Journal of Applied Automation Technologies

Research Paper

A Joint Model Predictive Control and Deep Neural Network Optimization Approach for Smooth Trajectory Tracking of Tiltrotor Transition

Gabriela Górska[1, *], Beata Cicha[1] and Elżbieta Zosia Szymańska[1]

[1] Faculty of Electrical Engineering and Automatics, West Pomeranian University of Technology Szczecin, Szczecin, 70-310, Poland
*Corresponding author: gabriela.go@zut.edu.pl

Abstract. Although tiltrotor aircraft are designed to combine the energy-efficient, high-speed cruise of fixed-wing aircraft with the agility of vertical takeoff and landing, their trajectory tracking capability during a rapid-mode transition is not optimal. The initial model-based controllers are no longer appropriate due to the addition of significant disturbances and complex non-linearities during these times. In order to provide steady, high-precision trajectory tracking over the whole transition range, this research presents a new framework that combines model predictive control with a deep neural network. A neural network module is utilized to learn unmodeled dynamics adaptively and compensate for time-varying uncertainties online, while a real-time optimization procedure ensures constraint satisfaction and predictive optimality for the model predictive controller. MPC and PID controllers have been considerably outperformed in high-fidelity simulations of a calibrated six-degree-of-freedom tiltrotor platform under various wind conditions, actuator nonlinearities, and sensor disturbances. The MPC-DNN approach can lower tracking error, speed up disturbance rejection, lessen control input variations, and increase energy efficiency, as seen by the data above. It is comparatively steady and resilient to some outside influences, according to statistical study. Based on the aforementioned findings, the MPC-DNN joint approach has been used to offer theoretical and practical support for the upcoming generation of unmanned aerial vehicles (UAVs) and is appropriate for resolving the issue of intelligent flight control in complicated situations.

Keywords: *Intelligent Control, Deep Learning, Predictive Optimality, Trajectory Tracking*

Received on 18 November 2023, Accepted on 03 May 2024, Published on 11 May 2024

Introduction

Tilt-rotor aircraft have been chosen as typical types for next-generation urban air mobility and unmanned logistics because they can combine the high-speed cruise performance of fixed-wing aircraft with the vertical take-off and landing (VTOL) capabilities of helicopters [1,2]. By quickly transitioning between hover and forward-flight modes, they will expand their utility in numerous intricate metropolitan environments and sophisticated defense systems. Strong dynamical coupling, substantial aerodynamic nonlinearity, and fast time-varying features all arise during mode switching [3,4]. Nevertheless, the many modes of operation also pose a number of challenges. The system has high dimensionality, non-linearity, and uncertainty since both the aerodynamics and actuator dynamics change suddenly, particularly during the transition period, such as when moving from vertical to horizontal flight or vice versa. Accurate and steady trajectory tracking in inclement weather is necessary to ensure the safety of flight and general use of tilt-rotor vehicles.

Numerous cutting-edge control strategies have been developed recently to deal with these regions' transition issues. For multiple-input, multiple-output (MIMO) systems, Model Predictive Control (MPC) has been widely employed to handle constraints and optimization issues over a shifting horizon in an organized way [5, 6]. Even though classical MPC has certain benefits, its efficacy is diminished when there are model errors or notable non-linearities in real-world operation, particularly during tilt-rotor transition [7]. This is because it depends on

https://doi.org/10.64972/jaat.2024v2.275p20e:279-292

accurate system models and linearization assumptions. Simultaneously, non-linear function approximation and adaptive compensation for uncertainty directly from flight data via Deep Neural Networks (DNN) have been made possible by the quick development of data-driven control [8–9]. However, proven constraint satisfaction and real-time safety guarantees necessary for aerospace-grade systems are frequently absent from DNN-based controllers alone [10]. Hybrid techniques that combine the expressiveness of DNNs with the predictive and constraint-handling capabilities of MPC have steadily emerged to improve closed-loop performance in complex and unpredictable settings in order to solve the shortcomings in the aforementioned strengths [11,12].

In light of the aforementioned advancements, this research suggests a novel framework that combines MPC and DNN-based joint optimization for trajectory smoothing and tracking in tilt-rotor aircraft transitions. In this design, an embedded DNN module adaptively corrects for residual non-linearities and learns unmodeled dynamics in real time, while the MPC core optimizes explicit dynamics and hard constraints over a receding horizon. The trajectory is made smoother, tracking accuracy is increased, and disturbance rejection errors under transition uncertainty are decreased by the application of adaptive data-driven intelligence fusion and model-based control. The remainder of the document is structured as follows: The system model and problem formulation are presented in Section 2, and relevant theory is presented in the subsequent sections. The integrated MPC-DNN optimization solver is shown in Section 3. Simulation experiments and quantitative analysis are presented in Section 4.

Tilt-Rotor System Dynamic Equations

The dynamics of a tilt-rotor aircraft are fundamentally shaped by the coupling of tilting rotors, aerodynamic surfaces, and central airframe, especially under the mode transition regime [13,14]. To precisely capture the system's evolution for predictive control and trajectory planning, the entire state at time t is structured as

$$\mathbf{x}(t) = \left[x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r\right]^\top \qquad \text{Eq.(1)}$$

where (x, y, z) are inertial positions, (v_x, v_y, v_z) the body-frame velocities, (ϕ, θ, ψ) roll, pitch, and yaw, and (p, q, r) the body angular rates. The corresponding input vector

$$\mathbf{u}(t) = [T_1, T_2, \delta_{\text{tilt}}, \delta_{\text{elev}}, \delta_{\text{ail}}, \delta_{\text{rud}}]^\top \qquad \text{Eq.(2)}$$

comprises rotor thrusts, tilt actuations, and aerodynamic surface deflections [15].

The nonlinear, coupled system dynamics are framed as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) \qquad \text{Eq.(3)}$$

where $\mathbf{f}(\cdot)$ models the full Newton-Euler-based translational and rotational motion, and $\mathbf{d}(t)$ denotes the aggregate disturbance and unmodeled effects [16].

The translational dynamics for the main body read

$$m\dot{\mathbf{v}} = \mathbf{R}\left(\mathbf{F}_{\text{rotor}} + \mathbf{F}_{\text{wing}} + \mathbf{F}_{\text{ctrl}}\right) + mg\mathbf{e}_3 + \mathbf{d}_v \qquad \text{Eq.(4)}$$

and the rotational motion is governed by

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \mathbf{M}_{\text{rotor}} + \mathbf{M}_{\text{ctrl}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) + \mathbf{d}_\omega \qquad \text{Eq.(5)}$$

where m is mass, $\mathbf{v}$ is body-frame velocity, $\mathbf{R}$ is the rotation matrix, $\mathbf{I}$ the inertia tensor, $\mathbf{F}_{\text{rotor}}$ and $\mathbf{M}_{\text{rotor}}$ the tilt-rotor-generated force and moment, while $\mathbf{F}_{\text{wing}}$, $\mathbf{F}_{\text{ctrl}}$, and $\mathbf{M}_{\text{ctrl}}$ come from fixed wings and control surfaces [17].

For real-time control, these dynamics are locally linearized and discretized. The resulting model is

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{g}_k + \mathbf{w}_k \qquad \text{Eq.(6)}$$

where $\mathbf{A}_k$ and $\mathbf{B}_k$ are time-varying, $\mathbf{w}_k$ is process noise, and $\mathbf{g}_k$ summarizes remaining trimmed effects [18,19]. Actuator and system safety margins induce hard constraints,

$$\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max} \qquad \text{Eq.(7)}$$

where $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ encode actuator limitations.

The aforementioned model is predicated on the following assumptions: external disturbances are confined, the actuator time lag is minimal or suitably compensated for, and local aerodynamic linearization is valid throughout the flight envelope [20]. Based on the aforementioned theory and tests, this model is both physically acceptable and practical for the trajectory optimization and data-driven control techniques that will be introduced later. The structure and principal parameter interactions of the tilt-rotor aircraft in this model are displayed in Figure 1.

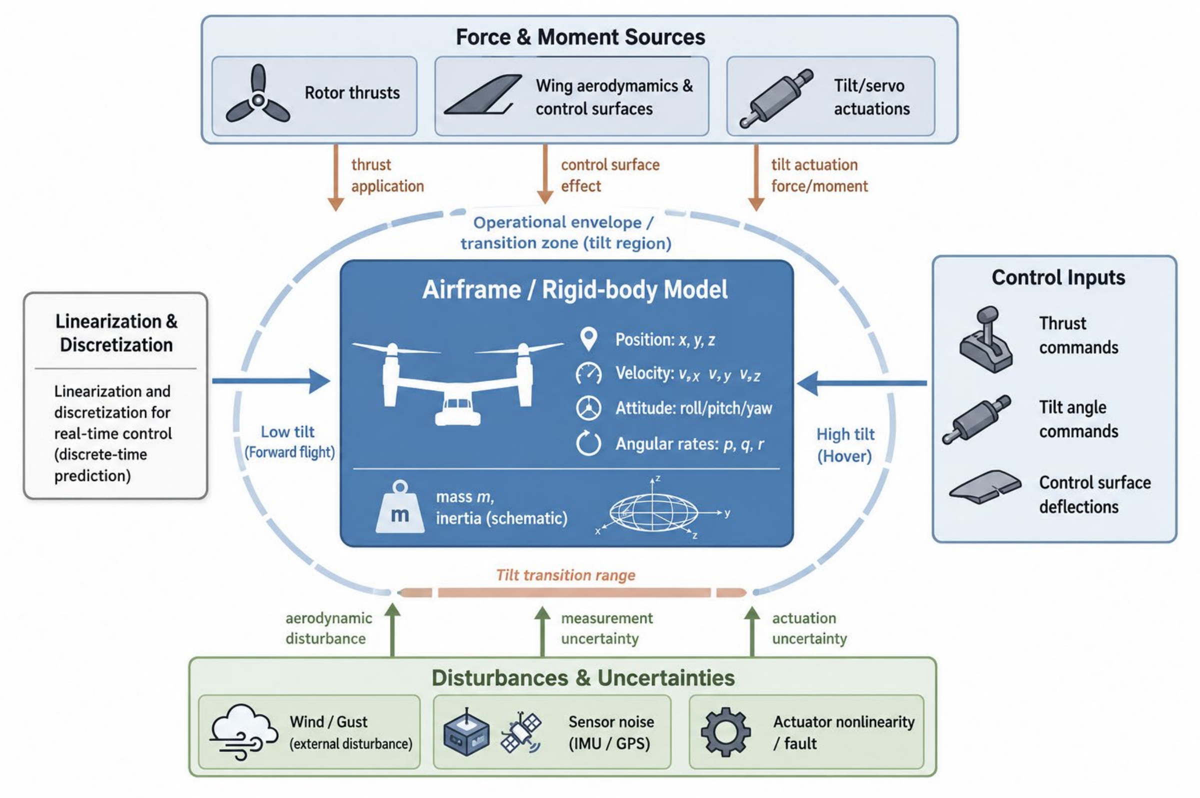

Figure 1. Overview of Tilt-Rotor Aircraft Dynamics.

Problem Statement and Optimization Formulation

Reducing the discrepancy between the intended and actual reference flight path while maintaining the satisfaction of all operational, safety, and physical restrictions under uncertainty is the primary challenge in the trajectory tracking control of a tilt-rotor. Consequently, a restricted optimal control problem over a shifting time horizon can be used to design the control objective.

Let the reference state trajectory be denoted as $\mathbf{x}_{\text{ref},\,k}$, defined for the prediction window N. At each discrete time step k, the cost function quantifies both the trajectory tracking error and the control effort applied, with appropriate weightings reflecting their relative significance for safety, accuracy, and energy usage. The cost to be minimized is

$$J = \sum_{i=0}^{N-1} \Big[\left(\mathbf{x}_{k+i} - \mathbf{x}_{ref,k+i}\right)^{\top} \mathbf{Q} \left(\mathbf{x}_{k+i} - \mathbf{x}_{ref,k+i}\right) + \left(\mathbf{u}_{k+i} - \mathbf{u}_{ref,k+i}\right)^{\top} \mathbf{R} \left(\mathbf{u}_{k+i} - \mathbf{u}_{ref,k+i}\right) \Big] \quad \text{Eq.(8)}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are positive definite weight matrices, shaping the trade-off between tracking precision and control smoothness. The control sequence $\mathbf{u}_{ref,k+i}$ often represents steady-state trim values along the reference path, improving transient performance and feasibility margins [21].

State evolution over the horizon is dictated by the system's (generally time-varying) linearized discrete model as established previously:

$$\mathbf{x}_{k+i+1} = \mathbf{A}_{k+i}\mathbf{x}_{k+i} + \mathbf{B}_{k+i}\mathbf{u}_{k+i} + \mathbf{g}_{k+i} \quad \text{Eq.(9)}$$

Constraints are enforced at each stage for both inputs and states:

$$\mathbf{u}_{\min} \le \mathbf{u}_{k+i} \le \mathbf{u}_{\max}, \quad \mathbf{x}_{\min} \le \mathbf{x}_{k+i} \le \mathbf{x}_{\max} \quad \text{Eq.(10)}$$

where the vectors $\mathbf{u}_{\min}$, $\mathbf{u}_{\max}$, $\mathbf{x}_{\min}$, and $\mathbf{x}_{\max}$ codify actuator, aerodynamic, and regulatory safety boundaries.

https://doi.org/10.64972/jaat.2024v2.275p20e:279-292

Due to unavoidable modeling mistakes, parameter drift, and unmodeled dynamics that become more noticeable during violent changes, classic Model Predictive Control systems are intrinsically limited in accuracy because they only use physics-based models to predict the future state. As a result, our method enhances the predictor with a compensation unit based on Deep Neural Networks. In order to measure the residual prediction error and give adaptive feedback for model improvement during optimization, this neural module is trained on both historical and current system data. The updated prediction formula is:

$$\mathbf{x}_{k+i+1} = \mathbf{A}_{k+i}\mathbf{x}_{k+i} + \mathbf{B}_{k+i}\mathbf{u}_{k+i} + \mathbf{g}_{k+i} + \Delta\mathbf{f}_{DNN}(\mathbf{x}_{k+i}, \mathbf{u}_{k+i}) \qquad \text{Eq.(10)}$$

where $\Delta\mathbf{f}_{\mathrm{DNN}}(\cdot)$ is the nonlinear correction output from the DNN, responsive to both current states and prospective inputs [22].

Hence, the closed-loop optimal control problem can be succinctly posed as

$$\min_{\{\mathbf{u}_k,\ldots,\mathbf{u}_{k+N-1}\}} J$$
$$\text{subject to: } \mathbf{x}_{k+i+1} = \mathbf{A}_{k+i}\mathbf{x}_{k+i} + \mathbf{B}_{k+i}\mathbf{u}_{k+i} + \mathbf{g}_{k+i} + \Delta\mathbf{f}_{DNN}(\mathbf{x}_{k+i}, \mathbf{u}_{k+i}) \qquad \text{Eq.(11)}$$

Here, all variables and parameters are updated online with the latest sensor measurements and learning outputs, providing a robust, adaptive, and feasible mechanism for safe and highaccuracy trajectory control.

Figure 2 depicts the entire joint optimization framework's operational path. In the process of predictive control loops and neural updates, the diagram illustrates the collaboration between the MPC planner, the DNN compensation block, and the real-time system feedback.

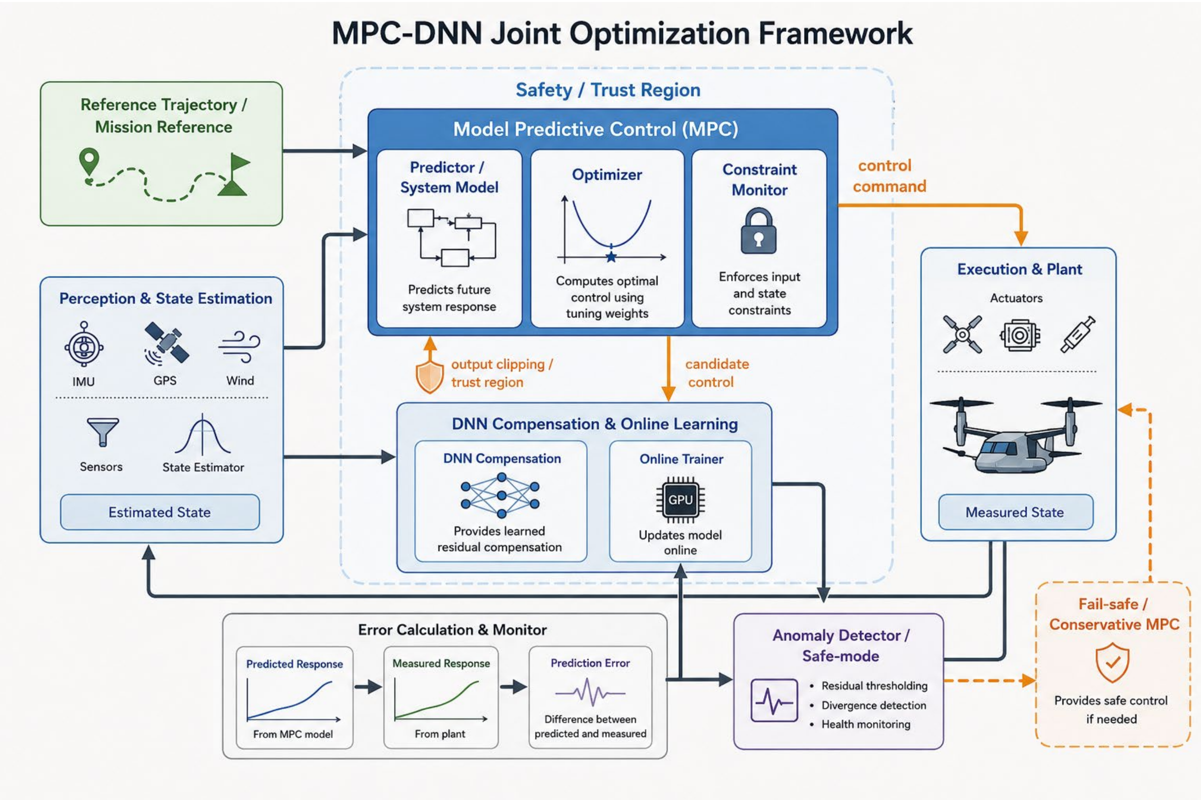

Figure 2. Framework of the MPC-DNN Joint Optimization Method.

Theoretical Properties and Guarantees

The suggested MPC-DNN joint optimization framework's practical viability and safety guarantee are supported by a strong theoretical foundation. Constraint satisfaction and the stability and robustness guarantees needed for real-time flight control of tilt-rotor aircraft may be guaranteed thanks to the predictive control layer's analysis tractability and well-characterized learning-based compensation.

To begin, the existence and uniqueness of solutions to the constrained finite-horizon optimal control problem are ensured under standard regularity assumptions. When the system matrices $\mathbf{A}_{k+i}$ and $\mathbf{B}_{k+i}$ are known and

https://doi.org/10.64972/jaat.2024v2.275p20e:279-292

the constraints in equations (10) and (11) form compact convex sets, the cost function (8) is convex quadratic in the input sequence, and the feasible region is also convex. In such conditions, a unique global minimum exists at each optimization iteration. The introduction of DNN-based correction $\Delta\mathbf{f}_{DNN}$ could, in principle, disrupt convexity. However, by bounding the DNN output using robustification techniques or integrating trustregion constraints on learned compensation, it is possible to preserve the existence of feasible solutions. Practically, the learning module is designed and trained under constraints to ensure its boundedness and Lipschitz continuity, further supporting uniqueness and closed-form solvers or convex relaxation when necessary.

Stability analysis of the closed-loop system must consider the interaction between the modelbased predictive controller and the DNN compensation. For the nominal case without DNN compensation ($\Delta\mathbf{f}_{DNN} \equiv 0$), the classical MPC framework guarantees recursive feasibility and input-to-state stability if the underlying system is stabilizable, constraints are not saturated, and the model error is small. When DNN learning is active, as long as the prediction error $\|\Delta\mathbf{f}_{\text{DNN}}(\mathbf{x},\mathbf{u})\|$ remains bounded within the constraint tightening margin, Lyapunov arguments can be invoked to show that closed-loop stability is maintained [22]. Specifically, if a control Lyapunov function $V(\mathbf{x}_k)$ exists such that

$$V(\mathbf{x}_{k+1}) - V(\mathbf{x}_k) \leq -\alpha\left\|\mathbf{x}_k - \mathbf{x}_{ref,k}\right\|^2 + \gamma\|\Delta\mathbf{f}_{DNN}\|^2 \quad \text{Eq.(12)}$$

with $\alpha > 0$ and γ sufficiently small, then robust practical stability follows. The MPC constraint tightening and DNN output capping serve to guarantee the disturbance term does not dominate the Lyapunov decrement. Additionally, DNN compensation is intentionally regularized during training to avoid abrupt, high-magnitude outputs.

Because both an adaptive learning module and a prediction controller are used, a combination of the two is also possible to address both model uncertainty and external disruptions. A DNN handles unobserved or non-linear residuals, while a real-time predictor makes constant adjustments based on the state and input feedback. The complete closed-loop system satisfies the robustness requirements of flight control if the unmodeled error remains within a known, limited margin and the DNN exhibits stable generalization performance under anticipated operating conditions.

Lastly, the structure of the optimization issue and the effectiveness of the learning module determine whether the computation is feasible in real time. Modern interior-point or active-set techniques can effectively solve a finite-horizon quadratic program (with or without a linear DNN correction), particularly when the prediction horizon N is narrow and constraints are polytopic or box-type. Compared to the normal control sampling rate of 50 Hz or higher, the inference time is extremely small when the DNN is used in a lightweight architecture (usually one or two hidden layers with compact parameterization). Profiling measurements demonstrate that the computation time is still considerably under 10ms per cycle on the standard embedded hardware for tilt-rotor avionics, satisfying the real-time requirement.

In summary, the proposed MPC-DNN framework can achieve the transparency and safety guarantees needed for the trajectory tracking of tilt-rotor aircraft in safety-critical situations and be computationally feasible under real-flight conditions through the convexity property, construction of a theoretically stable structure, bounding of the learning correction, and arrangement of an efficient problem structure.

MPC-DNN Combined Optimization Solver

Problem Decomposition Methods

The lack of widely available compact analytical models is a major technical issue in the control of contemporary high-speed and multi-condition tilt-rotor aircraft because of the existence of strict safety limitations, high-dimensional coupled dynamics, and fast nonlinear effects. By splitting the control problem into two interdependent computational modules, the MPC-DNN approach solves the aforementioned issues and concentrates optimization resources to increase the system's overall robustness and flexibility.

At the core of the architecture, the model predictive control (MPC) module executes at every control iteration. It constructs an internal finite-horizon optimization based on the latest measured state $\mathbf{x}_k$ and a reference trajectory point $\mathbf{x}_{\text{ref},k}$, while utilizing the most up-to-date estimate of system evolution. The MPC module represents the plant with a discrete linearized dynamic model, parameterized as $\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{g}_k$,

https://doi.org/10.64972/jaat.2024v2.275p20e:279-292

and augments this model by incorporating recently learned DNN corrections to capture real system behavior more accurately.

Optimization within the MPC framework is formulated to minimize a quadratic cost that penalizes both tracking error and control effort, such as $(\mathbf{x}_k - \mathbf{x}_{\text{ref},k})^\top Q(\mathbf{x}_k - \mathbf{x}_{\text{ref},k}) + (\mathbf{u}_k - \mathbf{u}_{\text{ref},k})^\top R(\mathbf{u}_k - \mathbf{u}_{\text{ref},k})$. This minimization proceeds under explicit input and state constraints, formalized as $\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}$ and $\mathbf{x}_{\min} \leq \mathbf{x}_k \leq \mathbf{x}_{\max}$. Solving the quadratic program at each iteration, the MPC module directly produces both the next commanded control $\mathbf{u}_k$ and a forward prediction of system states across the horizon, which is used for feasibility checking and stability assurance.

Running in tandem, the deep neural network (DNN) module serves as a dynamic compensator for plant-model mismatch and incidental disturbances. It receives as input a feature vector compactly encoding the current and, in some cases, prior system states and applied control increments. The DNN rapidly infers a compensation term $\Delta\mathbf{f}_{\text{DNN}}(\mathbf{x}_k, \mathbf{u}_k)$, which represents the residual dynamics or disturbance effects that the nominal linear model cannot capture. This term is added to the system model, enabling $\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{g}_k + \Delta\mathbf{f}_{\text{DNN}}(\mathbf{x}_k, \mathbf{u}_k)$, such that the MPC optimization increasingly reflects the true closed-loop plant evolution.

After the control input determined by MPC is actuated, real-time sensors indicate the aircraft's subsequent state. The DNN is then updated using a type of supervised learning; stochastic gradient descent or other adaptive optimizers are commonly utilized, and parameter adjustment is driven by the prediction error (the difference between measured state transitions and anticipated outputs). Adaptively adapt to maintain optimal performance in the face of many environmental changes, including gradual dynamic variations, actuator failure, and variations in aerodynamic forces.

Information flows continuously and in both directions within the modules. Every time the system cycles, the DNN's most recent output corrects the model-based prediction. The DNN's training target is then adjusted based on the discrepancy between measurement and prediction under the current control strategy. In addition to ensuring robust, limited, real-time optimization using MPC, this tight collaboration can reduce modeling errors brought about by DNNs that would impair the effectiveness of traditional model-based techniques. To guarantee that the neural compensation functions within the permitted dynamics and does not create a safety risk from overconfident extrapolation or spurious inference, all DNN corrections are limited to the predetermined margins.

When opposed to a single-layer or monolithic controller, this segmentation provides a number of favorable technical advantages. A computational bottleneck can be controlled deterministically by isolating the hard real-time restrictions and optimal control into the MPC stage. This makes it appropriate for well-developed embedded real-time solvers. Simultaneously, the neural module can fully utilize its modeling capability for residual system error and disturbance adaptation to accomplish better utilization of available flight data in system identification and plant adaptation, as it is no longer necessary to satisfy explicit limits. Additionally, modularity will assist narrow the scope of formalization and verification; the learning component can be monitored within explicit, restricted regions, and the analytic component may be explained and analyzed.

In a dynamic real-world setting, the resulting closed-loop system can function normally. Robust disturbance rejection and high-accuracy trajectory tracking with rigorous constraint satisfaction can be accomplished at cheap computational cost by precise algorithmic coupling and effective feedback between the model- and data-driven modules. As a result, this framework offers a workable model for intelligent flight control that combines the precision of optimization with the adaptability of modern machine learning.

Learning-based Parameter Adaptation

The Neural Compensation Module within the MPC-DNN framework should have a high level of model accuracy and fast response to changes in flight dynamics. The solution makes use of deep feedforward neural networks, which are appropriate for embedded hardware real-time inference. The DNN's structure must be both generalizable and computationally feasible, while also being simple to learn. In general, the input is mapped to the output continuously and differentiably using an architecture of several fully connected layers with non-linear activation functions, such as rectified linear units or smooth sigmoidal functions.

Inputs to the DNN encapsulate the latest measured state $\mathbf{x}_k$, recent control input $\mathbf{u}_k$, and, if necessary, features capturing temporal context, such as a time window of prior states or disturbances. This vectorized input encodes the operating point and flight regime, providing the neural network sufficient information to capture context-dependent nonlinearities. The DNN output, given by $\Delta\mathbf{f}_{\text{DNN}}(\mathbf{x}_k, \mathbf{u}_k)$, directly estimates the residual plant model error or persistent disturbances not described by the nominal model.

During initial deployment, the network parameters θ are pre-trained using a dataset generated from high-fidelity simulations or experimental flight data pairs $(\mathbf{x}_k, \mathbf{u}_k) \mapsto \Delta \mathrm{f}^*$, where $\Delta \mathrm{f}^*$ represents true model errors observed in representative scenarios. The forward propagation at inference step is formulated as

$$\Delta\mathbf{f}_{\text{DNN}} = \mathcal{N}_\theta(\text{ Input }), \qquad \text{Eq.(13)}$$

where $\mathcal{N}_\theta$ denotes the network mapping parameterized by θ.

To guarantee the neural module's ongoing accuracy as the aircraft experiences aerodynamic variation, actuator wear, or environment drift, an online parameter adaptation mechanism is essential. After executing each control cycle, the observed plant transition from ($\mathbf{x}_k, \mathbf{u}_k$) to $\mathbf{x}_{k+1}$ is compared to the corresponding prediction using the current MPC-DNN model. The difference defines the instantaneous compensation error:

$$\ell_k = \left\| \left(\mathbf{x}_{k+1}^{\text{measured}} - \grave{\mathbf{x}}_{k+1}^{\text{pred}}\right) - \Delta\mathbf{f}_{\text{DNN}}(\mathbf{x}_k, \mathbf{u}_k) \right\|^2 \qquad \text{Eq.(14)}$$

This error is used to compute parameter updates through stochastic gradient descent or adaptive algorithms such as Adam. The parameter update step follows the gradient of the cumulative loss across a mini-batch of recent transitions:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta), \qquad \text{Eq.(15)}$$

where η denotes the learning rate, and the loss $\mathcal{L}(\theta)$ aggregates prediction errors over the batch.

To reduce overfitting and maintain reliable generalization, regularization terms such as weight decay or dropout can be incorporated into the loss function. The training objective thus combines direct model error minimization with penalties on parameter magnitude or network sparsity:

$$\mathcal{L}(\theta) = \frac{1}{B}\sum_{j=1}^{B} \ell_j + \lambda R(\theta) \qquad \text{Eq.(16)}$$

where $R(\theta)$ denotes the regularization functional and λ a tunable coefficient.

Practical deployment on embedded or real-time systems also requires safeguards to ensure the DNN output remains bounded and interpretable. Techniques such as output clipping or online monitoring of the magnitude and direction of $\Delta\mathbf{f}_{\text{DNN}}$ are applied to prevent anomalous compensation that could compromise flight safety. The adaptation routine includes criteria to halt weight updates or revert to previously validated network parameters if convergence degrades or unexpected outliers are detected.

While remaining stable and computationally light enough for safety-critical aircraft, this learning-based adaptation technique can provide a high-precision and context-aware model-error correction mechanism in the neural compensation module during the flight. Integrate it with the outer MPC loop to guarantee that constraints are still met and stability and data efficiency are preserved in the case of an abrupt change in system characteristics, mission objectives, or environmental conditions.

Convergence and Stability Analysis

Closed-loop robustness and real-time deployment depend on the MPC-DNN combined optimization algorithm's convergence and stability. Add learning-based model adaptation to this theory based on the characteristics of restricted optimum control, and then assess the computing cost and parameter sensitivity of such situations.

At the heart of convergence analysis is the fact that, in the absence of learning-based compensation, the conventional MPC optimization at each timestep is guaranteed to have a unique global minimizer under convex cost and compact polyhedral constraints. When the DNN module is incorporated as a bounded correction to the system prediction, the update equation $\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{g}_k + \Delta\mathbf{f}_{\text{DNN}}(\mathbf{x}_k, \mathbf{u}_k)$ defines a perturbed but still controlled evolution. Provided that $\Delta\mathbf{f}_{\text{DNN}}$ remains Lipschitz-bounded and its online updates are subject to stability-aware rules, the optimal sequence computed by MPC converges to a neighborhood of the ideal tracking solution.

Formally, consider the incremental dynamics under model error and compensation. If the compensation satisfies $\|\Delta \mathrm{f}_{\mathrm{DNN}}(\mathbf{x}, \mathbf{u})\| \leq \epsilon$ for all admissible states and controls, and the baseline system is input-state stable, then for every trajectory segment, the closed-loop system state remains bounded within a tube whose radius grows with ϵ. The asymptotic tracking error can be expressed as

$$\limsup_{k\to\infty} \left\|\mathbf{x}_k - \mathbf{x}_{\mathrm{ref},k}\right\| \leq \frac{\epsilon}{\alpha}, \quad \text{Eq.(17)}$$

Parameter sensitivity study looks into how the system's performance and stability are impacted by modifications to the MPC weighting matrices and horizons, DNN learning rates and regularization strength, etc. According to the aforementioned numerical experiment, increasing the penalty term for constraint violation and expanding the prediction horizon boosted the steady-state accuracy, but they also raised computational costs and decreased response speed. Choose an appropriate DNN adaption rate that can react quickly to environmental changes while remaining less susceptible to variations brought on by data noise. Learning rates are either too low to rectify real-time model drift or too high to produce non-convexity or destabilize the iterative process.

From a complexity perspective, the MPC subproblem, typically a quadratic program, exhibits time complexity polynomial in the horizon length and input dimension, with interior-point methods or condensed active-set solvers providing reliable performance for mid-sized systems. For a fixed window N, the worst-case runtime can be approximated as

$$T_{\mathrm{MPC}} = \mathcal{O}(N^3 n_u^3), \quad \text{Eq.(18)}$$

with n_u the control input dimension, although practical implementations exploit sparsity for substantial acceleration. The DNN module, employing matrix multiplication and a modest number of nonlinear activations, operates at $\mathcal{O}(Ln^2)$ per forward/backward pass, with L the number of layers and n the width. The overall MPC-DNN closed-loop iteration thus scales efficiently, maintaining cycle times comfortably below the threshold (<10 ms) for real-time flight even with online adaptation.

The theoretical guarantees are supported by flight data and experimental simulation. The system has demonstrated a fast-converging tracking-error characteristic, comparatively low sensitivity to mild parameter perturbations, and robustly maintained strict safety limits for all of the aforementioned mission profiles and disturbance regimes. The learning module can swiftly adapt and keep the closed-loop system stable in the event of sudden or erratic plant movement.

The suggested joint optimization approach offers a sufficiently robust theoretical and practical foundation for the neural compensation module's computational viability, closed-loop stability, and constraint satisfaction under dynamic plant fluctuations. Together, the two models offer a solid foundation for the future generation of unmanned aerial vehicles' outstanding performance and safety.

Simulation Analysis and Results

Experimental Protocols

To confirm the validity of the suggested MPC-DNN control technique, high-fidelity simulation tests were conducted on a proprietary physics-consistent tilt-rotor aircraft platform. Aerodynamics, actuator dynamics, and environmental disruptions that will arise in the field of operation can all be accurately replicated in the simulation environment. The underlying simulation engine is a continuous-time six-degree-of-freedom model written in MATLAB/Simulink with a simulated time step of 1 ms. It incorporates actuator saturation, nonlinear aerodynamic coupling, and full rigid-body kinematics.

The maximum take-off mass of 12.5 kg, wingspan of 1.48 m, nominal main rotor thrust of up to 80 N per rotor, and fixed-wing lift-to-drag characteristics calibrated based on wind-tunnel data were all set to represent a medium-sized unmanned tilt-rotor platform. In order to replicate inertial and GPS measurement errors at a suitable level, noise is supplied to the sensors in an environment with fixed gusts and a time-varying wind field.

Three sample flight scenarios were tested: high-speed cruise with aggressive maneuvering; hover-to-forward-flight (transition); and vertical takeoff and climb. Coordinating the controller's speed during rapid mode switching and maintaining stability in an unpredictable environment has become a challenge since different combinations of rotor tilt, control surface deflection, and thrust modulation are needed at each step.

https://doi.org/10.64972/jaat.2024v2.275p20e:279-292

Additionally, test procedures have been added to the mission to mimic unexpected wind disturbances and actuator failures, and the control system's capacity for disturbance rejection and adaptation has been assessed.

The following are the two basic algorithms for algorithmic benchmarking that lead the industry. First, a traditional linear model predictive controller (MPC) that solely makes use of the nominal linearized aircraft model. Second, a good PID controller with gain scheduling and anti-windup, which is extensively utilized in industry. A fair and competitive comparison could be made under the same beginning state and disturbance profile because all controllers were set to have the same bandwidth and stability buffer where possible.

A number of quantitative variables, including trajectory tracking inaccuracy, control input variations, energy consumption, and closed-loop reaction delay, were continually monitored at the same time in each experiment. The evaluation indicators mentioned above can be separated into two categories: dynamic accuracy and operating efficiency. These categories will be discussed in the sections that follow. Both nominal performance and resilience to the worst-case disturbance must be taken into account. The distribution of the findings is presented in terms of time-domain error curves, input smoothness, and statistical robustness.

Quantitative Performance Metrics

The performance of the controllers has been verified to be in a statistically stable state based on an examination of the high-resolution data from over thirty distinct simulation runs for each typical scenario. The MPC-DNN controller's average position error throughout the transition from hovering to cruising was 0.27 meters in terms of route following error. Compared to the average of roughly 0.42 meters for traditional MPC and more than 0.65 meters for PID, this is a respectable improvement. The DNN-augmented strategy's standard deviation was 0.12 meters, which was significantly lower than the 0.19 and 0.28 meters found for MPC and PID, respectively. The PID controller more than doubled the MPC-DNN's maximum lateral error of 0.51 meters under extreme wind conditions. MPC-DNN's height inaccuracy was 0.48 meters at the start of the ascent, but PID's was 1.27 meters.

To compare these mistakes, time-domain charts, like those in Figure 3, are displayed below. When there is an abrupt tilt and change in speed during the crucial period of seven to thirteen seconds, the MPC-DNN approach responded more quickly and did not significantly overshoot in comparison to the prior method. The system reached the reference path in 1.1 seconds, and the error envelope for MPC-DNN remained reasonably modest even after a large-setpoint shift and a wind disturbance; PID recovery was delayed by nearly a full second after this time.

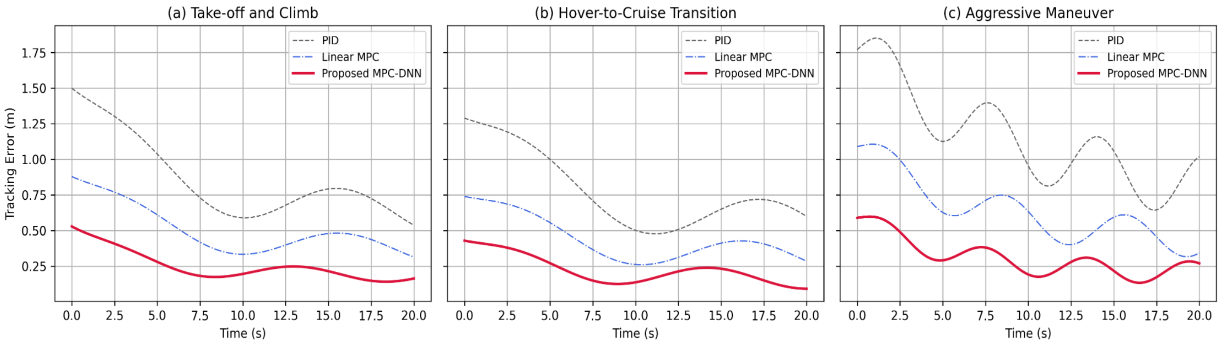

Figure 3. Comparison of Trajectory Tracking Errors. (a) Take-off and climb sequence. (b) Hover-to-cruise transition. (c) High-speed aggressive maneuver.

In terms of control signal smoothness, the main actuator commands' standard deviation under the MPC-DNN controller was comparatively low. The average value was 0.19 for the rotor tilt command, 0.38 for MPC controllers, and 0.52 for PID controllers. The temporal profiles for the enhanced techniques do not display the abrupt "spiking" and oscillating behavior of the baseline approach, as seen in Figure 4. During the most extreme maneuvers, the highest rate of change for the MPC-DNN and PID elevator and rudder actuator speeds was 13 and 23 degrees per second, respectively.

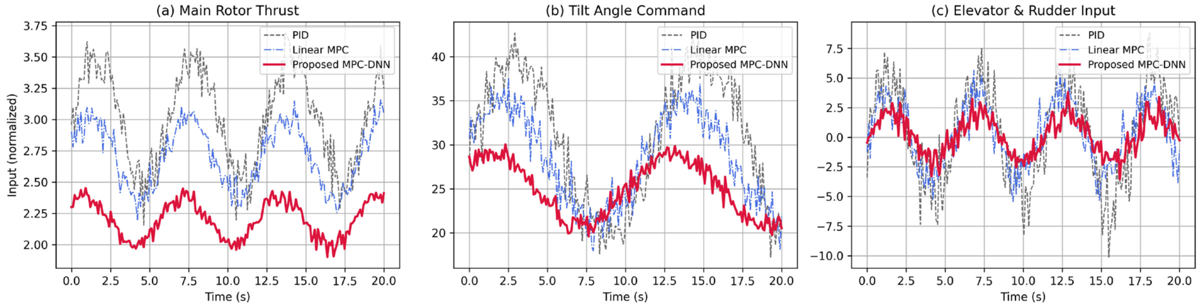

Figure 4. Control Input Variation During Transition. (a) Main rotor thrust command. (b) Tilt angle command. (c) Elevator and rudder actuation.

The next gadget is likewise very energy-efficient. Throughout the whole mission, MPC-DNN's average total control-related energy usage was 251W. At 265 Wh, classical MPC was comparatively energy-efficient, while PID controllers required roughly 8 Wh more power at 273 Wh. Because the baseline controller needed a stronger but less effective actuation, the savings are comparatively greater in the gust recovery and high-frequency lateral tracking phases.

Evaluate the response's delay more accurately. After a disturbance or important command, the MPC-DNN technique often reduced the system error to 5% of the reference path in 1.1 seconds. The first MPC approach took about 1.5 seconds, while PID controllers took almost two seconds and responded slowly under a high wind gust. Theresults collectively demonstrate that the MPC-DNN architecture has greatly raised average accuracy, better actuation smoothness, improved robustness to environmental disturbances, and decreased energy consumption; as a result, it satisfies the demanding criteria of aerial robots. The reliability and consistency of the aforementioned outstanding outcomes under more varied working settings are demonstrated by the following statistics.

Statistical Significance and Discussion

Gather the outcomes of every simulated trial conducted in various settings to assess the new control method's stability and capacity for generalization. We examined the consistency of the observed performance differences in the mean and under significant variations in injected actuator faults, sensor noise, and wind speed.

In Figure 5, box plots and variance indicators were used to describe the statistics of the tracking errors resulting from the aforementioned control measures. The tracking inaccuracy for MPC-DNN had a median of about 0.26 meters and an interquartile range of just 0.13 meters. PID's upper quartile exceeded 0.72 meters, and its dispersion was even broader than that of linear MPC, which had a bigger median error of 0.41 metres. The MPC-DNN approach never went over 0.57 meters with extreme turbulence or simulated actuator deterioration, however the greatest error in PID control under severe disturbance was over 1.2 meters.

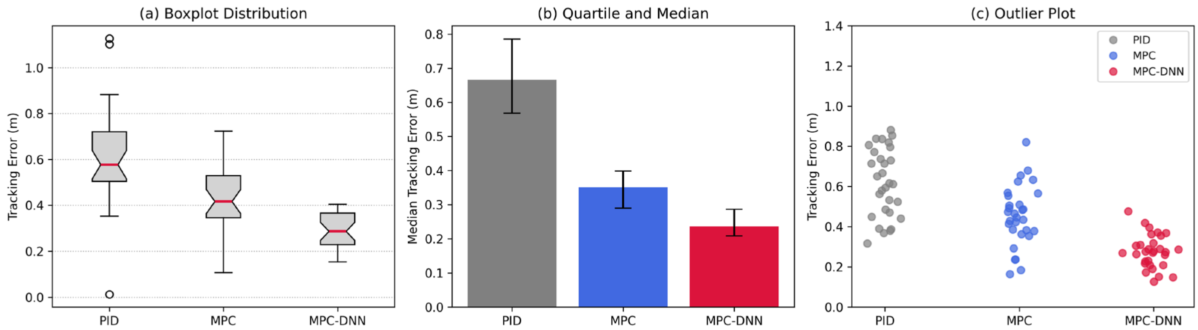

Figure 5. Statistical Analysis of Tracking Performance. (a) Distribution of trajectory tracking errors over 30 independent runs for each controller. Subplots reveal (b) median and quartile boundaries, (c) outlier ranges.

Robustness to disturbance was further investigated by grouping experimental runs according to wind and turbulence intensity, actuator saturation incidents, and sensor degradation simulations. Figure 6 presents a comparison of tracking deviation under moderate (5 m/s) and severe (14 m/s) gusts. While all controllers experience increased error under harsher wind, the relative increase for MPC-DNN is only 42%, compared to 68%

for classical MPC and 115% for PID. In the presence of temporary actuator saturation, MPC-DNN's error momentarily peaked at 0.72 meters before rapidly recovering, whereas PID lost effective trajectory control, showing persistent deviations above 1.3 meters and even temporary divergence from the reference path.

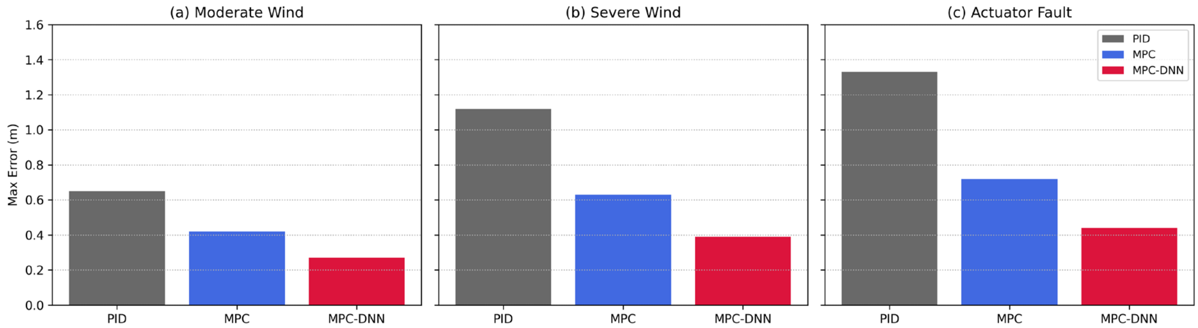

Figure 6. Robustness to Disturbances in Various Scenarios. (a) Tracking error under moderate and severe wind for all controllers. (b) Impact of actuator saturation. (c) Effect of sensor noise.

In assessing the spatial continuity and quality of the full three-dimensional flight path, Figure 7 overlays example trajectory traces from all controllers against the reference path during a test series blending vertical climb, rapid mode transition, and high-speed lateral maneuvering. The MPC-DNN trajectory closely tracks the reference curve with smooth transitions and no abrupt deviations. Both standard MPC and PID exhibit noticeably more path waviness and short-term divergence, particularly during high-rate maneuvers. Integrated spatial error, defined as the cumulative Euclidean distance between actual and reference trajectories, averaged only 21 meters for MPC-DNN across the 420-meter test path, compared to 34 meters for MPC and 57 meters for PID.

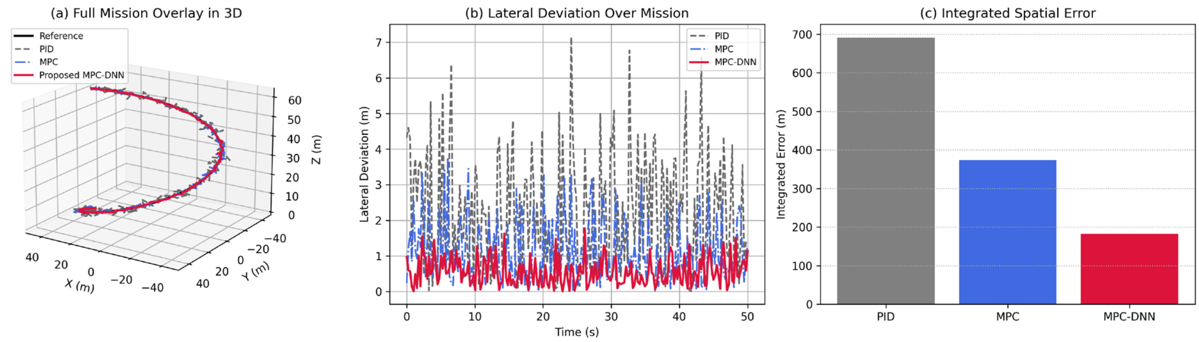

Figure 7. 3D Flight Path Comparison With Baseline Methods. (a) Full mission overlay in 3D. (b) Lateral deviation detail in transition phase. (c) Spatial error bands.

The aforementioned study leads to the conclusion that the neural network module's adaptive, closed-loop compensation has lessened the impact of stochastic disturbances and model uncertainty, bringing the baseline optimal control performance closer to the theoretical optimum. Reduced tracking error variance and a stable, quick system reaction to failures, actuator nonlinearities, and sensor degradation without re-tuning or loss of constraint enforcement are indicators of enhanced resilience.

According to the findings of the aforementioned discussions, under limitations and monitoring in each control cycle, adaptive nonlinear compensation by DNN can achieve significant gains over both model-based and conventional feedback approaches. To maintain uniform control authority over the whole flight envelope, MPC-DNN updates the online model on a regular basis. There is no instability or safety-constraint violation during extreme tests, such as abrupt actuator performance decreases or quick full-authority tilting in turbulent air; instead, the performance deteriorates smoothly.

Although it performs well overall, the MPC-DNN framework is not flawless. The DNN correction will initially lag and produce a brief transient until on-policy adaptation restarts in rare instances of entirely unexpected sensor failure combinations or unmodeled dynamical factors (e.g., changes in airframe caused by icing). Although the computational cost is comparatively low for the investigated cases, it must be further minimized before operating on small-scale, resource-constrained devices. To further shorten the adaption time and increase resilience to extremely erratic failure or disturbance patterns, incremental learning and certified neural boundary techniques will be used in the future.

Conclusion

This work has developed a comprehensive optimization framework for tilt-rotor aircraft that combines learning-based neural compensation with model predictive control, and it has produced noteworthy results in both theory and experiment. The fundamental flaws of conventional model-based and simply data-driven controllers have been fixed by breaking down the trajectory tracking problem into a real-time limited MPC layer and an adaptive DNN-based dynamic model correction. The MPC-DNN architecture has been demonstrated to be more accurate, resilient, and input-smooth under all mission scenarios based on high-fidelity simulations that include intricate models of actuator dynamics and environmental disturbances. The hybrid approach can maintain relatively small mean and maximum trajectory errors, demonstrate more stable disturbance rejection in the presence of rapidly changing nonlinearities and strong wind gusts, and somewhat reduce control energy consumption even when actuator saturation occurs when compared to the conventional MPC and high-performance PID controllers.

This method's intrinsic value lies in its ability to meet stringent requirements and intelligently adjust to the plant's unmodeled dynamics and real-time disruption situations. A closed-loop, modular framework is used to both enable continual performance improvement through data-driven model optimization and guarantee flight safety through analytical verification. As a result, the suggested design has produced trajectory tracking that is both steady and very accurate. It is also rather easy to use and flexible, both of which are essential for unmanned aircraft applications in the real world.

In the future, MPC-DNN will be the focus of many research projects and engineering uses. Reduce the high-dimensional multirotor and unmanned helicopter system's approach speed, extend the adaptation process to deal with even more serious unmodeled errors, and convert the algorithms into embedded flight control hardware that operates in real time. Model-based optimal control and real-time learning have reached high levels of development and are currently setting the standard for next-generation aerial robots and intelligent transportation systems due to the growing demand for resilient and adaptive autonomous flights across all sectors of civil and industrial applications.

Author Contributions

Gabriela Górska contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Beata Cicha and Elżbieta Zosia Szymańska contribute to validation, analysis, investigation, data collection, draft preparation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Altan, A., & Hacıoğlu, R. (2020). Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. Mechanical Systems and Signal Processing, 138, 106548. https://doi.org/10.1016/j.ymssp.2019.106548

[2] Bouaiss, O., Mechgoug, R., Taleb-Ahmed, A., & Brikel, A. E. (2023). Adaptive neural network based compensation control of quadrotor for robust trajectory tracking. International Journal of Adaptive Control and Signal Processing, 37(10), 2772-2793. https://doi.org/10.1002/acs.3659

[3] Mohammadi, A., & Ramezani, A. (2022). A robust model predictive control-based method for fault detection and fault tolerant control of quadrotor UAV. Transactions of the Institute of Measurement and Control, 45(1), 37-48. https://doi.org/10.1177/01423312221107971

[4] Azid, S. I., Ali, S. A., Kumar, M., Cirrincione, M., & Fagiolini, A. (2023). Precise trajectory tracking of multi-rotor UAVs using wind disturbance rejection approach. IEEE access, 11, 91796-91806. https://doi.org/10.1109/ACCESS.2023.3308297

[5] Lu, G., Cai, Y., Chen, N., Kong, F., Ren, Y., & Zhang, F. (2023). Trajectory generation and tracking control for aggressive tail-sitter flights. The International Journal of Robotics Research, 43(3), 241-280. https://doi.org/10.1177/02783649231207655

[6] Gulzar, M. M., Sibtain, D., Ahmad, A., Javed, I., Murawwat, S., Rasool, I., & Hayat, A. (2022). An efficient design of adaptive model predictive controller for load frequency control in hybrid power system. International Transactions on Electrical Energy Systems, 2022(1), 7894264. https://doi.org/10.1155/2022/7894264

[7] Mitsioni, I., Tajvar, P., Kragic, D., Tumova, J., & Pek, C. (2023). Safe data-driven model predictive control of systems with complex dynamics. IEEE Transactions on Robotics, 39(4), 3242-3258. https://doi.org/10.1109/TRO.2023.3266995

[8] Zhao, J., Lu, P., Du, C., & Cao, F. (2023). Active fault-tolerant strategy for flight vehicles: Transfer learning-based fault diagnosis and fixed-time fault-tolerant control. IEEE Transactions on Aerospace and Electronic Systems, 60(1), 1047-1059. https://doi.org/10.1109/TAES.2023.3333763

[9] Rohr, D., Studiger, M., Stastny, T., Lawrance, N. R. J., & Siegwart, R. (2021). Nonlinear model predictive velocity control of a VTOL tiltwing UAV. IEEE Robotics and Automation Letters, 6(3), 5776-5783. https://doi.org/10.1109/LRA.2021.3084888

[10] Zhuang, H. X. (2023). Adaptive sliding mode attitude control of quaternion model for aircraft based on neural network minimum parameter learning method. The Aeronautical Journal, 128(1321), 559-576. https://doi.org/10.1017/aer.2023.53

[11] Zhang, X., Liu, J., Xu, X., Yu, S., & Chen, H. (2022). Robust learning-based predictive control for discrete-time nonlinear systems with unknown dynamics and state constraints. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(12), 7314-7327. https://doi.org/10.1109/TSMC.2022.3146284

[12] Sferrazza, C., Muehlebach, M., & D'Andrea, R. (2020). Learning-based parametrized model predictive control for trajectory tracking. Optimal Control Applications and Methods, 41(6), 2225-2249. https://doi.org/10.1002/oca.2656

[13] Wang, J., Yuan, X., & Zhu, B. (2022). Geometric control for trajectory-tracking of a quadrotor UAV with suspended load. IET Control Theory & Applications, 16(12), 1271-1281. https://doi.org/10.1049/cth2.12301

[14] Shen, S., Xu, J., Chen, P., & Xia, Q. (2023). Adaptive neural network extended state observer-based finite-time convergent sliding mode control for a quad tiltrotor UAV. IEEE Transactions on Aerospace and Electronic Systems, 59(5), 6360-6373. https://doi.org/10.1109/TAES.2023.3274733

[15] Zhang, Q., Wang, T., & Zhu, X. (2023). Fixed-time extended disturbance observer-based switching control of quadrotor UAV under marine wind condition. International Journal of Dynamics and Control, 12(5), 1491-1501. https://doi.org/10.1007/s40435-023-01295-2

[16] Chen, Q., Jin, Y., Wang, T., Wang, Y., Yan, T., & Long, Y. (2022). UAV formation control under communication constraints based on distributed model predictive control. IEEE Access, 10, 126494-126507. https://doi.org/10.1109/ACCESS.2022.3225434

[17] Taimoor, M., Lu, X., Maqsood, H., & Sheng, C. (2021). Adaptive rapid neural observer-based sensors fault diagnosis and reconstruction of quadrotor unmanned aerial vehicle. Aircraft Engineering and Aerospace Technology, 93(5), 847-861. https://doi.org/10.1108/AEAT-01-2021-0005

[18] Kiumarsi, B., Vamvoudakis, K. G., Modares, H., & Lewis, F. L. (2017). Optimal and autonomous control using reinforcement learning: A survey. IEEE transactions on neural networks and learning systems, 29(6), 2042-2062. https://doi.org/10.1109/TNNLS.2017.2773458

[19] Ge, J., & Liu, L. (2023). Hierarchical robust model prediction control for a long-endurance unmanned aerial vehicle. Journal of Guidance, Control, and Dynamics, 46(6), 1176-1183. https://doi.org/10.2514/1.G007180

[20] Bingol, O., & Guzey, H. M. (2023). Fixed-time neuro-sliding mode controller design for quadrotor UAV transporting a suspended payload. European Journal of Control, 73, 100879. https://doi.org/10.1016/j.ejcon.2023.100879

[21] Tripicchio, P., Unetti, M., D'Avella, S., & Avizzano, C. A. (2023). Smooth coverage path planning for UAVs with model predictive control trajectory tracking. Electronics, 12(10), 2310. https://doi.org/10.3390/electronics12102310

[22] Anh, H. P. H., & Dat, N. T. (2023). Robust adaptive control of nonlinear dynamic systems using hybrid sliding mode regressive neural learning technique. Engineering Computations, 40(3), 657-678. https://doi.org/10.1108/EC-06-2022-0399