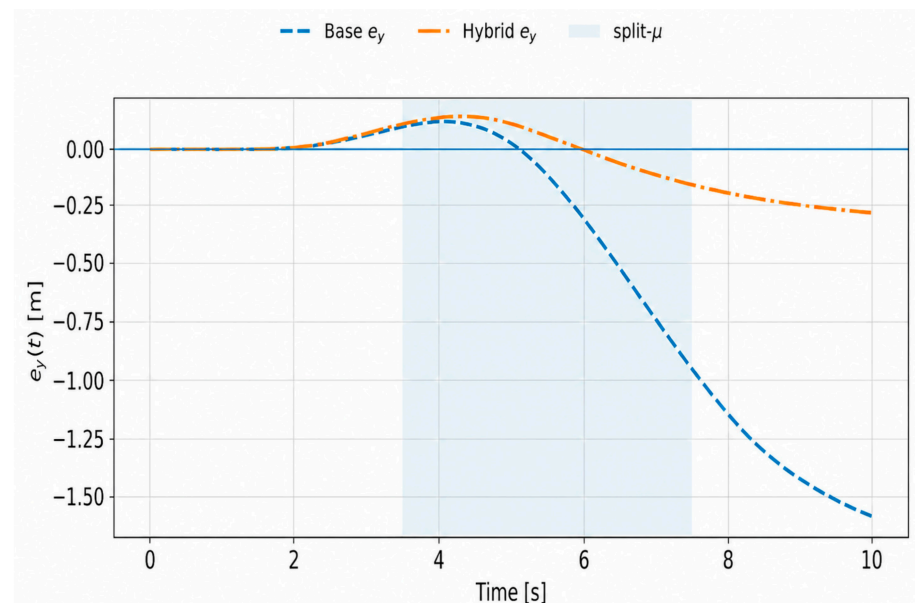


Figure 20. Lateral tracking error of a mobile platform moving under asymmetric adhesion conditions (split- μ mode).

Table 5. Summary of scenario evaluations and practical interpretation of the proposed control architecture.

Evaluation Scenario	Purpose of Evaluation	Main Observed Effect	Practical Interpretation
Nominal motion	Verification of baseline tracking behavior	Stable velocity tracking with reduced transient error when hybrid compensation is enabled	Suitable for normal indoor operation under predictable surface conditions
Parametric uncertainty	Testing robustness to changes in model parameters	Neural residual compensation reduces the influence of model mismatch	Useful when robot mass, load distribution, or drive parameters vary
Variable adhesion	Evaluation under changing wheel–ground contact conditions	Contact-aware compensation improves tracking consistency	Relevant for transitions between dry, wet, or contaminated surfaces
Split- μ disturbance	Testing asymmetric left–right traction	Hybrid control reduces yaw-related degradation and improves drive coordination	Important for robots moving on partially slippery or uneven surfaces
Actuator saturation	Verification of control feasibility under voltage limits	Safety filtering prevents physically infeasible commands	Supports implementation on real motor drivers with limited voltage/current capacity
Neural adaptation effect	Evaluation of adaptive residual correction	Online adaptation improves tracking while Lyapunov supervision limits destabilizing updates	Enables adaptive compensation without fully replacing the physical controller

As shown in Table 5, the proposed control architecture is most beneficial in operating modes where the nominal physical model alone is insufficient, particularly under uncertain contact conditions, asymmetric traction, and model mismatch. In nominal motion, the model-based controller already provides stable behavior, while the neural residual compensator improves transient response and reduces the tracking error. Under disturbed conditions, the Lyapunov-type supervision mechanism prevents excessive adaptive correction and supports stable closed-loop behavior.

From a practical point of view, the proposed architecture can be implemented in real mobile robotic platforms as a drive-level adaptive control layer below the navigation and trajectory-planning modules. The navigation system generates the desired velocity commands, while the proposed controller converts them into left and right actuator commands under voltage, torque, and traction constraints. Potential real-world applications include warehouse mobile robots, inspection robots, service robotic platforms, and outdoor ground robots operating on surfaces with variable adhesion. For experimental implementation, the controller requires measurements or estimates of wheel angular velocities, motor currents, platform velocity, and tracking error, which can be obtained from standard encoders, motor drivers, inertial sensors, and odometry modules.

Overall, the scenario-based discussion confirms that the proposed method is not limited to an idealized simulation case. Instead, it provides a physically interpretable and implementation-oriented control framework that can be adapted to real robotic platforms after experimental validation. The main practical advantage of the approach is the combination of model-based transparency, neural-network adaptability, actuator-level feasibility, and stability-oriented supervision within one drive-control architecture.

It should be noted that the present study is limited to numerical validation of the proposed hybrid control architecture. This limitation was intentionally accepted at this stage because the primary objective of the work was to formulate a physically interpretable control structure, define the residual neural compensation mechanism, and verify its stability-related behavior under reproducible and systematically varied operating conditions. The simulation framework allowed identical initial conditions, disturbance profiles, friction variations, actuator constraints, and uncertainty levels to be imposed on the baseline and hybrid controllers, which is difficult to guarantee in preliminary experimental trials. Nevertheless, the absence of hardware validation limits the possibility of making final conclusions about implementation robustness under real sensor noise, actuator nonlinearities, communication delays, and mechanical imperfections. Therefore, future work will focus on experimental verification of the proposed architecture on a differential-drive mobile robotic platform, including real-time implementation of the neural compensator, validation under split- μ surface conditions, and comparison with measured current, torque, velocity, and tracking-error data.

6. Conclusions

This paper develops a physically based hybrid architecture for controlling the drives of a mobile robot. It combines a model-based controller and an adaptive neural network compensator for residual dynamics with a stability supervision mechanism. The proposed approach preserves the physical interpretability of the model and ensures system adaptation to changing parameters, adhesion conditions, and external disturbances. Numerical simulation results demonstrated a significant improvement in dynamic performance: the root mean square tracking error was reduced from approximately 0.056 to 0.009 m/s (more than sixfold), the peak error from 0.130 to 0.026 m/s (approximately fivefold), and the settling time decreased from 2.1 to 0.9 s. Under asymmetric adhesion conditions (split- μ), an additional reduction in the RMS lateral error of approximately 29% and a decrease in the maximum deviation of 23% were recorded, confirming the improved robustness of the control system. Thus, the proposed hybrid control system demonstrates a quantitatively confirmed improvement in the accuracy, stability, and adaptability of the mobile robotic platform and can be considered a promising solution for practical mobile robot control problems. Further research focuses on experimental validation of the method and expanding its scope of application.

Author Contributions: Conceptualization, A.F., K.A. and T.I.; methodology, A.F. and T.I.; software, K.A.; validation, A.F., K.A., F.O. and K.M.; formal analysis, A.F.; investigation, A.F., F.O. and K.M.; resources, K.A. and K.M.; data curation, F.O.; writing—original draft preparation, A.F.; writing—review and editing, K.A., K.M., T.I. and F.O.; visualization, A.F. and K.A.; supervision, K.A. and T.I.; project administration, K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number BR27195331.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PINNs	Physics-informed neural networks
MPC	Model predictive control
AGV	Automated guided vehicle
RMS	Root mean square error
DC	Direct current
EMF	Electromotive force
FSM	Finite state machine
LQR	Linear quadratic regulator
NN	Neural network
PID	Proportional–Integral–Derivative
PMDC	Permanent magnet direct current
ReLU	Rectified linear unit

References

1. Shin, H.; Kim, S.; Jung, E.; Han, Y.; Lee, N.; Park, S.; Jeon, C.; Woo, J. A Comprehensive Review of Autonomous Mobile Robots in Healthcare: Implications for Patient-Transporting Human-Following Robots. *Robotics* **2025**, *14*, 179. [\[CrossRef\]](#)
2. Rybczak, M.; Popowniak, N.; Lazarowska, A. A Survey of Machine Learning Approaches for Mobile Robot Control. *Robotics* **2024**, *13*, 12. [\[CrossRef\]](#)
3. Guffanti, D.; Pavon, W. Multivariate System Identification of Differential Drive Robot. *Sensors* **2025**, *25*, 5821. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Abadi, A.; Labbadi, M.; Mekki, H.; Ramdani, N. Robust Tracking Control of Wheeled Mobile Robot Based on Fractional-Order Sliding Mode Control with Challenge Validation. *Sensors* **2024**, *24*, 2849. [\[CrossRef\]](#)
5. Huang, H.; Gao, J. Backstepping and Novel Sliding Mode Trajectory Tracking Controller for Wheeled Mobile Robots. *Mathematics* **2024**, *12*, 1458. [\[CrossRef\]](#)
6. Silaa, M.Y.; Bencherif, A.; Barambones, O. Indirect Adaptive Control Using Neural Network and Discrete Extended Kalman Filter for Wheeled Mobile Robot. *Actuators* **2024**, *13*, 51. [\[CrossRef\]](#)
7. Liu, J.; Borja, P.; Della Santina, C. Physics-Informed Neural Networks to Model and Control Robots: A Theoretical and Experimental Investigation. *Adv. Intell. Syst.* **2024**, *6*, 2300385. [\[CrossRef\]](#)
8. Sun, F.; Wu, S.; Li, Z.; Xiong, P.; Chen, W. Physics-informed neural networks for the prediction of robot dynamics considering motor and external force couplings. *Front. Inf. Technol. Electron. Eng.* **2025**, *26*, 2604–2622. [\[CrossRef\]](#)
9. Farea, A.; Yli-Harja, O.; Emmert-Streib, F. Understanding Physics-Informed Neural Networks: Techniques, Applications, Trends, and Challenges. *AI* **2024**, *5*, 1534–1557. [\[CrossRef\]](#)
10. Li, Y.; Liu, L. Physics-Informed Neural Network-Based Nonlinear Model Predictive Control for Automated Guided Vehicle Trajectory Tracking. *World Electr. Veh. J.* **2024**, *15*, 460. [\[CrossRef\]](#)
11. Li, Z.; Guo, J.; Wang, T.; Xiong, X.; Feng, Y.; Li, X. Trajectory Tracking Control for Wheeled Mobile Robots under Unknown Slip Ratio Conditions. *Machines* **2025**, *13*, 765. [\[CrossRef\]](#)
12. Azimirad, V.; Khodkam, S.Y.; Bolouri, A. A new hybrid learning control system for robots based on spiking neural networks. *Neural Netw.* **2024**, *180*, 106656. [\[CrossRef\]](#)
13. Leal, H.M.; Barbosa, R.S.; Jesus, I.S. Control of a Mobile Line-Following Robot Using Neural Networks. *Algorithms* **2025**, *18*, 51. [\[CrossRef\]](#)

14. Szeremeta, M.; Szuster, M. Neural Tracking Control of a Four-Wheeled Mobile Robot with Mecanum Wheels. *Appl. Sci.* **2022**, *12*, 5322. [\[CrossRef\]](#)
15. Razali, M.R.; Mohd Faudzi, A.A.; Shamsudin, A.U.; Mohamaddan, S. A hybrid controller method with genetic algorithm optimization to measure position and angular for mobile robot motion control. *Front. Robot. AI* **2023**, *9*, 1087371. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Xu, S.; Wu, Z. Adaptive learning control of robot manipulators via incremental hybrid neural network. *Neurocomputing* **2023**, *568*, 127045. [\[CrossRef\]](#)
17. Nicodemus, J.; Kneifl, J.; Fehr, J.; Unger, B. Physics-Informed Neural Networks-Based Model Predictive Control for Multi-Link Manipulators. *IFAC-PapersOnLine* **2022**, *55*, 331–336. [\[CrossRef\]](#)
18. Pinosky, A.; Abraham, I.; Broad, A.; Argall, B.; Murphey, T.D. Hybrid Control for Combining Model-Based and Model-Free Reinforcement Learning. *Int. J. Robot. Res.* **2022**, *42*, 337–355. [\[CrossRef\]](#)
19. Chai, B.; Zhang, K.; Tan, M.; Wang, J. An Optimal Robust Trajectory Tracking Control Strategy for the Wheeled Mobile Robot. *Int. J. Control Autom. Syst.* **2024**, *22*, 1050–1065. [\[CrossRef\]](#)
20. Peng, J.; Xiao, H.; Lai, G. Nonlinear Disturbance Observer Incorporated Model Predictive Strategy for Wheeled Mobile Robot's Trajectory Tracking Control. *Int. J. Control Autom. Syst.* **2024**, *22*, 2251–2262. [\[CrossRef\]](#)
21. Guffanti, D.; Murillo, M.F.M.; Sanchez, S.B.; Gutiérrez, J.O.O.; Hinojosa, M.A.; Brunete, A.; Hernando, M.; Álvarez, D. Robust Model Predictive Control with a Dynamic Look-Ahead Distance for a Four-Wheel Differential-Drive Mobile Robot. *Sensors* **2026**, *26*, 520. [\[CrossRef\]](#)
22. Xu, C.; Zhou, X.; Chen, R.; Li, B.; He, W.; Li, Y.; Ye, F. Trajectory Tracking for 3-Wheeled Independent Drive and Steering Mobile Robot Based on Dynamic Model Predictive Control. *Appl. Sci.* **2025**, *15*, 485. [\[CrossRef\]](#)
23. Korayem, M.H.; Namdarpour, F.; Lademakhi, N.Y. Hybrid Stable Adaptive Control Approach for Navigation of Nonholonomic Wheeled Mobile Robots on Slippery Surfaces with Obstacles. *Int. J. Control Autom. Syst.* **2025**, *23*, 1542–1550. [\[CrossRef\]](#)
24. Boubaker, S.; Gaia, J.; Zavalla, E.; Kamel, S.; Alsubaei, F.S.; Bourennani, F.; Rossomando, F. Trajectory Tracking of WMR with Neural Adaptive Correction. *Mathematics* **2025**, *13*, 3178. [\[CrossRef\]](#)
25. Tang, M.; Tang, K.; Zhang, Y.; Qiu, J.; Chen, X. Motion/force coordinated trajectory tracking control of nonholonomic wheeled mobile robot. *Sci. Rep.* **2024**, *14*, 18402. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Teji, M.D.; Zou, T.; Zeleke, D.S. Adaptive Longitudinal Slip Compensation for Wheeled Mobile Robots. *IFAC-PapersOnLine* **2025**, *59*, 385–390. [\[CrossRef\]](#)
27. Gameiro, T.; Pereira, T.; Moghadaspoura, H.; Di Giorgio, F.; Viegas, C.; Ferreira, N.; Ferreira, J.; Soares, S.; Valente, A. Evaluation of PID-Based Algorithms for UGVs. *Algorithms* **2025**, *18*, 63. [\[CrossRef\]](#)
28. Hoseinnezhad, R. A Comprehensive Review of Deep Learning Techniques in Mobile Robot Path Planning. *Appl. Sci.* **2025**, *15*, 2179. [\[CrossRef\]](#)
29. Zhu, Y.; Hasan, W.Z.W.; Ramli, H.R.H.; Norsahperi, N.M.H.; Kassim, M.S.M.; Yao, Y. Deep Reinforcement Learning of Mobile Robot Navigation in Dynamic Environments: A Comparative Analysis. *Sensors* **2025**, *25*, 3394. [\[CrossRef\]](#)
30. Ait Dahmad, H.; Ayad, H.; García Cerezo, A.; Mousannif, H. Adaptive Model Predictive Control for 4WD-4WS Mobile Robot: A Multivariate Gaussian Mixture Model-Ant Colony Optimization for Robust Trajectory Tracking and Obstacle Avoidance. *Sensors* **2025**, *25*, 3805. [\[CrossRef\]](#)
31. Aremu, M.B.; Ahmed, G.; Elferik, S.; Saif, A.-W.A. Autonomous Mobile Robot Path Planning Techniques—A Systematic Review. *Robotics* **2026**, *15*, 23. [\[CrossRef\]](#)
32. Al Mahmud, S.; Lin, X.; Kim, J.-H. Advancements and Challenges in Mobile Robot Navigation: A Comprehensive Review. *J. Intell. Robot. Syst.* **2024**, *110*, 63. [\[CrossRef\]](#)
33. Canudas de Wit, C.; Horowitz, R.; Tsiotras, P. Model-based observers for tire/road contact friction prediction. In *New Directions in Nonlinear Observer Design*; Nijmeijer, H., Fossen, T., Eds.; Lecture Notes in Control and Information Sciences; Springer: London, UK, 1999; Volume 244. [\[CrossRef\]](#)
34. Vieira, D.; Orjuela, R.; Spisser, M.; Basset, M. An Adapted Burckhardt Tire Model for Off-Road Vehicle Applications. *J. Terramechanics* **2022**, *104*, 15–24. [\[CrossRef\]](#)
35. Fazylova, A.; Iliev, T.; Stoyanov, I.; Siemens, E. Height Control System for Wind Turbines Based on Critical Wind Speed Calculation. *Appl. Sci.* **2024**, *14*, 9802. [\[CrossRef\]](#)
36. Zhang, H.; Song, B.; Xu, J.; Li, H.; Li, S. Adhesion Coefficient Identification of Wheeled Mobile Robot under Unstructured Pavement. *Sensors* **2024**, *24*, 1316. [\[CrossRef\]](#)
37. Han, Y.; Lu, Y.; Chen, N.; Wang, H. Research on the Identification of Tyre-Road Peak Friction Coefficient under Full Slip Rate Range Based on Normalized Tyre Model. *Actuators* **2022**, *11*, 59. [\[CrossRef\]](#)
38. Warth, G.; Sieberg, P.; Unterreiner, M.; Schramm, D. A Concept for Using Road Wetness Information in an All-Wheel-Drive Control. *Energies* **2022**, *15*, 1284. [\[CrossRef\]](#)

39. Fazylova, A.; Alipbayev, K.; Iliev, T.; Aden, A. Advanced Control Strategies for Wind Turbine Blade Pitch Optimization. In Proceedings of the 5th IEEE International Conference on Communications, Information, Electronic and Energy Systems (CIEES), Veliko Tarnovo, Bulgaria, 20–22 November 2024; pp. 1–4. [\[CrossRef\]](#)
40. Pantoja-García, J.S.; Rodríguez-Molina, A.; Villarreal-Cervantes, M.G.; Palma-Huerta, A.A.; Aldape-Pérez, M.; Sandoval-Gutiérrez, J. Generalized Kinematic Modeling of Wheeled Mobile Robots: A Unified Framework for Heterogeneous Architectures. *Mathematics* **2026**, *14*, 415. [\[CrossRef\]](#)
41. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv* **2019**. [\[CrossRef\]](#)
42. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [\[CrossRef\]](#)
43. Fazylova, A.; Tultayev, B.; Iliev, T.; Stoyanov, I.; Kabasheva, M.; Kosunalp, S. Experimental Study of an Industrial Data Transmission Network in the Automatic Control System of a Wind Turbine. *Machines* **2024**, *12*, 746. [\[CrossRef\]](#)
44. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.