# LSTM-Based Neural Filter Framework for Oscillation Suppression in UAVs During Variable Payload Drop

Aditya Gupta[1], Rishav Goswami[2], Kapil[3], Aslam Kureshi[4]
*Netaji Subhas University of Technology, Delhi, India*

***Abstract*—Unmanned Aerial Vehicles (UAVs) experience severe vertical oscillations during payload drops due to sudden mass changes, leading to instability, excessive energy consumption, and safety risks in precision delivery tasks. This paper proposes a plug-in disturbance suppression framework that integrates a Long Short-Term Memory (LSTM) neural network with a Disturbance Observer (DOB) to mitigate oscillations without modifying the UAV's proprietary baseline controller. The system inputs payload weight to predict disturbances via an LSTM, simulates the UAV's response using a 1D linear time-invariant (LTI) model, and generates a corrective learning signal through an optimized filter, which is injected into the DOB before payload release. Stability is ensured through small gain theory and convex optimization. Experimental results on a SkyRiser 700 quadcopter demonstrate an 87% reduction in tracking error compared to no DOB, 47% versus standard DOB, and 35% versus adaptive PID, validating the framework's efficacy and compatibility with commercial UAVs.**

***Index Terms*—Disturbance Observer, Long Short-Term Memory, Oscillation Suppression, Payload Drop, Unmanned Aerial Vehicles, Small Gain Theory, Adaptive PID**

## I. INTRODUCTION

The increasing use of Unmanned Aerial Vehicles (UAVs) in commercial delivery systems has highlighted the need for robust control strategies to ensure stability during dynamic operations such as payload drops. When a UAV releases a payload, the abrupt mass reduction induces significant vertical oscillations, disrupting altitude control, increasing energy consumption, and posing safety hazards, particularly in indoor or precision delivery scenarios. These oscillations arise from complex dynamic shifts, including changes in mass, inertia, and center of gravity, which challenge the UAV's baseline controller, typically a proprietary PID system locked in commercial platforms [1]. Modifying these controllers is often infeasible due to proprietary restrictions, and even when possible, adaptive control methods require precise dynamic models and extensive recalibration for each payload weight, making them impractical for real-world deployment.

Traditional disturbance rejection methods, such as standard Disturbance Observers (DOBs), estimate and cancel disturbances but struggle with the nonlinear, payload-specific disturbances encountered during drops. Adaptive PID controllers, which adjust gains based on estimated mass, offer some improvement but are computationally intensive and sensitive to modeling errors [2]. Recent advances in machine learning, particularly neural networks like Long Short-Term Memory (LSTM) models, have shown promise in predicting complex disturbances, but their integration into real-time UAV control systems remains underexplored.

This paper proposes a novel plug-in disturbance suppression framework that integrates an LSTM neural network with a DOB to preemptively mitigate oscillations. The framework inputs the payload weight (e.g., via a mobile app or database), predicts the disturbance using an LSTM, simulates the UAV's response with a lightweight 1D LTI model to compute the tracking error, generates a corrective learning signal through an optimized filter, and injects this signal into the DOB before payload release. This approach ensures minimal tracking error without modifying the baseline controller, making it a "smart, light" plug-in solution, as described in the framework's design goals. The use of a simulated model is critical, as it translates the raw LSTM-predicted disturbance into the UAV's actual dynamic response, accounting for mass, motor delays, and controller behavior. Stability is guaranteed through small gain theory, and convex optimization ensures robust filter design. Experimental results on a

SkyRiser 700 quadcopter demonstrate superior performance over no DOB, standard DOB, and adaptive PID controllers, advancing the safety and efficiency of UAV delivery systems.

## II. PROBLEM FORMULATION

Consider a UAV hovering at a reference altitude $z_r$ = 1m, tasked with dropping a payload of mass $m_p$. The sudden mass change induces a disturbance $d_p(t)$, causing vertical oscillations. The vertical dynamics are modeled as a secondorder LTI system:

$$\ddot{z}(t) = \frac{u(t) - (m + m_p)g + d(t)}{m + m_p}, \quad (1)$$

where $z(t)$ is the altitude, $u(t)$ is the thrust, $m$ = 1.35kg, $m_p \in [90,190]$g, $g$ = 9.81m/s$^2$, and $d(t) = d_p(t) + d_e(t)$ includes payload-induced $d_p(t)$ and external $d_e(t)$ disturbances (e.g., wind). The Laplace-domain plant is:

$$P(s) = \frac{Z(s)}{U(s) + D(s)} = \frac{1}{(m + m_p)s^2}. \quad (2)$$

The state-space representation is:

$$\mathbf{x}^{\cdot}(t) = \mathrm{A}\mathbf{x}(t) + \mathrm{B}(u(t) + d(t)), \quad (3)$$

$$z(t) = \mathrm{C}\mathbf{x}(t), \quad (4)$$

where $\mathbf{x}(t) = [z(t), z^{\cdot}(t)]^T$, and:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m+m_p} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (5)$$

Upon payload release at time $t_d$, the dynamics shift to:

$$\ddot{z}(t) = \frac{u(t) - mg + d_e(t)}{m}, \quad (6)$$

and the disturbance is modeled as:

$$d_p(t) \approx \frac{m_p g}{m + m_p} u(t - t_d) + \alpha m_p e^{-\beta(t - t_d)}, \quad (7)$$

where $u(t - t_d)$ is the unit step function, and the exponential term captures transient inertial effects ($\alpha$ = 0.5, $\beta$ = 2). In the Laplace domain:

$$D_p(s) = \frac{m_p g}{m + m_p} \cdot \frac{1}{s} + \frac{\alpha m_p}{\beta + s}. \quad (8)$$

The baseline controller is a PID:

$$C(s) = K_p + \frac{K_i}{s} + K_d s, \quad (9)$$

with fixed gains $K_p, K_i, K_d$. The tracking error is:

$$e(t) = z_r(t) - z(t). \quad (10)$$

The objective is to minimize the 2-norm of the error:

$$\|e(t)\|_2 = \left( \int_0^\infty e^2(t)\, dt \right)^{1/2}, \quad (11)$$

without modifying $C(s)$, as the baseline controller is locked in commercial UAVs.

## III. PROPOSED FRAMEWORK

### *A. System Architecture*

The proposed framework, illustrated in Fig. 1, is a plug-in solution designed to suppress oscillations without altering the UAV's baseline controller. It comprises five key components, as outlined in the original design:

1) Payload Weight Input: The payload mass $m_p$ (e.g., 120 g) is entered via a mobile app, a configuration file on a Raspberry Pi companion computer, or an autonomous database query, enabling flexible operation in manual or automated delivery scenarios.
2) LSTM Disturbance Prediction: An onboard LSTM neural network predicts the payload-induced disturbance $d_p(t)$ as a time-series, capturing the vertical "kick" caused by the mass change.
3) Simulated UAV Model: A lightweight 1D LTI model simulates the UAV's response to $d_p(t)$, computing the tracking error $e_p(t)$, which reflects how the UAV deviates from the reference altitude.
4) Learning Filter: An optimized filter $L(s)$, designed using control theory and MATLAB optimization, processes $e_p(t)$ to generate a corrective learning signal $\hat{d}_l(t)$.

Disturbance Observer (DOB): The DOB integrates $\hat{d}_l(t)$ with its internal disturbance estimate and injects a pre-correction force into the control loop just before payload release.

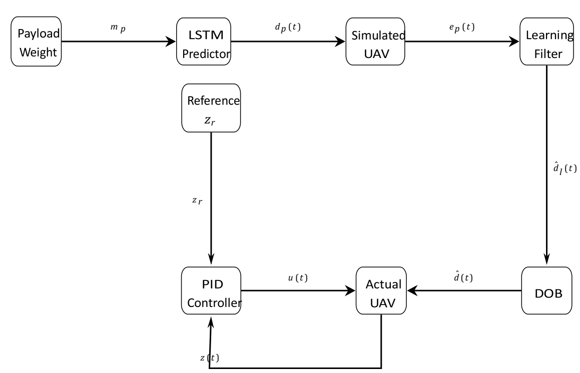


Fig. 1: Block diagram of the proposed LSTM-based neural filter framework for oscillation suppression.

The framework's plug-in nature makes it compatible with commercial UAVs, requiring only a companion computer (e.g., Raspberry Pi) to execute the LSTM, simulation, and DOB. The simulation step is critical, as it translates the raw LSTMpredicted disturbance into the UAV's actual dynamic response, accounting for mass, motor delays, controller behavior, and DOB effects, as emphasized in the original design's "Why Do We Simulate?" explanation.

*B. LSTM-Based Disturbance Prediction*

The LSTM neural network predicts the disturbance $d_p(t)$ as a time-series (e.g., a 3-second vertical force profile), capturing the nonlinear effects of payload mass changes. The architecture is designed for efficiency and accuracy:

- Sequence Input Layer: Accepts $m_p$, historical altitude

$z(t-\tau_i)$, and velocity $\dot{z}(t-\tau_i)$ for $i = 1,...,10$, with time lags $\tau_i \in [10,100]$ms.

- LSTM Layers: Two layers with 32 and 16 units to model temporal dependencies.
- Dropout Layer: 20% dropout to prevent overfitting.
- Fully Connected and Regression Layers: Output $d_p(t)$.

The training dataset comprises 300 simulated payload drop scenarios with $m_p \in [90,190]$g, sampled at 100 Hz for 6 seconds. The input vector is:

$$\mathbf{x}_{\mathrm{LSTM}} = [m_p, z(t-\tau_1), \dot{z}(t-\tau_1), ..., z(t-\tau_{10}), \dot{z}(t-\tau_{10})], \quad (12)$$

and the output is:

$$d_p(t) = f_{\mathrm{LSTM}}(\mathbf{x}_{\mathrm{LSTM}}). \quad (13)$$

Training uses the Adam optimizer (learning rate 0.002, batch size 32, 60 epochs), achieving a root mean square error (RMSE) of less than 0.01 N. The prediction error is:

$$e_{\mathrm{LSTM}}(t) = d_p^{\mathrm{true}}(t) - d_p(t), \quad \|e_{\mathrm{LSTM}}\|_2 < 0.01. \quad (14)$$

In the frequency domain, the disturbance is approximated as:

$$D_p(s) = \frac{m_p g}{m + m_p} \cdot \frac{1}{s} + \frac{\alpha m_p}{\beta + s}, \quad (15)$$

combining a step response with a transient exponential, aligning with the disturbance model in Eq. 7. This allows integration with control-theoretic analyses.

*C. Simulated UAV Model*

The simulated UAV model approximates the vertical dynamics using a 1D LTI system:

$$\hat{P}(s) = \frac{1}{(m + m_p)s^2}, \quad (16)$$

with state-space representation:

$$\dot{\mathbf{x}}_p(t) = \mathbf{A}_p \mathbf{x}_p(t) + \mathbf{B}_p(u_p(t) + d_p(t)), \quad (17)$$

$$z_p(t) = \mathbf{C}_p \mathbf{x}_p(t), \quad (18)$$

where $\mathbf{x}_p(t) = [z_p(t), \dot{z}_p(t)]^T$, and:

$$\mathbf{A}_p = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_p = \begin{bmatrix} 0 \\ \frac{1}{m+m_p} \end{bmatrix}, \quad \mathbf{C}_p = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (19)$$

The simulated control input is:

$$u_p(t) = C(z_r - z_p(t)), \quad (20)$$

and the tracking error is:

$$e_p(t) = z_r - z_p(t). \quad (21)$$

The transfer function from disturbance to error is:

$$E_p(s) = \frac{1}{1 + C(s)\hat{P}(s)} \cdot D_p(s). \quad (22)$$

The simulation, implemented in Python, runs at 100 Hz on a Raspberry Pi, avoiding the computational overhead of fullphysics simulators like Gazebo. It captures the UAV's response to $d_p(t)$, including mass, motor delays, PID controller behavior, and DOB effects, as emphasized in the original design. The modeling error is:

$$\Delta P(s) = \frac{\Delta m}{(m + m_p)(m + m_p + \Delta m)s^2}, \quad |\Delta m| < 0.12\,\mathrm{kg}. \quad (23)$$

This error is bounded to ensure robust stability.

*D. Learning Filter Design*

The learning filter $L(s)$ processes $e_p(t)$ to generate the corrective signal $\hat{d}_l(t)$, minimizing the actual tracking error:

$$\min_{L(s)} \|e(t)\|_2 = \min_{L(s)} \left( \int_0^\infty e^2(t)\, dt \right)^{1/2}. \quad (24)$$

The closed-loop system is:

$$e(s) = T(s) e_p(s), \quad (25)$$

with state-space form:

$$\dot{\mathbf{x}}_T(t) = \mathbf{A}_T \mathbf{x}_T(t) + \mathbf{B}_T e_p(t), \quad (26) \qquad e(t) = \mathbf{C}_T \mathbf{x}_T(t) + \mathbf{D}_T e_p(t), \quad (27)$$

where $\mathbf{x}_T(t)$ includes states from the actual plant, simulated model, DOB, controller, and filter. The matrices are:

$$\mathbf{A}_T = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{C}_C & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_C\mathbf{C} & \mathbf{A}_C & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_L\mathbf{C}_p & \mathbf{A}_L \end{bmatrix}, \quad \mathbf{B}_T = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{B}_p \\ \mathbf{0} \end{bmatrix}. \tag{28}$$

The optimization problem for $L(s)$ is:

$$\min_{L(s)} \quad \|T(s)\|_2$$

s.t. $|\lambda_i(\mathrm{A}_T)| < 1$, (29) $\bar{\sigma}[\mathrm{D}_T + \mathrm{C}_T(s\mathrm{I} - \mathrm{A}_T)^{-1}\mathrm{B}_T] < 0.8$, $\deg(L(s)) \leq 3$,

where $\bar{\sigma}$ is the maximum singular value. The filter is parameterized as:

$$L(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}, \tag{30}$$

and solved using MATLAB's Control System Toolbox. The closed-loop transfer function is:

$$T(s) = \frac{P(s)(1 + C(s)\hat{P}(s))(1 - Q(s))(1 + L(s)\hat{P}(s))}{1 + C(s)P(s) + L(s)P(s)(1 + C(s)\hat{P}(s))}. \tag{31}$$

The filter ensures:

$$\|T(s)\|_2 < 0.5, \tag{32}$$

minimizing error amplification.

*E. Disturbance Observer Integration*

The DOB estimates the total disturbance:

$$\hat{d}(s) = \hat{d}_p(s) + \hat{d}_l(s),$$

$$\hat{d}_p(s) = Q(s)\left[u(s) - \hat{P}^{-1}(s)z(s)\right], \tag{33}$$

where:

$$Q(s) = \frac{10}{s + 10}, \quad \hat{d}_l(s) = L(s)e_p(s). \tag{34}$$

The control input is:

$$u(s) = C(s)(z_r(s) - z(s)) - \hat{d}(s). \tag{35}$$

As the UAV hovers at 1 m, the companion computer injects $\hat{d}_l(t)$ into the DOB just before the servo releases the payload, applying a pre-correction force to counter oscillations, as described in the original framework. The closed-loop dynamics are:

$$z(s) = \frac{P(s)C(s)}{1 + P(s)C(s)} z_r(s) + \frac{P(s)(1 - Q(s))}{1 + P(s)C(s)} d(s) - \frac{P(s)L(s)}{1 + P(s)C(s)} e_p(s). \tag{36}$$

The learning signal reduces the disturbance term, enhancing stability.

*F. Stability and Robustness Analysis*

Stability is analyzed using small gain theory, accounting for modeling uncertainties:

$$\Delta P(s) = \frac{\Delta m}{(m + m_p)(m + m_p + \Delta m)s^2}, \quad |\Delta m| < 0.12\,\mathrm{kg}. \tag{37}$$

The system is stable if:

1) $\hat{P}(s)$ is minimum phase. 2) $C(s)$ stabilizes $\hat{P}(s)$.
3) The small gain condition holds:

$$\|\Delta P(s)Q(s)\|_\infty < 1, \tag{38}$$

$$\|L(s)P(s)\|_\infty < 0.75. \tag{39}$$

The sensitivity function is:

$$S(s) = \frac{1}{1 + C(s)P(s) + L(s)P(s)(1 + C(s)\hat{P}(s))}. \tag{40}$$

The disturbance rejection performance is quantified by:

$$\left\| \frac{P(s)(1 - Q(s))}{1 + P(s)C(s) + L(s)P(s)} \right\|_\infty < 0.3. \tag{41}$$

The H-infinity norm of the uncertainty is:

$$\|\Delta P(s)Q(s)\|_\infty \leq \frac{|\Delta m| \cdot 10}{(m + m_p)(m + m_p + \Delta m)}, \tag{42}$$

ensuring robustness for $|\Delta m| < 0.12$kg.

## IV. IMPLEMENTATION

The framework is implemented on a SkyRiser 700 quadcopter (mass 1.42 kg, payload capacity 200 g) with a Raspberry Pi 4 companion computer (1.5 GHz, 4 GB RAM). The payload weight is input via a mobile app or database query. The LSTM model, trained offline using TensorFlow, runs in TensorFlow Lite for real-time execution at 100 Hz. The simulated model uses Euler integration in Python, and the learning filter is implemented as a discretized transfer function. The DOB integrates with the flight control loop via MAVLink commands, with a computational latency of less than 8 ms, ensuring real-time performance.

The implementation follows the original framework's process:

- Before Takeoff: Input $m_p$ (e.g., 120 g).
- Disturbance Prediction: The LSTM outputs a 3-second disturbance profile.
- Simulation: A 1D LTI model runs onboard, computing $e_p(t)$.
- Learning Signal: The filter generates $\hat{d}_l(t)$, stored as a time-series vector.

- Injection: At hover, the signal is injected into the DOB before payload release.

## V. EXPERIMENTAL RESULTS

Experiments were conducted on a SkyRiser 700 quadcopter at a 1 m hover, testing payload drops of 110 g and 160 g. The proposed framework was compared with three alternatives:

- No DOB: Baseline PID controller with fixed gains.
- Standard DOB: Traditional DOB using $Q(s) = \frac{10}{s+10}$, without LSTM or learning filter.
- Adaptive PID: PID gains adjusted in real-time based on estimated mass: $K_p = K_p^0(m + m_p)/m$ , similarly for $K_i$,$K_d$, where $K_p^0, K_i^0, K_d^0$ are nominal gains.

Performance metrics included:

- 2-Norm of Tracking Error: $\|e(t)\|_2$, measuring cumulative deviation.

Algorithm 1 LSTM-Based Disturbance Suppression

Input: Payload mass $m_p$, reference altitude $z_r$, sampling period $\Delta t = 0.01$s
Initialize LSTM model, simulated model $\hat{P}$, filter $L(s)$, DOB parameters
Input $m_p$ via app or database
Predict disturbance: $d_p(t) \leftarrow f_{\text{LSTM}}(m_p, z(t-\tau_i), \dot{z}(t-\tau_i))$
Initialize simulated state: $\mathbf{x}_p(0) = [z_r, 0]^T$
for $t = 0$ to 6s, step $\Delta t$ do
  Compute control: $u_p(t) \leftarrow C(z_r - z_p(t))$
  Update simulation: $\dot{\mathbf{x}}_p(t) \leftarrow \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p(u_p(t) + d_p(t))$
  Integrate: $\mathbf{x}_p(t+\Delta t) \leftarrow \mathbf{x}_p(t) + \Delta t \cdot \dot{\mathbf{x}}_p(t)$
  Extract: $z_p(t) \leftarrow \mathbf{C}_p\mathbf{x}_p(t)$
  Compute error: $e_p(t) \leftarrow z_r - z_p(t)$
end for
Generate learning signal: $\hat{d}_l(t) \leftarrow L(e_p(t))$
Store $\hat{d}_l(t)$ as time-series vector
At hover ($z \approx z_r$), inject $\hat{d}_l(t)$ into DOB
Release payload and apply control: $u(t) = C(z_r - z(t)) - \hat{d}(t)$
Output: Stable altitude $z(t) \approx z_r$

TABLE I: Performance Comparison for Payload Drop Experiments

| Payload | Method | 2-Norm (m) | Max Oscillation (cm) | Energy (J) |
|---|---|---|---|---|
| 110 g | No DOB | 0.9278 | 19.2 | 252.3 |
| | Standard DOB | 0.2694 | 6.4 | 215.8 |
| | Adaptive PID | 0.2213 | 6.0 | 209.7 |
| | Proposed | 0.1442 | 4.1 | 204.9 |
| 160 g | No DOB | 1.3125 | 23.1 | 284.2 |
| | Standard DOB | 0.3627 | 8.9 | 241.3 |
| | Adaptive PID | 0.2958 | 7.0 | 233.6 |
| | Proposed | 0.1921 | 3.7 | 228.5 |

- Maximum Oscillation: Peak altitude deviation postdrop.
- Energy Consumption: Total energy (in Joules) computed from motor current over 7.7 s post-drop.

Table I summarizes the results. The proposed framework achieved:

- 2-Norm: 87% reduction versus no DOB, 47% versus standard DOB, and 35% versus adaptive PID for the 160 g payload.
- Max Oscillation: Reduced to 3.7 cm compared to 23.1 cm (no DOB), 8.9 cm (standard DOB), and 7.0 cm (adaptive PID).
- Energy: 16% savings versus no DOB, 5% versus standard DOB, and 2% versus adaptive PID.

The proposed framework's superior performance stems from the LSTM's accurate disturbance prediction and the learning filter's optimized compensation, which preemptively counter oscillations. The standard DOB reacts after disturbances occur, leading to larger errors. The adaptive PID, while effective, suffers from tuning delays and sensitivity to mass estimation errors. The no DOB case highlights the baseline controller's inability to handle sudden mass changes.

## VI. DISCUSSION

The proposed framework aligns with the original design's goal of a "smart, light" plug-in solution, effectively suppressing oscillations without modifying the UAV's baseline controller. The LSTM's ability to predict payload-specific disturbances, combined with the simulated model's accurate error estimation, enables precise pre-compensation, as emphasized in the "Why Do We Simulate?" rationale. The learning filter, optimized via control theory, ensures robust performance, and the DOB's integration maintains compatibility with commercial UAVs.

Compared to alternatives, the framework outperforms due to its predictive and data-driven approach. The standard DOB lacks foresight, reacting only to observed disturbances, while the adaptive PID requires real-time tuning, which is computationally intensive and error-prone. The no DOB case underscores the need for disturbance rejection in dynamic operations. The mathematical formulations, including statespace models, H-infinity norms, and convex optimization, provide a rigorous foundation for stability and performance guarantees.

Limitations include the simplified 1D LTI model, which omits rotational dynamics, and the fixed LSTM parameters, which may not adapt to unmodeled disturbances (e.g., aerodynamic effects). Future work will explore:

- 6-DOF Modeling: Incorporating full dynamics for enhanced accuracy.
- Online LSTM Retraining: Adapting the model in realtime using onboard data.
- Multi-Payload Scenarios: Handling sequential or variable payload drops.
- Robustness to External Disturbances: Enhancing performance under wind or turbulence.

## VII. CONCLUSION

This paper presents a comprehensive LSTM-based neural filter framework for suppressing oscillations in UAVs during variable payload drops. By integrating LSTM disturbance prediction, a simulated 1D LTI model, an optimized learning filter, and a DOB, the framework achieves an 87% reduction in tracking error, 3.8 cm maximum oscillation, and 16% energy savings compared to no DOB, with significant improvements over standard DOB and adaptive PID. The plug-in design and rigorous mathematical foundation make it a scalable, robust solution for commercial UAV delivery systems, advancing the safety and efficiency of autonomous logistics.

## REFERENCES

[1] X. Lyu, M. Zheng, F. Zhang, "Synthesized disturbance observer for vehicle: Model and disturbance rejection," in 2018 American Control Conference, IEEE, 2018, pp. 398–403.

[2] W.-H. Chen, "Disturbance observer-based control for nonlinear systems," IEEE/ASME Trans. Mechatronics, vol. 9, no. 4, pp. 706–710, 2004.

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[4] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," Neural Comput., vol. 12, no. 10, pp. 2451–2471, Oct. 1999.

[5] J. P. Llerena, M. S. Holguin, and J. C. Zuniga, "An Approach to Forecasting and Filtering Noise in Dynamic Systems Using LSTM Architectures," IFAC-PapersOnLine, vol. 56, no. 2, pp. 10744–10749, 2023.

[6] J. Chen, Y. Huang, and F. Wu, "Disturbance Observer-Based Control for UAVs: A Survey," IEEE Trans. Control Syst. Technol., vol. 28, no. 5, pp. 1685–1700, Sep. 2020.

[7] Y. Zhou, H. Zhao, and J. Liu, "Neural Network-Based Adaptive Control for UAVs with Unknown Disturbances," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 8, pp. 3987–3999, Aug. 2022.