https://doi.org/10.31891/csit-2026-1-19

UDC 004.8:004.94:007.52:629.7.05

# HYBRID METHOD OF ADAPTIVE CONTROL OF VARIABLE MODE OF UNMANNED AERIAL VEHICLES WITH INTELLIGENT ONLINE COMPENSATION OF DISTURBANCES

**Stepan TANASIICHUK**
PhD Student of Computer Engineering & Information Systems Department,
Khmelnytskyi National University,
Khmelnytskyi, Ukraine
https://orcid.org/0009-0002-2657-6190
e-mail: tanas@i.ua

Received: 20/02/2026
Accepted:20/03/2026
Published:26/03/2026



*The article resolves the current scientific and technical contradiction between the need to increase the accuracy of navigation control of autonomous unmanned aerial vehicles (UAVs) and the strict resource constraints of on-board computing systems. An intelligent-robust control architecture is proposed, based on the synthesis of the adaptive alternating mode method (ASMC) and the recurrent neuro-fuzzy network RSEFNN. The scientific novelty of the work lies in the improvement of the hybrid approach, which, unlike classical robust methods, uses an intelligent observer for online identification and compensation of nonlinear components of dynamics and external disturbances. This made it possible to significantly reduce the gain coefficients of the discontinuous part of the controller, minimize the "rattling" effect, and increase the energy efficiency of actuators. Mathematical proof of the stability of the closed-loop system using the direct Lyapunov method confirmed the asymptotic convergence of trajectory tracking errors to zero and guaranteed the numerical stability of the neural network training processes. An important practical contribution is the implementation of methods for suppressing high-frequency oscillations by replacing the discontinuous control function with its smooth approximation based on the adaptive boundary layer and hyperbolic tangent. To ensure the determinism of the computational cycle in real time, optimization using the Padé method was applied, which allowed minimizing algorithmic latency and achieving a control frequency of up to 1000 Hz on embedded CPUs without specialized accelerators. The results of the comparative analysis confirmed the high robustness of the developed method under conditions of intense wind loads. In particular, the use of the ASMC+RSEFNN controller allowed to increase the positioning accuracy in steady state by 10.2–12.6 times compared to classical PID controllers. The integrated neuro-fuzzy identifier provided effective compensation for systematic wind shear, which is a critical factor for performing UAV precision guidance tasks in difficult meteorological conditions.*

*Keywords: autonomous unmanned aerial vehicles (UAVs), adaptive switching mode control (ASMC), recurrent neural fuzzy network (RSEFNN), control system robustness, precision homing, online disturbance identification, Lyapunov method, chattering effect, embedded real-time systems, SWaP constraints, algorithmic latency, Padé method, multisensor data fusion, control invariance.*

## Introduction

Currently, the scope of application of unmanned aerial vehicles (UAVs) is constantly expanding both in the civilian sector of the economy and in the defense complex, where they provide the implementation of critical missions in remote sensing, high-precision orientation in space and independent transportation of objects. The latest autonomous UAV is considered as an integrated cyber-physical structure, the functioning of which is limited by the performance of on-board computing resources; in such a system, data processing processes, network interaction and aerodynamic dynamics are combined into a single closed loop operating in real time [1].

Current UAV control complexes are strategically important embedded real-time computing nodes. In them, the level of algorithm complexity directly correlates with the appearance of time lags, the occurrence of phase shifts and a decrease in the overall robustness of the control loop [2, 3]. Compared to traditional means of automation, the operation of UAVs occurs under the simultaneous influence of nonlinear aerodynamic factors, random external interference, as well as strict limits on the dimensions, mass and power consumption of on-board devices (SWaP parameters) [4–6]. These factors dictate the need for a comprehensive study of the relationship between the architectural solutions of the hardware, control algorithms and software. The growth of demands for the precision of maneuvering of UAVs under the pressure of variable external factors and structural uncertainties of the object determines the relevance of creating progressive methods of automated control with high computational efficiency.

Analysis of existing control methods revealed a number of shortcomings [7-9]:

1) classical linear controllers exhibit a low level of invariance and do not provide resistance to structural changes and external disturbances;

2) robust methods (in particular, sliding modes) demonstrate high stability, but create a “rattling” effect, which negatively affects actuators and data buses;

3) intelligent approaches based on deep learning are too resource-intensive for embedded computers.

The results of the review of current methods and tools for controlling autonomous unmanned aerial vehicles indicate that existing approaches do not allow simultaneously achieving high accuracy of navigation control, resistance to external influences and clear determinism of computational processes under conditions of shortage of on-board equipment resources [10, 11]. In particular, traditional linear controllers exhibit a low level of invariance with respect to parametric and structural uncertainties of the object, while the latest adaptive and intelligent algorithms often lead to increased computational costs, increased latency, and unpredictability of the duration of operations [12, 13].

These limiting factors become particularly acute at the stage of precision UAV guidance, when the control complex must operate in an environment with noisy or fragmented navigation information, under the influence of non-stationary aerodynamic perturbations, and when changing the dynamic characteristics of the device itself, while ensuring stable stability and a given localization accuracy [14-16]. The study of multisensor data fusion and navigation computing architectures also confirmed that time delays caused by information processing and synchronization of heterogeneous sensor channels have a destructive effect on the quality of regulation and can cause the loss of robust characteristics of the closed-loop control system [17, 18].

Therefore, there is a contradiction between the need to increase the accuracy of precision UAV guidance in noisy data conditions and limited hardware resources. The solution to this problem is to create an intelligently robust control architecture. The aim of the study is to develop computationally efficient methods based on self-evolving structures that will ensure stable system operation under latency and dynamic uncertainties.

**Synthesis of a hybrid method for adaptive variable-mode control of UAVs with intelligent online disturbance compensation**

The UAV control process is aimed at minimizing the discrepancy between the given trajectory $\xi_d(t)$ and the real state of the device $\xi(t)$. Let’s define the tracking error vector as:

$$e(t) = \xi_d(t) - \xi(t). \tag{1}$$

For the UAV computing system, this vector consists of six independent components:

$$e = \left[e_x, e_y, e_z, e_\phi, e_\theta, e_\psi\right]^T.$$

The error rate $\dot{e}(t)$ is defined as the difference between the desired speed and the current speed obtained from the Sensor Fusion module:

$$\dot{e}(t) = \dot{\xi}_d(t) - \dot{\xi}(t). \tag{2}$$

From the standpoint of computer engineering, the calculation of $\dot{e}$ in embedded systems is a noisy process. Since the differentiation of input signals amplifies high-frequency sensor noise, the software implementation of the ASMC controller uses a finite-difference algorithm with pre-filtering of the signal by a low-pass filter (LPF) or a digital state observer.

The task of synthesizing the control law is to determine such a vector of generalized forces τ that would guarantee the convergence of the system state to the sliding surface $s = 0$ and its retention on it regardless of external influences. The structure of the proposed hybrid control law is based on a combination of an equivalent component that compensates for deterministic dynamics, an adaptive discontinuous component, and an intelligent disturbance estimator.

Let’s present the complete mathematical model of the UAV in a compact matrix form, convenient for software implementation and synthesis of adaptive control:

$$\begin{cases} \ddot{\eta} = \dfrac{1}{m}(R(\Theta)F_B + F_G + F_D) \\ \dot{\Omega} = I^{-1}(\Gamma_B - \Omega \times (I\Omega) + \Gamma_D) \end{cases} \tag{3}$$

For the purposes of synthesis of adaptive control and software implementation of algorithms in real time, it is advisable to present the system of differential equations (3) in the general vector-matrix form adopted in the theory of robotic systems:

$$M(\xi)\ddot{\xi} + C\big(\xi, \dot{\xi}\big)\dot{\xi} + G(\xi) + D\big(\dot{\xi}\big) = \tau + \tau_d, \tag{4}$$

where $M(\xi) \in \mathbb{R}^{6\times6}$ – is the matrix of inertia and masses – it is positive definite and symmetric, for UAVs with GPP its structure takes into account not only the mass of the hull, but also the distribution of the moments of inertia of the additional engines; $C(\xi,\dot{\xi}) \in \mathbb{R}^{6\times6}$ – is the matrix of Coriolis and centrifugal forces - these forces describe the dynamic relationships between different degrees of freedom that arise during rapid maneuvering (for example, the effect of yaw rate on roll); $G(\xi) \in \mathbb{R}^{6\times1}$ is the vector of gravitational forces; $D(\dot{\xi}) \in \mathbb{R}^{6\times1}$ is the vector of aerodynamic drag - since the homing stage occurs in wind conditions, this term is described as the sum of linear and quadratic air resistance: $d_{linear}\dot{\xi} + d_{quad}\dot{\xi}^2$; $\tau \in \mathbb{R}^{6\times1}$ is the vector of generalized control forces and moments (engines generate thrust, which is then converted into the vector $\tau$ through the matrix B); $\tau_d$ is the vector of unknown external disturbances, which includes turbulence and model approximation errors.

*Derivation of equivalent control.* To find the structure of the equivalent control, we consider the condition $\dot{s} = 0$:

$$\dot{s} = \ddot{e} + \dot{\Lambda e} = 0, \quad (5)$$

which, taking into account (1) and (2), gives an expression for the desired acceleration: $\ddot{\xi} = \ddot{\xi}_d + \Lambda\dot{e}$. Next, substituting the obtained expression into the matrix model of UAV dynamics (4), we define the complete equivalent control structure $\tau_{eq}$, which provides compensation for the nominal dynamics of the device:

$$\tau_{eq} = M(\xi)\left(\ddot{\xi}_d + \Lambda\dot{e}\right) + C(\xi,\dot{\xi})\dot{\xi} + G(\xi) + D(\dot{\xi}). \quad (6)$$

Direct calculation of expression (6) in the on-board software is complicated by high algorithmic complexity. The matrices $C(\xi,\dot{\xi})$ and $D(\dot{\xi})$ contain dozens of trigonometric and nonlinear dependencies, which leads to significant processor time consumption and increased computational latency.

In order to optimize the resources of the on-board computer, this work proposes to use a simplified form of equivalent control, where the calculation of complex components *C* and *D* is replaced by their online identification using the recurrent neural network RSEFNN. This approach allows to radically reduce the number of multiplication operations in the main control cycle, transferring the load to the adaptive intelligent block.

*Integration of the intelligent estimator RSEFNN.* To compensate for unknown external disturbances $\tau_d$ (wind, turbulence), an estimate $\hat{\tau}_d$, is introduced into the control loop, which is generated by the recurrent neural network RSEFNN. The full control law takes the form:

$$\tau = \tau_{eq} + \tau_{sw} - \hat{\tau}_d. \quad (7)$$

The use of neural network estimation $\hat{\tau}_d$ allows to “unload” the discontinuous component $\tau_{sw}$, since the main part of the disturbance energy is compensated by the predictive signal of the network. This is a key factor in increasing the accuracy at the refinement stage.

Unlike classical robust methods, where high gain coefficients are required to suppress the wind, the proposed hybrid approach is based on online compensation of disturbances by a neural network. This allows to significantly reduce the amplitude of the discontinuous control, ensuring minimization of the “rattling” effect and increasing the energy efficiency of the actuators while maintaining the mathematically proven global asymptotic stability.

*Adaptive robust control.* The discontinuous component $\tau_{sw}$ (Switching control) is designed to suppress the residual error of the neural network approximation. In this work, it is proposed to use an adaptive gain coefficient $\hat{K}(t)$:

$$\tau_{sw} = \hat{K}(t) \cdot sgn(s). \quad (8)$$

To ensure numerical stability and prevent unlimited growth of the coefficient ("wind-up" effect) in software implementation, the adaptation law with a modified structure is used:

$$\dot{\hat{K}} = \gamma|s| - \sigma\hat{K}. \quad (9)$$

where $\gamma$ – is the learning rate, and $\sigma$ is the "leakage" coefficient (leakage term), which guarantees the boundedness of $\hat{K}$ during numerical integration in real time.

*Computational structure of the algorithm.* The proposed control law (6) is a sequential computational pipeline implemented in the on-board software:

1. Error identification block: Calculation of $e, \dot{e}$ and formation of the vector $s$.
2. Neural network output block: Calculation of $\hat{\tau}_d$ using the current RSEFNN weights.
3. Coefficient adaptation block: Updating $\hat{K}$ through integration by the Euler method.
4. Control synthesis block: Summing the components (6) and outputting the result to the actuation matrix $B$.

Such a structure ensures determinism of calculations: the control cycle execution time remains constant, which allows the system to operate at a frequency of 500 Hz and higher, ensuring robust stability of the UAV even in the event of sudden sensor failures or extreme wind gusts.

### Mathematical proof of the stability of a closed-loop system using the Lyapunov method

To guarantee the reliability of the autonomous UAV operation at the homing stage, it is necessary to mathematically confirm that the proposed hybrid architecture (ASMC controller and RSEFNN neural network observer) provides asymptotic stability of tracking errors. The main analysis tool is the direct Lyapunov method, which allows us to prove the convergence of the system states to the sliding surface under the action of non-stationary disturbances.

The proof of the stability of the proposed hybrid control system is based on the second (direct) Lyapunov method. The key design stage is the construction of a scalar positive definite function $V(t)$, which acts as a generalized "loss function" (Cost Function). Its minimization during the operation of the UAV computing system guarantees not only the physical stability of the flight, but also the numerical convergence of the online training processes of the RSEFNN neural network.

For a comprehensive study of the hybrid circuit "ASMC + RSEFNN", a combined Lyapunov function of quadratic form was chosen, which takes into account the dynamics of the sliding error, the state of the parametric adaptation of the controller and the learning energy of the neural network observer:

$$V = \frac{1}{2}s^T M(\xi)s + \frac{1}{2\gamma}\tilde{K}^2 + \frac{1}{2}tr\left(\tilde{W}^T\Gamma^{-1}\tilde{W}\right). \quad (10)$$

To conduct a correct analysis, it is necessary to detail the physical and mathematical content of the components that form this generalized function. Each term of equation (10) is responsible for the stability of a separate segment of the adaptive computational cycle:

1. The energy of the slip error ($\frac{1}{2}s^T M(\xi)s$). This term characterizes the energy of the UAV deviation from the desired trajectory:

– $s = \left[s_x, s_y, s_z, s_\phi, s_\theta, s_\psi\right]^T \in \mathbb{R}^{6\times1}$ is a six-dimensional vector of slip surfaces, defined in formula (5), which integrates static ($e$) and dynamic ($\dot{e}$) state errors.

– $M(\xi) \in \mathbb{R}^{6\times6}$ is a symmetric positive definite matrix of inertia and mass of the device.

From the standpoint of computer engineering, minimizing this term means forcing the computational model to return to a deterministic working zone, where the positioning error asymptotically approaches zero.

2. Robust gain adaptation error energy ($\frac{1}{2\gamma}\tilde{K}^2$). This component is responsible for the convergence of the wind suppression effort selection algorithm. Since the upper limit of external disturbances is not known to the UAV software a priori, the system must independently adapt the level of its "aggressiveness":

– $\hat{K}(t)$ is the current value of the gain coefficient, which is calculated in the on-board software.

– $K_{max}$ is the theoretical limit of disturbances necessary for complete invariance of the system.

– $\tilde{K} = K_{max} - \hat{K}$ is the parametric adaptation error, which reflects the discrepancy between the current level of robustness and the required one.

– $\gamma > 0$ – adaptation rate coefficient, which in the software implementation determines the step of discrete integration of the adaptive law.

3. Neural network training energy ($\frac{1}{2}tr\left(\tilde{W}^T\Gamma^{-1}\tilde{W}\right)$). This term characterizes the quality of identification of nonlinearities of dynamics by the intelligent block. Using the matrix trace operator ($tr$ – the sum of diagonal elements) allows to obtain a scalar estimate of the "training energy" of the entire neural network weight table:

– $\hat{W} \in \mathbb{R}^{n\times m}$ – matrix of current weight coefficients of the RSEFNN neural network stored in the controller's RAM.

– $W^*$ – matrix of optimal weights, which provides ideal approximation of disturbances according to the properties of fuzzy systems.

– $\tilde{W} = W^* - \hat{W}$ – neural network training error, the minimization of which guarantees high accuracy of wind compensation.

– $\Gamma = diag(\Gamma_1, \ldots, \Gamma_n)$ is a positive definite learning rate matrix that regulates the rate of weight updates in the gradient algorithm.

It is important to note that when analyzing the dynamics of the selected function $V(t)$ the parameter $\epsilon$ is taken into account – the residual error of the fuzzy approximation. Although $\epsilon$ is not a state variable and is not directly included in expression (10), this parameter determines the accuracy limit of the RSEFNN neural network. Taking $\epsilon$ in further calculations allows us to prove that the proposed adaptive control law ASMC has a sufficient margin of robustness to compensate for the inaccuracies of the intelligent circuit. Therefore, when deploying the dynamics of the system, the accuracy limit of the neural network structure is taken into account ($\epsilon \in \mathbb{R}^{6\times1}$ – the vector of the residual approximation error, which arises due to the limited number of neural network rules and the discrete nature of the calculations):

$$\epsilon = d(t) - \hat{d}(W^*, \Psi), \quad (11)$$

where $d(t)$ is the real disturbance; $\hat{d}(\cdot)$ is the output of the ideal network; $\Psi(s) \in \mathbb{R}^{m\times1}$ – is a vector of neural network activation functions, the components of which $\psi_i$ reflect the degree of truth (activation) of the $i$-th fuzzy rule for the current state of the sliding surface. From the standpoint of the computing system, this vector is the result of the work of the fuzzification and aggregation layers of the RSEFNN network rules.

The introduction of the parameter $\epsilon$ allows us to take into account the physical limitations of on-board computers and to bring the robustness of the system to the errors of modeling AI structures.

To ensure stability, the ASMC algorithm is designed so that the robust component K ̂ always overlaps the value $|\epsilon|$. By its structure, the function $V(t)$ satisfies the requirements of the Lyapunov theorem: it is positive definite ($V > 0$ at $s \neq 0, \tilde{K} \neq 0, \tilde{W} \neq 0$) and radially unbounded. From the standpoint of computer implementation, this means that the algorithm is internally protected from overflow: if the energy $V$ decreases, then all internal variables (network weights and adaptation coefficients) are guaranteed to remain within the permissible memory ranges.

Thus, the chosen Lyapunov function is an integral criterion for the quality of the functioning of the UAV digital twin.

To determine the stability of the proposed algorithm, it is necessary to analyze the rate of change of the selected energy function $V(t)$ in time. According to the direct Lyapunov method, to ensure asymptotic stability, the first derivative of the function with respect to time must be negative definite.

Let's differentiate the function (10) with respect to time:

$$\dot{V} = s^T M(\xi)\dot{s} + \frac{1}{2}s^T \dot{M}(\xi)s - \frac{1}{\gamma}\tilde{K}\dot{\hat{K}} - tr\left(\tilde{W}^T \Gamma^{-1}\dot{\hat{W}}\right). \quad (12)$$

Here we have taken into account that the derivative of the adaptation error $\dot{\tilde{K}} = \frac{d}{dt}\left(K_{max} - \hat{K}\right) = -\dot{\hat{K}}$, similarly for the weights of the neural network $\dot{\tilde{W}} = -\dot{\hat{W}}$.

*Unfolding the dynamics of the sliding surface.* To ensure the asymptotic stability of the tracking error, we introduce a linear first-order slip surface $s(t)$ in the following form:

$$s = \dot{e} + \Lambda e, \quad (13)$$

where $\Lambda = diag(\lambda_1, \dots, \lambda_6)$ is a diagonal matrix of positive definite gain coefficients ($\lambda_i > 0$).

Using the definition of the slip surface (5) and its derivative $\dot{s} = \ddot{e} + \Lambda\dot{e} = \left(\ddot{\xi}_d - \ddot{\xi}\right) + \Lambda\dot{e}$, let's express the product $M\dot{s}$ in terms of the dynamic parameters of the UAV. Substituting the matrix equation of dynamics (4) into the expression for $\ddot{\xi}$, we obtain:

$$M\dot{s} = M\left(\ddot{\xi}_d + \Lambda\dot{e}\right) - \left(\tau + \tau_d - C\left(\xi, \dot{\xi}\right)\dot{\xi} - G(\xi) - D\left(\dot{\xi}\right)\right), \quad (14)$$

Substitute the obtained expression (14) into the general equation of the Lyapunov derivative (12):

$$\dot{V} = s^T\left[M\left(\ddot{\xi}_d + \Lambda\dot{e}\right) + C\dot{\xi} + G + D - \tau - \tau_d\right] + \frac{1}{2}s^T\dot{M}s - \frac{1}{\gamma}\tilde{K}\dot{\hat{K}} - tr\left(\tilde{W}^T\Gamma^{-1}\dot{\hat{W}}\right). \quad (15)$$

*Using the property of skew symmetry.* One of the fundamental properties of the dynamics of robotic systems, to which UAVs belong, is that the matrix $\left(\dot{M} - 2C\right)$ is skew symmetric. This leads to the following equality, which is important for computational stability:

$$s^T\left(\dot{M} - 2C\right)s = 0 \Longrightarrow \frac{1}{2}s^T\dot{M}s = s^T C s \quad (16)$$

Using this property allows to simplify expression (15) by replacing the term with the derivative of the inertia matrix with the term with the Coriolis force matrix. This is critically important for software implementation, since the derivative $\dot{M}$ is extremely difficult to calculate in real time, while the matrix C is already present in the model.

After transformations we obtain:

$$\dot{V} = s^T\left[M\left(\ddot{\xi}_d + \Lambda\dot{e}\right) + C\left(\xi, \dot{\xi}\right)\dot{\xi} + G + D - \tau - \tau_d\right] + s^T C\left(\xi, \dot{\xi}\right)s - \frac{1}{\gamma}\tilde{K}\dot{\hat{K}} - tr\left(\tilde{W}^T\Gamma^{-1}\dot{\hat{W}}\right) \quad (17)$$

*Substitution of the hybrid control law.* Let's substitute the structure of the developed control law $\tau = \tau_{eq} + \tau_{sw} - \hat{\tau}_d$ from formula (7) into equation (17). Considering that the equivalent control $\tau_{eq}$ (6) by its construction completely compensates the nominal (known) components *M,C,G,D*, the derivative equation takes the form:

$$\dot{V} = s^T[\hat{\tau}_d - \tau_d - \tau_{sw}] - \frac{1}{\gamma}\tilde{K}\dot{\hat{K}} - tr\left(\tilde{W}^T\Gamma^{-1}\dot{\hat{W}}\right), \quad (18)$$

where the difference $(\hat{\tau}_d - \tau_d)$ reflects the error in identifying disturbances by the neural network, and $\tau_{sw}$ is the robust effort that should suppress this error.

From the standpoint of computer engineering, the obtained expression (18) shows that the stability of the system now depends exclusively on the quality of the neural network training algorithm and the speed of adaptation of the robust coefficient. This confirms the validity of the hierarchical structure of the software, where the low-level controller compensates for dynamic residuals that the neural network did not have time to identify.

To complete the stability analysis, it is necessary to investigate the dynamics of the training error of the RSEFNN neural network and its interaction with the main control loop.

*Mathematical representation of the disturbance identification error.* According to the properties of fuzzy systems, the real external disturbance $\tau_d$, acting on the UAV, can be represented through an ideal neural network with optimal weights $W^*$:

$$\tau_d = W^{*T}\Psi(s) + \epsilon, \quad (19)$$

where $\Psi(s)$ is the vector of neural network activation functions (firing strengths), and $\epsilon$ is the residual approximation error due to the finite number of network rules.

The current estimate of the disturbance $\hat{\tau}_d$ generated by the RSEFNN computing unit has the form:

$$\hat{\tau}_d = \hat{W}^T\Psi(s), \quad (20)$$

where $\hat{W}$ is the current weight matrix stored in the controller RAM.

Let's determine the difference between the estimate and the real disturbance, which was included in the equation of the Lyapunov derivative (18):

$$\hat{\tau}_d - \tau_d = \hat{W}^T\Psi(s) - (W^{*T}\Psi(s) + \epsilon) = \left(\hat{W} - W^*\right)^T\Psi(s) - \epsilon = -\tilde{W}^T\Psi(s) - \epsilon, \quad (21)$$

where $\widetilde{W} = W^* - \widehat{W}$ is the learning error (weight mismatch).

*Compensation of learning energy in the Lyapunov derivative.* Let's substitute the obtained error expression (21) into the equation of the Lyapunov derivative (18):

$$\dot{V} = s^T\left[-\widetilde{W}^T\Psi(s) - \epsilon - \tau_{sw}\right] - \frac{1}{\gamma}\widetilde{K}\dot{\widehat{K}} - tr\left(\widetilde{W}^T\Gamma^{-1}\dot{\widehat{W}}\right). \quad (22)$$

Open the brackets for the first term. Using the linearity property of the trace operator of the matrix ($a^TBc = tr(Bca^T)$), we transform the scalar product $s^T\widetilde{W}^T\Psi(s)$ into matrix form:

$$\dot{V} = -tr\left(\widetilde{W}^T\Psi(s)s^T\right) - s^T\epsilon - s^T\tau_{sw} - \frac{1}{\gamma}\widetilde{K}\dot{\widehat{K}} - tr\left(\widetilde{W}^T\Gamma^{-1}\dot{\widehat{W}}\right). \quad (23)$$

Grouping the terms containing the weight error $\widetilde{W}^T$, we obtain:

$$\dot{V} = -tr\left(\widetilde{W}^T\left[\Psi(s)s^T + \Gamma^{-1}\dot{\widehat{W}}\right]\right) - s^T\epsilon - s^T\tau_{sw} - \frac{1}{\gamma}\widetilde{K}\dot{\widehat{K}}. \quad (24)$$

*Synthesis of the neural network learning law.* From the standpoint of computer engineering, the most rational is such a learning law that would zero the first term of equation (24), thereby guaranteeing that changing the AI weights always leads to a decrease in the total energy of the system error.

For this, the expression in square brackets must be equal to zero:

$$\Psi(s)s^T + \Gamma^{-1}\dot{\widehat{W}} = 0 \Rightarrow \dot{\widehat{W}} = -\Gamma\Psi(s)s^T. \quad (25)$$

Equation (25) is the fundamental law of online training of the RSEFNN neural network. Its features for implementation in embedded software:

1. Algorithmic simplicity: Updating the weights requires only the operation of multiplying the vector by the vector-transpose, which has a complexity of $O(n \cdot m)$. This allows training to be performed on each control cycle without overloading the processor.
2. No need for target sata: Unlike classical "supervised" learning, which requires reference disturbance data, this law uses only the slip error $s$. The network learns to "close" those errors that actually occur in flight.
3. Learning locality: Thanks to the vector $\Psi(s)$, only those fuzzy logic rules that are activated in the current state are updated, which increases the computational stability of the algorithm.

After applying the learning law (25), the derivative of the Lyapunov function is simplified to a form that depends only on the robust properties of the controller:

$$\dot{V} = -s^T\epsilon - s^T\tau_{sw} - \frac{1}{\gamma}\widetilde{K}\dot{\widehat{K}}. \quad (26)$$

This proves that the intelligent loop successfully identifies nonlinear dynamics, leaving only the task of suppressing the residual error $\epsilon$ for the adaptive ASMC controller.

After applying the neural network learning law (25), the derivative of the Lyapunov function was reduced to expression (26), which reflects the dynamics of the sliding error under the influence of the residual approximation error and the operation of the adaptive robust controller.

To complete the proof, we substitute the structure of the adaptive discontinuous component $\tau_{sw} = \widehat{K} \cdot sign(s)$ from formula (8). Taking into account the mathematical property $s^T \cdot sign(s) = |s|$, where $|s|$ is the sum of the absolute values of the vector components, we obtain:

$$\dot{V} = -s^T\epsilon - \widehat{K}|s| - \frac{1}{\gamma}\widetilde{K}\dot{\widehat{K}}. \quad (27)$$

Using the definition of the parametric adaptation error $\widetilde{K} = K_{max} - \widehat{K}$ and the proposed adaptive law of updating the coefficient $\dot{\widehat{K}} = \gamma|s|$ (9), we transform the last term:

$$\dot{V} = -s^T\epsilon - \widehat{K}|s| - \frac{1}{\gamma}\left(K_{max} - \widehat{K}\right) \cdot (\gamma|s|). \quad (28)$$

After algebraic simplification and mutual cancellation of the terms $\widehat{K}|s|$, the derivative of the energy function takes the form:

$$\dot{V} = -s^T\epsilon - K_{max}|s|. \quad (29)$$

*Lyapunov convergence analysis.* In order for the system to be asymptotically stable, it is necessary that $\dot{V}$ be strictly less than zero. Since $s^T\epsilon \le |s| \cdot |\epsilon|$, we can rewrite (29) as the inequality:

$$\dot{V} \le -|s|(K_{max} - |\epsilon|). \quad (30)$$

Considering that under the algorithm design conditions, the value of $K_{max}$ is chosen as the upper bound of the possible approximation error ($K_{max} > |\epsilon|$), the expression in parentheses will always be positive. Therefore, the derivative of the Lyapunov function is negative semidefinite ($\dot{V} \le 0$).

To prove global asymptotic stability (stability in large) it is necessary to take into account two factors:

1. Radial unboundedness: Since the function $V\left(s, \widetilde{K}, \widetilde{W}\right)$ chosen in (10) tends to infinity with unlimited growth of any of the errors, the system is stable in the entire state space.
2. Barbalata's lemma: Since $V(t)$ is bounded from below, and $\dot{V}(t) \le 0$ and is a uniformly continuous function (due to the boundedness of the UAV accelerations), then according to Barbalata's lemma $\dot{V}(t) \to 0$ at $t \to \infty$. This directly means that the sliding surface vector $s \to 0$.

The obtained result (30) is a theoretical guarantee that: the process of online training of the neural network will not lead to instability of the computational cycle; the homing error will be eliminated in a finite time, and the

UAV trajectory will remain on the surface s=0 even when the wind force changes; since V(t) does not increase, all internal algorithm variables (AI weights, adaptation coefficients) will remain within the allocated memory ranges, which prevents the occurrence of critical errors such as floating-point overflow.

Mathematically proven stability allows us to proceed to the next stage - optimizing the quality of control signals by suppressing high-frequency oscillations (rattling) that arise due to the presence of the sgn(s) function in the proven control law.

**Rattle suppression methods in variable-mode systems**

Despite the proven mathematical robustness, the practical implementation of adaptive variable mode control (ASMC) on UAV on-board computers faces the problem of high-frequency oscillations of the control signal, known as the "chattering" effect. From the standpoint of computer engineering, chattering is a destabilizing factor that causes excessive loading of ESC controllers, overheating of actuators and introduces significant digital noise into the feedback channels.

In theoretical calculations, the sign function $sgn(s)$ assumes an instantaneous switching when passing through the zero value of the sliding surface. However, in real computing systems, the control law is implemented with a fixed sampling step $\Delta t$ (time sampling). This leads to the fact that the state of the system cannot be exactly held on the surface $s = 0$, but constantly crosses it with high frequency, creating a limit cycle.

For the UAV computer system, this creates the following negative effects:

1. Information overload of the data bus: High-frequency changes in the command signal generate excessive traffic between the flight controller and the power subsystem.
2. Degradation of integration accuracy: Since numerical methods for solving differential equations (for example, Runge-Kutt) are based on the smoothness of functions, discontinuous control introduces computational artifacts that accumulate over time.
3. Excitation of parasitic resonances: Horizontal thrusters (HSTs), which have low inertia, instantly process these noises, which leads to intense body vibrations and noise in accelerometer readings.

To eliminate the discontinuous control in operation, it is proposed to replace the function sgn(s) with its continuous approximation within the adaptive boundary layer with a thickness of Φ.

The first stage of optimization is the use of the saturation function:

$$sat(s/\Phi) = \begin{cases} s/\Phi, |s| \leq \Phi \\ sgn(s), |s| > \Phi' \end{cases} \quad (31)$$

where $\Phi > 0$ is the parameter that determines the width of the linear zone near the sliding surface.

However, the saturation function has discontinuities of the first derivative at the layer boundaries, which is critical for the RSEFNN neural network training algorithm (which is based on gradient descent). To ensure the differentiability of the control signal, the thesis justifies the use of a smooth approximation based on the hyperbolic tangent:

$$\tau_{sw} = \hat{K}(t) \cdot tanh\left(\frac{s}{\Phi}\right). \quad (32)$$

From the standpoint of computer engineering, the *tanh* function ensures the smoothness of the control signal over the entire range of values.

Using a static value of Φ creates a computational contradiction: a large value of Φ effectively suppresses rattling, but increases the static positioning error; a small value increases the accuracy, but causes the return of oscillations.

To resolve this contradiction, the software implements an adaptive boundary layer algorithm, the thickness of which is a function of the state error:

$$\Phi(t) = \Phi_0 \cdot exp(-\alpha \|e\|), \quad (33)$$

where $\Phi_0$ is the initial value, $\alpha$ is the adaptation rate coefficient, $|e|$ is the trajectory tracking error rate.

Such logic allows the computing system to have a wide "protective" layer for significant deviations (ensuring smooth maneuvering) and automatically narrow it when the UAV approaches the target, which guarantees precise homing accuracy without the occurrence of parasitic oscillations.

To quantitatively assess the effectiveness of the methods for suppressing rattling in simulation experiments, the Total Variation (TV) metric was introduced, which describes the variability of the control signal:

$$\mathrm{TV}(u) = \sum_{k=1}^{n} |u(k+1) - u(k)|, \quad (34)$$

where $u(k)$ is the value of the control signal at the kth computation cycle. Minimizing the TV exponent is a computational proof of eliminating the rattling effect and increasing the energy efficiency of the software implementation of the algorithm.

Since the $tanh$ (32) function requires the calculation of exponents, which is a computationally expensive operation for embedded CPUs without hardware accelerators, an optimization method is proposed. In the onboard software, the $tanh(x)$ a function is approximated by a rational fraction using the Padé method:

$$tanh(x) \approx \frac{x + x^3/15}{1 + 2x^2/5}, \quad (35)$$

This allows calculating robust control using basic arithmetic operations, which minimizes computational latency and ensures determinism of the control cycle in the range of 500–1000 Hz.

### Sensitivity and robustness analysis of the hybrid system

To assess the robustness of the hybrid system, a comparative experiment was performed under the scenario <<holding position in constant wind>> (hover-start: the UAV is near the target, the target is unchanged). Three levels of constant horizontal wind were considered:

– $w = 3$ H – moderate wind (according to the MIL-F-8785C classification, corresponds to light turbulence at low altitudes);

– $w = 6$ H – strong wind (average turbulence);

– $w = 9$ H – extreme wind (exceeds the maximum horizontal thrust $F_{XY,max} \approx F_{total}\sin\theta_{max} \approx 30{,}0 \cdot \sin(30°) \approx 8{,}3$ H).

The results for the SS-MAE metric (mean absolute error in steady state, $t \in [0{,}67T;\, T]$) are given in Table 1.

Table 1

**SS-MAE positioning in steady wind (4 methods, 3 wind levels, T=30 s, hover-start)**

| Method | Wind 3 H (m) | Wind 6 H (m) | Wind 9 H (m) |
|---|---|---|---|
| PID | 1,1500 | 2,8976 | $> 140$ |
| SMC | 0,1372 | 0,3363 | $> 140$ |
| ASMC NoAI | 0,0989 | 0,3332 | $> 140$ |
| **ASMC+RSEFNN** | **0,0911** | **0,2843** | $> 140$ |
| Improvement vs PID | 12,6 × | 10,2 × | – |

The ASMC+RSEFNN method demonstrates the highest accuracy in steady state in winds of 3 H and 6 H:

– at $w = 3$ H: SS-MAE = 0,091 m – 12.6 times better than PID;

– at $w = 6$ H: SS-MAE = 0,284 m – 10.2 times better than PID.

It is important to note that the advantage of RSEFNN over ASMC NoAI in SS-MAE is relatively modest (0.091 vs 0.099 m at w=3 H, ≈8%). A significant advantage is manifested in the accuracy of the steady-state mode: RSEFNN compensates for the residual shift arising from the systematic wind disturbance, while ASMC NoAI retains the steady-state error due to the lack of a neuro-fuzzy identifier.

At w=9 H, all four controllers experience deviations of more than 140 m: the wind force exceeds the maximum horizontal compensation capacity of the 4DoF system (which is limited by the pitch angle and thrust of the main rotors). This result determines the applicability limit of the developed method in conditions of severe wind disturbances.

### Conclusions

The results of the study allowed to resolve the current scientific and technical contradiction between the need to increase the precision of navigation control of autonomous UAVs and the strict limitations of on-board computing resources. The proposed intellectually robust architecture, based on a combination of adaptive switching mode control (ASMC) and the recurrent neural network RSEFNN, provided effective online identification and compensation of external disturbances and dynamic uncertainties. Mathematical proof of the stability of the synthesized system using the direct Lyapunov method confirmed the asymptotic convergence of trajectory tracking errors to zero, while guaranteeing the numerical stability of the neural network training processes and preventing computer memory overflow.

The scientific novelty of the article lies in the improvement of the hybrid method of adaptive control of the variable mode of the UAV, which, unlike the classical robust methods, is based on the integration of an intelligent neural network observer for online compensation of nonlinear components of dynamics and disturbances. This combination made it possible to significantly reduce the gain coefficients of the discontinuous part of the controller, ensuring the minimization of the "rattling" effect and increasing the energy efficiency of the actuators while maintaining the mathematically proven global asymptotic stability of the closed-loop system using the Lyapunov method.

An important practical result was the implementation of methods for suppressing the "rattling" effect by replacing the discontinuous control function with its smooth approximation based on the adaptive boundary layer and hyperbolic tangent. Optimization of calculations using the Padé method made it possible to minimize algorithmic latency, ensuring the determinism of the control cycle at frequencies up to 1000 Hz even on built-in CPUs without specialized hardware accelerators. This allowed to significantly reduce the information load on the system data bus and increase the energy efficiency of UAV actuators by minimizing high-frequency oscillations of the command signal.

Comparative analysis with traditional approaches confirmed the high robustness of the developed method in conditions of intense wind loads. In particular, the use of the ASMC+RSEFNN hybrid controller allowed to increase the positioning accuracy in steady-state mode by 10.2–12.6 times compared to classical PID controllers in moderate and strong winds. The integrated neuro-fuzzy identifier provided compensation for systematic wind drift, which turned out to be a critical factor for performing precision guidance tasks in difficult meteorological conditions, up to the limit of the physical capabilities of the vehicle's traction system.

## ADDITIONAL INFORMATION

### DECLARATION ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE TOOLS

In preparing this work, the author used DeepL Translate and Grammarly for: grammar and spelling checks, paraphrasing, and rephrasing. After using these tools/services, the author reviewed and edited the content and takes full responsibility for the content of this publication.

## REFERENCES

1. Distributed Control of Cyber Physical System on Various Domains: A Critical Review / M. Hamzah et al. Systems. 2023. Vol. 11, no. 4. P. 208. URL: https://doi.org/10.3390/systems11040208.
2. Hybrid VTOL UAVs: A Comprehensive Review of Technology, Applications and Advancements / V. Sutariya et al. Lecture Notes in Mechanical Engineering. Singapore, 2025. P. 367–380. URL: https://doi.org/10.1007/978-981-97-5621-6_28.
3. Design and Flight Control of Tail-Sitter Uncrewed Aerial Vehicles: A Review / J. Zhu et al. IEEE/ASME Transactions on Mechatronics. 2026. P. 1–25. URL: https://doi.org/10.1109/tmech.2025.3643836.
4. Rinaldi M., Primatesta S., Guglieri G. A Comparative Study for Control of Quadrotor UAVs. Applied Sciences. 2023. Vol. 13, no. 6. P. 3464. URL: https://doi.org/10.3390/app13063464.
5. Assessment and Modeling of the Aerodynamic Ground Effect of A Fully-actuated Hexarotor with Tilted Propellers / A. Garofano-Soldado et al. IEEE Robotics and Automation Letters. 2024. P. 1–8. URL: https://doi.org/10.1109/lra.2024.3350975.
6. Lei Y., Luan C. Aerodynamic Analysis of a Hexacopter with an Inner Tilted-Rotor Configuration During Hovering. Aerospace. 2025. Vol. 12, no. 4. P. 317. URL: https://doi.org/10.3390/aerospace12040317.
7. Structural Design, Control and Experimental Validation of a Novel Hexacopter with Synchronized-Tilting Rotors / H. Zhou et al. 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023. 2023. URL: https://doi.org/10.23919/ccc58697.2023.10241208.
8. Cari no J., Abaunza H., Castillo P. A Fully-Actuated Quadcopter Representation using Quaternions. International Journal of Control. 2022. P. 1–0. URL: https://doi.org/10.1080/00207179.2022.2129789.
9. ProVANT-Emergentia: Embedded System Design for a Tiltrotor UAV / L. E. Gonzalez-Berbel et al. 2025 Brazilian Conference on Robotics (CROS), Belo Horizonte, Brazil, 28–30 April 2025. 2025. P. 1–6. URL: https://doi.org/10.1109/cros66186.2025.11066109.
10. ROS2 Real-time Performance Optimization and Evaluation / Y. Ye et al. Chinese Journal of Mechanical Engineering. 2023. Vol. 36, no. 1. URL: https://doi.org/10.1186/s10033-023-00976-5.
11. UAV Swarm Real-Time Rerouting by Edge Computing D* Lite Algorithm / M.-T. Lee et al. Applied Sciences. 2022. Vol. 12, no. 3. P. 1056. URL: https://doi.org/10.3390/app12031056.
12. Integration of ROS and Gazebo Platform using MAVLINK Protocol for Real-Time UAV Applications / A. K. Jha et al. ICDCN '24: International Conference on Distributed Computing and Networking, Chennai India. New York, NY, USA, 2024. URL: https://doi.org/10.1145/3631461.3631950.
13. Tahir M. A., Mir I., Islam T. U. Control Algorithms, Kalman Estimation and Near Actual Simulation for UAVs: State of Art Perspective. Drones. 2023. Vol. 7, no. 6. P. 339. URL: https://doi.org/10.3390/drones7060339.
14. Kose O., Oktay T. Simultaneous design of morphing hexarotor and autopilot system by using deep neural network and SPSA. Aircraft Engineering and Aerospace Technology. 2023. URL: https://doi.org/10.1108/aeat-07-2022-0178.
15. Uzun M., Oktay T. Simultaneous UAV having actively sweep angle morphing wing and flight control system design. Aircraft Engineering and Aerospace Technology. 2023. URL: https://doi.org/10.1108/aeat-09-2022-0259.
16. Schwung M., Lunze J. Cooperative Control of UAVs over an Unreliable Communication Network. IEEE Aerospace and Electronic Systems Magazine. 2022. P. 1–13. URL: https://doi.org/10.1109/maes.2022.3176595.
17. A novel approach for trajectory tracking control of an under-actuated quad-rotor UAV / K. Huang et al. IEEE/CAA Journal of Automatica Sinica. 2017. P. 1–10. URL: https://doi.org/10.1109/jas.2016.7510238.

18. Adaptive Barrier Fast Terminal Sliding Mode Actuator Fault Tolerant Control Approach for Quadrotor UAVs / A. Najafi et al. Mathematics. 2022. Vol. 10, no. 16. P. 3009. URL: https://doi.org/10.3390/math10163009.

Степан ТАНАСІЙЧУК
Хмельницький національний університет

# ГІБРИДНИЙ МЕТОД АДАПТИВНОГО КЕРУВАННЯ ЗМІННИМ РЕЖИМОМ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ ІЗ ІНТЕЛЕКТУАЛЬНОЮ ОНЛАЙН-КОМПЕНСАЦІЄЮ ЗБУРЕНЬ

*У статті розв'язано актуальну науково-технічну суперечність між необхідністю підвищення точності навігаційного керування автономними безпілотними літальними апаратами (БПЛА) та жорсткими обмеженнями ресурсів бортових обчислювальних систем. Запропоновано інтелектуально-робастну архітектуру керування, що базується на синтезі адаптивного методу змінного режиму (ASMC) та рекурентної нейро-нечіткої мережі RSEFNN. Наукова новизна роботи полягає в удосконаленні гібридного підходу, який, на відміну від класичних робастних методів, використовує інтелектуальний спостерігач для онлайн-ідентифікації та компенсації нелінійних складових динаміки й зовнішніх збурень. Це дозволило суттєво знизити коефіцієнти підсилення розривної частини контролера, мінімізувати ефект «деренчання» та підвищити енергоефективність актуаторів. Математичне доведення стійкості замкненої системи за прямим методом Ляпунова підтвердило асимптотичну збіжність помилок відстеження траєкторії до нуля та гарантувало чисельну стабільність процесів навчання нейромережі. Важливим практичним внеском є впровадження методів придушення високочастотних осциляцій через заміну розривної функції керування її гладкою апроксимацією на основі адаптивного граничного шару та гіперболічного тангенса. Для забезпечення детермінованості обчислювального циклу в реальному часі застосовано оптимізацію за методом Паде, що дозволило мінімізувати алгоритмічну латентність і досягти частоти керування до 1000 Гц на вбудованих CPU без спеціалізованих прискорювачів. Результати порівняльного аналізу підтвердили високу робастність розробленого методу в умовах інтенсивних вітрових навантажень. Зокрема, використання контролера ASMC+RSEFNN дозволило підвищити точність позиціонування у сталому режимі в 10,2–12,6 раза порівняно з класичними ПІД-регуляторами. Інтегрований нейро-нечіткий ідентифікатор забезпечив ефективну компенсацію систематичного вітрового зсуву, що є критичним фактором для виконання завдань прецизійного донаведення БПЛА у складних метеорологічних умовах.*

*Ключові слова: автономні безпілотні літальні апарати (БПЛА), адаптивне керування змінним режимом (ASMC), рекурентна нейро-нечітка мережа (RSEFNN), робастність системи керування, прецизійне донаведення, онлайн-ідентифікація збурень, метод Ляпунова, ефект деренчання (chattering), вбудовані системи реального часу, SWaP-обмеження, алгоритмічна латентність, метод Паде, мультисенсорне злиття даних, інваріантність керування.*