

# A GP-Enhanced MPC Framework with Information-Driven Adaptation and Risk-Awareness for UAV Trajectory Tracking

**Abstract**—High-precision trajectory tracking control for UAV faces significant challenges from unmodeled dynamics and unknown time-varying disturbances, and general online learning methods often struggle to meet real-time requirements due to computational bottlenecks. To this end, this paper proposes an information-driven, adaptive risk-aware Model Predictive Control (AR-MPC) framework. The framework employs a hierarchical learning strategy, compensating for system biases with an offline-trained static Gaussian Process (GP), and designs an efficient online GP, supported by an asynchronous architecture and an information-value-driven data management mechanism, to compensate for dynamic residuals. Meanwhile, by integrating the uncertainty from online learning into the MPC cost function, an autonomous trade-off between performance and risk is achieved. Experimental results under time-varying wind fields demonstrate that the proposed framework reduces trajectory tracking error by up to 44% compared to a static GP-MPC baseline, while satisfying 50 Hz real-time control requirements.

**Keywords**—Gaussian process, Learning-based control, Model predictive control, UAV trajectory tracking, Adaptive systems

## I. INTRODUCTION

In recent years, Unmanned Aerial Vehicle (UAV) technology has demonstrated immense potential in cutting-edge applications such as search and rescue, high-speed inspection, and drone racing, owing to its exceptional maneuverability and deployment flexibility (Agha-mohammadi et al., 2014; Langston, 2024). These complex scenarios impose stringent requirements on the autonomous control systems of UAVs, demanding precise and reliable tracking of complex three-dimensional trajectories under extreme flight conditions of high speed and aggressive maneuvers. Model Predictive Control (MPC), with its forward-looking optimization capabilities and its ability to explicitly handle constraints on multivariable system states and inputs, has become one of the mainstream technologies for achieving high-performance trajectory tracking for unmanned systems (Gedefaw et al., 2025; Hewing et al., 2020; Miranda-Colorado et al., 2024).

However, the performance of MPC is fundamentally dependent on the accuracy of its internal predictive model. A UAV, as a typical nonlinear, strongly coupled, and underactuated system, possesses highly complex dynamics (Liu & Toth, 2021; Schmid et al., 2022). The nominal model, traditionally established from first principles, can describe the main dynamic characteristics but often neglects complex aerodynamic effects (e.g., fuselage drag, rotor-to-rotor downwash interference) and other unmodeled dynamics that become increasingly significant during high-speed flight (Ventura Diaz & Yoon, 2018). This mismatch between the model and physical reality is drastically amplified when the UAV performs aggressive maneuvers, causing the MPC to

generate erroneous predictions and suboptimal control decisions, thereby severely compromising trajectory tracking accuracy. This phenomenon reveals a fundamental performance ceiling for control strategies that rely solely on a nominal model: when unmodeled dynamics become the dominant source of error, merely optimizing the control algorithm itself can no longer yield significant improvements—the bottleneck lies in the model itself.

To break through this performance ceiling, data-driven modeling strategies have emerged (Martinez-Vasquez et al.; Zhu et al.), with machine learning-based model enhancement methods becoming a research hotspot. Several distinct philosophies have been developed. The RTN-MPC framework by Salzmann et al. integrates a deep neural network with millions of parameters into the real-time MPC loop via an efficient computational architecture (Salzmann et al., 2023), relying on the model's immense expressive power and massive offline data for generalization. In contrast, the L1-NMPC framework by Hanover et al. cascades a fast L1 adaptive controller atop a standard NMPC, rapidly compensating for model-reality discrepancies at the actuator level (Hanover et al., 2021). Another core idea is to learn and compensate for the difference between the nominal model and the true dynamics, the so-called "residual dynamics" (Choo & Kayacan, 2023; Kulathunga et al., 2024; Liu et al., 2022; Torrente et al., 2021; Wei et al., 2024). This approach directly integrates the learned residual term into the state-space equations, enabling more accurate future predictions. Among these, the Gaussian Process (GP), a powerful non-parametric Bayesian regression tool, is highly favored for its ability to learn complex nonlinear functions and provide quantified uncertainty for its predictions (Rasmussen, 2003). The pioneering work of Torrente et al., which we refer to as SGP-MPC, is a distinguished representative of this direction. They proposed a GP-enhanced MPC framework that uses extensive offline flight data to learn and compensate for systematic, repeatable model errors in the UAV dynamics (Torrente et al., 2021). This method has achieved great success, powerfully demonstrating the effectiveness of offline learning for correcting static model biases.

Although offline learning methods are highly effective in handling repeatable static errors, their core limitation is the static nature of the learned model. Once training is complete, the GP model is fixed and cannot adapt to dynamically changing disturbances not fully represented in the training data, such as time-varying wind fields, payload changes, or ground effects (Wei et al., 2024). Consequently, to achieve truly robust autonomous flight in unstructured environments, the controller must possess online adaptive capabilities (Kocijan, 2016; Maiworm et al., 2021).

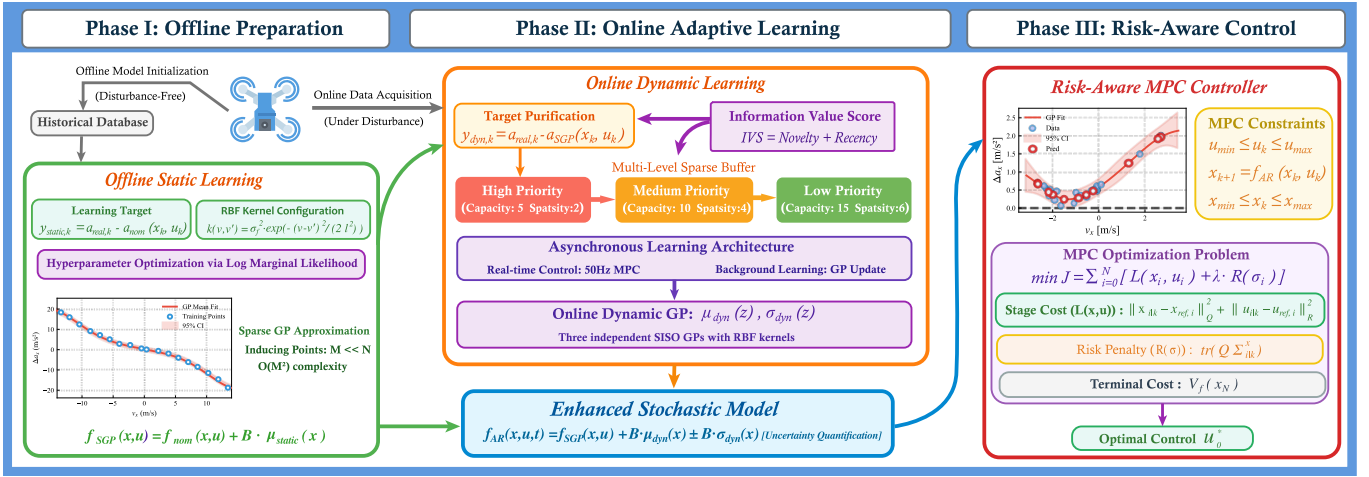


Fig. 1. The overall architecture of the proposed Adaptive Risk-aware Model Predictive Control (AR-MPC) framework.

However, online adaptation faces two severe challenges. First, the real-time computational bottleneck: the training cost of a standard GP scales cubically with the number of data points, which is unacceptable for an MPC controller that must operate within milliseconds (Guerrero-Fernandez et al., 2023; Miranda-Colorado et al., 2024). Second, inefficient data management: conventional methods like sliding windows or forgetting factors manage data passively based on time, failing to actively distinguish and retain high-value data points. To address these challenges, researchers have explored various avenues. For instance, Schmid et al. accelerated inference and gradient calculations by reformulating the GP as a state-space model (Schmid et al., 2022), while Liu et al. used a dual-GP architecture with a recursive least squares algorithm to update the online model and mitigate catastrophic forgetting (Liu et al., 2022).

Building on the analysis above, this paper proposes an Adaptive Risk-aware Model Predictive Control (AR-MPC) framework. Its main contributions are summarized as follows:

- We propose an integrated control architecture (AR-MPC) that combines online adaptive learning with risk-aware decision-making for robust UAV trajectory tracking under unknown and time-varying disturbances. The framework continuously refines its internal model and explicitly reasons about predictive uncertainty to ensure robust performance.
- To ensure computational tractability, we introduce a novel online learning mechanism that builds upon a hierarchical framework decoupling model residuals into static and dynamic components. While an asynchronous learning architecture is employed to satisfy real-time constraints, we design an information-value-driven data management strategy. This strategy intelligently curates the data stream to enhance the learning efficiency of the online GP.
- We develop a risk-aware MPC formulation that directly integrates the predictive uncertainty from the online GP into the optimization. By penalizing the predicted state

variance in the cost function, the controller autonomously balances aggressive tracking performance against the risks of model uncertainty. This results in a cautious yet high-performing control strategy that enhances safety during flight in unpredictable conditions.

The overall architecture of our proposed AR-MPC framework is illustrated in Fig. 1. Experiments show that when dealing with unknown dynamic disturbances, the AR-MPC can significantly improve trajectory tracking performance while meeting real-time requirements.

## II. MODELING

### A. Nominal UAV Dynamics

We consider the UAV as a six-degree-of-freedom rigid body moving in three-dimensional space. Its state vector  $x \in \mathbb{R}^{13}$  is defined as:

$$x = [p^T, q^T, v^T, \omega^T]^T \quad (1)$$

where  $p \in \mathbb{R}^3$  is the position of the UAV's center of mass in the world inertial frame  $W$ ;  $q \in S^3$  is the unit quaternion representing the rotation from the body frame  $B$  to the world frame  $W$ ;  $v \in \mathbb{R}^3$  is its velocity in the world frame; and  $\omega \in \mathbb{R}^3$  is its angular velocity in the body frame. The control input  $u \in \mathbb{R}^4$  represents the thrust produced by the four rotors.

The nominal dynamics model of the UAV,  $\dot{x} = f_{\text{nom}}(x, u)$ , originates from the Newton-Euler equations and can be expressed as the following continuous-time state-space equation:

$$\dot{x} = f_{\text{nom}}(x, u) = \begin{bmatrix} v \\ \frac{1}{2} q \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} \\ \frac{1}{m} R_q T_u - g e_3 \\ J^{-1}(\tau_u - \omega \times J \omega) \end{bmatrix} \quad (2)$$

where  $m$  is the mass of the UAV,  $g$  is the gravitational acceleration, and  $e_3 = [0, 0, 1]^T$ .  $R_q \in SO(3)$  is the rotation matrix converted from the quaternion  $q$ .  $T_u = [0, 0, \sum_{i=1}^4 u_i]^T$  is the total thrust vector in the body frame  $B$ .  $J$  is the inertia tensor, and  $\tau_u$  is the total torque produced by the rotor thrusts. The symbol  $\otimes$  denotes quaternion multiplication.

However, the actual dynamics of a flying UAV,  $\dot{x}_{real}$ , are affected by various unmodeled factors. Thus, the true dynamics can be written as:

$$\dot{x}_{real} = f_{nom}(x, u) + d_{total}(x, u, t) \quad (3)$$

where  $d_{total}(x, u, t) \in R^{13}$  is the lumped model mismatch term, which includes complex aerodynamic effects, external time-varying wind disturbances, motor dynamic delays, and all other unmodeled dynamics. It is the fundamental reason for the controller's performance degradation during aggressive maneuvers. Directly modeling this complex, time-varying disturbance term is extremely difficult.

### B. Model-Augmented State-Space Representation

To effectively handle the complex model mismatch term, we decouple  $d_{total}(x, u, t)$  into two parts: a state-dependent static residual term  $d_{static}(x, u)$ , and a dynamic residual term  $d_{dyn}(x, u, t)$  that changes in real-time during flight.

$$d_{total}(x, u, t) \approx d_{static}(x, u) + d_{dyn}(x, u, t) \quad (4)$$

This decomposition has a clear physical correspondence.  $d_{static}$  primarily captures inherent, repeatable system errors, which mainly arise from intrinsic aerodynamic effects related to the flight state, particularly the velocity components.  $d_{dyn}$  is used to represent changing external disturbances encountered during flight, such as continuously varying wind fields or the sudden appearance and disappearance of ground effects.

Based on this model decomposition, we employ Gaussian Processes (GPs) as non-parametric regression tools. A static GP, trained on offline data, is used to model  $d_{static}$ , while an online adaptive GP is used for the real-time estimation of  $d_{dyn}$ . The specific implementation of the latter constitutes a core contribution of this paper. Physically, these unmodeled residuals act on the UAV primarily as forces and torques, ultimately affecting its acceleration. Therefore, the true acceleration areal can be approximated by our final augmented model  $a_{AR}$ :

$$a_{real}(x, u, t) \approx a_{AR}(x, u) + g_{static}(z) + g_{dyn}(z, t) \quad (5)$$

where  $z$  is chosen as a low-dimensional feature vector most relevant to the residuals; in our experiments, we select it to be the velocity components in each dimension.  $g_{static}(z)$  and  $g_{dyn}(z, t)$  are the mean function predictions of the two independent GPs, respectively. By substituting this augmented acceleration model, we obtain the final state-space model for the design of the AR-MPC controller:

$$\dot{x} = f_{AR}(x, u, t) = f_{nom}(x, u) + B_d (g_{static}(z) + g_{dyn}(z, t)) \quad (6)$$

where  $B_d = [0_{3 \times 3}, 0_{3 \times 4}, I_{3 \times 3}, 0_{3 \times 3}]^T$  is a constant distribution matrix that maps the 3-dimensional acceleration residual to the corresponding positions in the 13-dimensional state derivative vector. In this way, we internalize the effects of external time-varying disturbances into the prediction model, enabling the MPC to proactively address these disturbances.

To apply this continuous-time model in a discrete-time MPC framework, we need to perform numerical integration. Since the enhanced model contains complex nonlinear GP terms, it cannot be discretized analytically. Therefore, we use the classic fourth-order Runge-Kutta method. RK4 obtains an accurate estimate of the next state  $x_{k+1}$  by performing a weighted average of four evaluations of the state derivative function  $f_{AR}$  within each discrete time step  $\Delta t$ . Let the coefficients be  $k_1, k_2, k_3, k_4$ . Finally, the discrete state-space model we obtain can be represented as a nonlinear function:

$$x_{k+1} = f_{AR}(x_k, u_k) = x_k + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (7)$$

## III. DUAL GAUSSIAN PROCESS REGRESSION

To accurately estimate and compensate for the static and dynamic model mismatch components defined in Section II, we adopt a hierarchical modeling strategy. First, an offline, sparse-approximation-based static GP is utilized to capture inherent system biases. Subsequently, an online adaptive GP is designed to estimate and compensate for dynamic residuals in real-time. The core of this section lies in presenting a novel, information-value-driven intelligent data management and learning method for this online GP, aimed at resolving the fundamental trade-off between data redundancy and catastrophic forgetting in high-frequency MPC.

### A. Offline Modeling of Static Residuals

To reduce the dimensionality and computational complexity of the learning problem, we decouple the three-dimensional residual vector into three independent SISO GP models. Each GP model is responsible for learning a specific axial mapping, i.e., from the body velocity  $v_{B,i}$  on that axis to the corresponding acceleration residual  $a_{res,i}$ , where  $i \in \{x, y, z\}$ . The learning target for this GP,  $y_{static,k}$ , is defined as the residual between the observed true acceleration  $a_{real,k}$  at sampling time  $k$  and the acceleration  $a_{nom}(x_k, u_k)$  predicted by the nominal dynamics model, with measurement noise  $w_k$ :

$$y_{static,k} = a_{real,k} - a_{nom}(x_k, u_k) = g_{static}(z_k) + w_k \quad (8)$$

For each SISO model, we first set the corresponding GP prior. Since we have no preference for the residual before seeing data, the prior mean function is zero, i.e.,  $\mu(v_{B,i}) = 0$ . We choose the Radial Basis Function (RBF) kernel, which is adept at capturing smooth nonlinear relationships:

$$k(v, v') = \sigma_f^2 \exp\left(-\frac{(v - v')^2}{2l^2}\right) \quad (9)$$

where the hyperparameter set  $\theta = \{\sigma_f^2, l, \sigma_n^2\}$  represents the signal variance, length scale, and noise variance, respectively.

We collect a large-scale offline dataset covering the UAV's entire operating range by executing a series of sufficiently exciting flight trajectories. Subsequently, we optimize the hyperparameters  $\theta$  by maximizing the log marginal likelihood to obtain the GP posterior model that best explains the observed data.

Due to the large size of the offline dataset, the computational cost of using a full GP for online prediction is too high. Therefore, we employ sparse Gaussian process approximation techniques. By introducing a set of inducing points, much smaller in size than  $N$ , we construct a computationally lighter approximate posterior distribution (Ma, 2023). This allows for rapid calculation of the predictive mean during flight, ensuring real-time control.

After offline training is complete, the static residual GP is fixed as a static nonlinear function, which is called for posterior computation in the MPC control. Combined with the nominal model, it forms the static augmented model for controller design:

$$f_{SGP}(x, u) = f_{nom}(x, u) + B_d g_{static}(z(x, u)) \quad (10)$$

### B. Online Adaptive Modeling of Dynamic Residuals

To ensure the efficiency and accuracy of online learning, we must first establish a clean and uncontaminated learning target. If the learning target contains information already modeled by the static GP, it will lead to data contamination and reduced learning efficiency. Therefore, we define the learning target for the dynamic residual,  $y_{dyn,k}$ , as the difference between the observed true acceleration  $a_{real,k}$  at sampling time  $k$  and the acceleration  $a_{SGP}(x_k, u_k)$  predicted by the static GP-augmented model:

$$y_{dyn,k} = a_{real,k} - a_{SGP}(x_k, u_k) = g_{dyn}(z_k) + w'_k \quad (11)$$

In this way, the static GP acts as a pre-processor, stripping away predictable system biases and providing the online GP with a clean signal that isolates the truly time-varying, unknown dynamic disturbances. Similar to the static GP, we model the dynamic residual using three independent SISO GPs with RBF kernels, implemented using GPyTorch to construct a standard GP model.

The performance and long-term stability of an online learning GP within a high-frequency MPC framework are critically dependent on the effective management of the continuous data stream. Conventional methods, such as sliding windows or first-in-first-out (FIFO) queues, are computationally efficient but their uniform forgetting

mechanisms do not differentiate data based on informational content. Consequently, they risk discarding samples containing rare but crucial dynamic information along with outdated data, which can lead to catastrophic forgetting. To address this challenge, we propose an Information-Value-Driven Multi-Level Sparse Buffer strategy.

The core principle of this strategy is that a data point's retention is determined not by its temporal age, but by a metric that quantifies its contribution to the model's predictive accuracy—the Information Value Score (IVS). Our buffer architecture is composed of multiple tiers, each with a fixed capacity and sparsity factor. This tiered structure implements a hierarchical data retention policy: higher-priority tiers utilize smaller capacities and denser sampling to maintain high sensitivity to recent dynamics, while lower-priority tiers use larger capacities and sparser sampling to preserve historically significant data. When any tier's buffer is full, the system discards the data point with the lowest IVS, not the oldest one.

The IVS is formulated to quantify a data point's potential to improve the current GP model by balancing its novelty and recency. we design a more robust weighted-sum formulation:

$$IVS(d_i) = w_{novelty} \cdot V_{novelty}^{norm}(d_i) + w_{recency} \cdot A_{recency}(d_i) \quad (12)$$

where  $d_i$  is the  $i$  data point, and  $w_{novelty}$  and  $w_{recency}$  are weighting coefficients satisfying  $w_{novelty} + w_{recency} = 1$ , providing an explicit and interpretable means to tune their relative importance.

The Novelty Value,  $V_{novelty}$ , quantifies the degree of model uncertainty associated with a data point. Within the Bayesian framework of GPs, the predictive variance  $\sigma^2(z)$  is a natural measure of the model's uncertainty at an input point  $z$ . A larger variance at  $d_i$  implies that the model has less knowledge in that region, making the data point more informative. To eliminate the influence of GP hyperparameters and output scales on the raw variance, we normalize it. At each decision step, we compute the predictive variances for all data points in the current buffer and linearly scale them to the interval  $[0,1]$ :

$$V_{novelty}^{norm}(d_i) = \frac{\sigma^2(z_i) - \min_j(\sigma^2(z_j))}{\max_j(\sigma^2(z_j)) - \min_j(\sigma^2(z_j))} \quad (13)$$

The Recency Weight,  $A_{recency}$ , prioritizes adaptation to recent dynamic changes, which is crucial for handling time-varying disturbances. We employ an exponential decay function where the influence of a data point diminishes over time:

$$A_{recency}(d_i) = \exp\left(-\frac{t_{current} - t_i}{h}\right) \quad (14)$$

where  $t_{current}$  is the current time,  $t_i$  is the timestamp of the data point, and  $h$  is the decay rate hyperparameter, defining the rate of decay.

This weighted-sum formulation of the IVS, through normalization and explicit weighting, ensures a balanced trade-off between the two evaluation dimensions within a unified and stable framework. It allows a data point to be retained for its extremely high novelty or its high recency, thus achieving a superior balance between intelligent forgetting and the retention of critical information.

The data flow management follows a rigorous workflow based on this IVS metric. Each new data point first attempts to enter the highest-priority tier. If this buffer is full, the IVS of all its occupants is calculated, and the new point replaces the one with the lowest score. The displaced point then becomes a candidate for the next tier, where it must pass a sparsity check and again compete based on IVS. This cascaded filtering mechanism, where a data point displaced from a higher-priority tier is re-evaluated for inclusion in the subsequent tier, ensures that the data buffer maintains an information-dense sample set, providing a solid foundation for efficient online GP learning.

Finally, the augmented model proposed in this paper,  $f_{AR}$ , is composed of the static augmented model and the mean prediction of the online dynamic GP. It will be used to construct the risk-aware MPC controller in the next section.

$$f_{AR}(x, u) = f_{SGP}(x, u) + B_d \mu_{dyn}(z(x, u)) \quad (15)$$

Through this intelligent data management strategy, the online GP learner is equipped with an efficient data selection and update mechanism. This enables the model to extract and retain the most valuable information from a continuous data stream while satisfying real-time computational constraints. This provides powerful adaptive capabilities for navigating unknown and time-varying dynamic environments.

#### IV. CAUTIOUS MODEL PREDICTIVE CONTROL FRAMEWORK

Inspired by (Hewing et al., 2019), we construct a framework for robust optimization in the presence of model uncertainty by integrating the GP-based model into the Model Predictive Controller. Since the prediction of the dynamic residual is inherently a probability distribution, the entire system's predictive model becomes stochastic. This section details how this stochastic optimization problem is transformed, through a series of rigorous approximations, into a deterministic Nonlinear Programming (NLP) problem that can be efficiently solved in real-time, ultimately achieving cautious control.

##### A. Stochastic Optimal Control Problem

With the introduction of the online GP, the system's discrete-time state transition equation (7) evolves into a stochastic process, as the dynamic residual term  $g_{dyn}$  is a probability distribution. At each sampling time  $k$ , the controller's task is thus transformed from a deterministic optimization problem into solving a Stochastic Optimal Control Problem (SOCP). Its objective is to minimize the

expected cumulative cost over a prediction horizon  $N$ , which consists of a stage cost  $l(\cdot)$  and a terminal cost  $\phi(\cdot)$ :

$$\begin{aligned} \min_{\{u_{i|k}\}_{i=0}^{N-1}} & \mathbb{E} \left[ \phi(x_{N|k}) + \sum_{i=0}^{N-1} l(x_{i|k}, u_{i|k}) \right] \\ \text{s.t. } & x_{0|k} = x(k) \\ & x_{i+1|k} = f_{SGP}(x_{i|k}, u_{i|k}) + B_d g_{dyn}(z_{i|k}) \\ & u_{i|k} \in \mathcal{U} \end{aligned} \quad (16)$$

where  $x_{i|k}$  and  $u_{i|k}$  represent the predictions for the state and input at future step  $i$ , made at time  $k$ .  $f_{SGP}(\cdot)$  is the deterministic part composed of the nominal model and the static GP.  $g_{dyn}(\cdot)$ , the dynamic GP prediction, is a probability distribution:  $g_{dyn}(z_{i|k}) \sim \mathcal{N}(\mu_{dyn}(z_{i|k}), \Sigma_{dyn}(z_{i|k}))$ . Its input feature  $z_{i|k}$  is taken as the body-frame velocity  $v_{B,i|k}$ . Directly solving this stochastic OCP is generally intractable due to the challenge of propagating probability distributions through nonlinear functions.

##### B. Approximate Probability Distribution Propagation

To render the problem tractable, we adopt a widely used approximation method: we assume that the state at each step within the prediction horizon follows a Gaussian distribution, i.e.,  $x_{i|k} \sim \mathcal{N}(\bar{x}_{i|k}, \Sigma_{i|k}^x)$ . We linearize the dynamics by performing a first-order Taylor expansion of the nonlinear function  $f_{SGP}$  around the current predicted mean point  $(\bar{x}_{i|k}, u_{i|k})$ . This allows us to analytically derive the propagation equations for the state mean and covariance. The propagation of the state mean is determined by both the deterministic dynamics and the mean of the dynamic GP prediction:

$$\bar{x}_{i+1|k} \approx f_{SGP}(\bar{x}_{i|k}, u_{i|k}) + B_d \mu_{dyn}(z(\bar{x}_{i|k})) \quad (17)$$

where  $z(\bar{x}_{i|k})$  represents the GP input feature calculated from the mean state  $\bar{x}_{i|k}$ .

The evolution of the state covariance,  $\Sigma_{i+1|k}$ , reflects the accumulation of uncertainty. Based on the linearization assumption, the covariance at the next time step consists of two parts: the propagation of uncertainty from the previous step, and the new model uncertainty introduced by the dynamic GP at the current step:

$$\Sigma_{i+1|k}^x \approx A_{i|k} \Sigma_{i|k}^x A_{i|k}^T + B_d \Sigma_{dyn}(z(\bar{x}_{i|k})) B_d^T \quad (18)$$

where  $A_{i|k} = \nabla_{\bar{x}} f_{SGP}(\bar{x}_{i|k}, u_{i|k})$  is the Jacobian matrix of the deterministic model  $f_{SGP}$  with respect to the state at the current prediction point. This formula clearly reveals the sources and accumulation process of uncertainty, which is fundamental to achieving cautious control.

##### C. Computable Cautious Control Optimization Problem

After obtaining the propagation methods for the state mean and covariance, we can transform the original

stochastic OCP into a computable deterministic problem. The key to this transformation lies in the expectation calculation of the cost function. For a standard quadratic stage cost  $l(x, u) = \|x - x_{ref}\|_Q^2 + \|u - u_{ref}\|_R^2$ , its expected value is:

$$\mathbb{E}[l(x_{i|k}, u_{i|k})] = \|\bar{x}_{i|k} - x_{ref}\|_Q^2 + \text{tr}(Q\Sigma_{i|k}^x) + \|u_{i|k} - u_{ref}\|_R^2 \quad (19)$$

where  $\text{tr}(\cdot)$  denotes the trace of a matrix. This formula is the core embodiment of cautious control. The total expected cost consists of two parts:

**Performance Cost:**  $\|\bar{x}_{i|k} - x_{ref}\|_Q^2 + \|u_{i|k} - u_{ref}\|_R^2$ , which penalizes the deviation of the mean trajectory from the reference trajectory, aiming for higher tracking performance.

**Uncertainty Cost:**  $\text{tr}(Q\Sigma_{i|k}^x)$ , which is proportional to the variance of the state prediction and quantifies the uncertainty.

This means that when the dynamic GP is more uncertain about its prediction, i.e.,  $\Sigma_{dyn}$  is larger, the future state variance  $\Sigma_x$  will also increase, leading to a higher total cost. To minimize the total cost, the optimizer must trade-off between performance and risk. It will autonomously seek a path that is not only close to the reference trajectory but also has low state uncertainty. This mechanism makes the controller behave more cautiously when facing regions of high model uncertainty, possibly sacrificing some tracking performance for higher safety and robustness.

By substituting this expected cost into the original OCP, we finally obtain a deterministic optimization problem that can be efficiently solved by a standard NLP solver. In this problem, the decision variables are the control sequence  $\{u_{i|k}\}_{i=0}^{N-1}$ :

$$\begin{aligned} \min_{u_{i|k}} & \left( \|\bar{x}_{N|k} - x_{ref,N}\|_P^2 + \text{tr}(P\Sigma_{N|k}^x) \right) + \\ & \sum_{i=0}^{N-1} \left( \|\bar{x}_{i|k} - x_{ref,i}\|_Q^2 + \text{tr}(Q\Sigma_{i|k}^x) + \|u_{i|k} - u_{ref,i}\|_R^2 \right) \quad (20) \\ \text{s.t. } & \bar{x}_{0|k} = x(k), \Sigma_{0|k}^x = 0 \\ & \bar{x}_{i+1|k} = f_{GP}(\bar{x}_{i|k}, u_{i|k}) + B_d \mu_{dyn,i} \\ & \Sigma_{i+1|k}^x = A_{i|k} \Sigma_{i|k}^x A_{i|k}^T + B_d \Sigma_{dyn,i} B_d^T \\ & u_{i|k} \in \mathcal{U} \end{aligned}$$

The predictive mean sequence  $\{\mu_{dyn,i}\}$  and variance sequence  $\{\Sigma_{dyn,i}\}$  of the dynamic GP are computed inside the solver. Before each MPC control cycle, we embed the dynamic GP model as an external function in the model settings of the Acados solver. Therefore, during each iteration within the NLP solver, when it needs to evaluate the dynamics of a candidate trajectory, Acados directly calls this external function. The function receives the current candidate state and input, calculates the dynamic GP's predictive mean  $\mu_{dyn}$  and variance  $\Sigma_{dyn}$  in real-time, and uses them for the mean and covariance propagation calculations. In this way, the optimizer can accurately consider the impact of model

uncertainty throughout its entire solving process and achieve cautious control.

## V. EXPERIMENTS AND RESULT ANALYSIS

To comprehensively evaluate the effectiveness of our proposed Adaptive Risk-aware Model Predictive Control (AR-MPC) framework, this section details a series of rigorous, high-fidelity simulation experiments. We aim to answer the following three core research questions: (i) When faced with unknown, dynamically changing external disturbances, can our proposed AR-MPC significantly improve trajectory tracking accuracy compared to methods that only compensate for static biases? (ii) Can our online learning module, particularly its information-value-driven data management strategy, effectively and rapidly learn and compensate for time-varying disturbances during flight? (iii) While delivering performance improvements, does the computational overhead of our enhanced framework satisfy the requirements of real-time control?

### A. Experimental Setup

The experiments are conducted on a high-fidelity simulation platform developed in Python. The UAV's full six-degree-of-freedom nonlinear dynamics model is numerically integrated using the fourth-order Runge-Kutta method. We use a very small integration step size of  $\Delta t_{sim} = 0.5\text{ms}$  to ensure the numerical accuracy of the dynamic evolution. Additionally, we simulate zero-mean Gaussian noise forces and torques acting on the quadrotor body, as well as asymmetric noise on the motor voltage signals.

To evaluate the controller's robustness to external time-varying disturbances, we designed and implemented a non-stationary wind field model as shown in Fig. 2. This model generates a continuously varying complex wind field in the world frame by superimposing multiple sine waves of different frequencies and amplitudes. To ensure physical realism, this wind vector is first accurately transformed into the body frame based on the UAV's current attitude at each integration step before being applied to the dynamics equations.

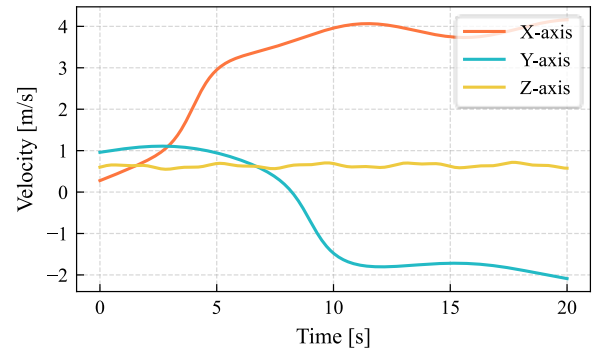


Fig. 2. The simulated time-varying wind profile. The plot shows the wind velocity components along the X, Y, and Z axes over time.

We configured three controllers for comparative evaluation under the same MPC framework by changing their internal predictive models:

- **Nominal MPC:** The state-space expression uses only the nominal dynamics model  $f_{\text{nom}}$  described in Section II.A, without any data-driven compensation. It serves as our baseline for performance evaluation.
- **SGP-MPC:** This represents the baseline method for comparison. The controller uses a static augmented model  $f_{\text{SGP}} = f_{\text{nom}} + B_d g_{\text{static}}(z)$ . The  $g_{\text{static}}(z)$  is obtained by offline training on data collected from various excitation trajectories to compensate for inherent, repeatable aerodynamic biases.
- **AR-MPC:** This is our proposed framework. The controller adopts the cautious control strategy described in Section IV. It uses the static GP for baseline compensation and runs an online adaptive GP module in real-time to estimate dynamic residuals. The module's mean prediction  $\mu_{\text{dyn}}(z)$  is used to augment the model, while its predictive variance  $\Sigma_{\text{dyn}}$  is incorporated into the MPC's cost function. As shown in (19), the penalty term  $\text{tr}(Q\Sigma_t^x)$  incentivizes the controller to actively avoid regions of high model uncertainty, achieving risk-aware optimal control.

All controllers share the same MPC parameters: a prediction horizon of  $T_h = 1.0\text{s}$ , discretized into  $N = 10$  nodes, and a control frequency of 50Hz (i.e., MPC solving period of 20ms). We use the ACADOS software library as the solver for the resulting nonlinear programming problems.

To resolve the real-time conflict between online GP hyperparameter optimization and high-frequency MPC, we designed and implemented an asynchronous learning architecture for the AR-MPC. This architecture decouples the

system into two independent, parallel-running processes: The high-frequency real-time control process executes the MPC control loop. At each solving instance, it uses the currently fixed online GP hyperparameters for rapid mean and variance prediction, ensuring strict real-time performance of the control decisions. Our asynchronous architecture decouples the system into two parallel processes. The high-frequency control process feeds newly collected data into a buffer. Periodically (every 17 data points in our experiments), the low-frequency learning process fetches high-value samples from this buffer to perform computationally intensive hyperparameter optimization. This module uses a three-level sparse data buffer with capacities of [5, 10, 15] and sparsity factors of [2, 4, 6] to balance recent data against long-term memory.

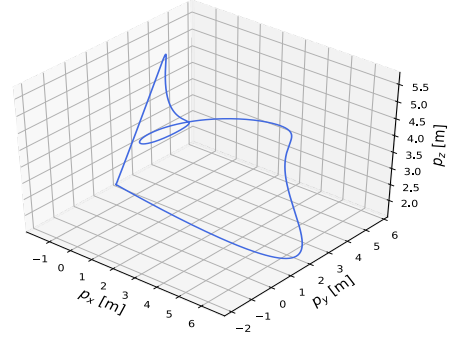


Fig. 3. The 3D reference trajectory used in the experiments. The plot shows the desired path in terms of position coordinates  $(p_x, p_y, p_z)$ .

We selected a highly challenging three-dimensional trajectory, as shown in Fig. 3, which includes high-speed turns, climbs, and dives, as the main evaluation scenario to fully demonstrate the controller's tracking performance. The core performance evaluation metric is the Root Mean Square Error (RMSE) of the position tracking.

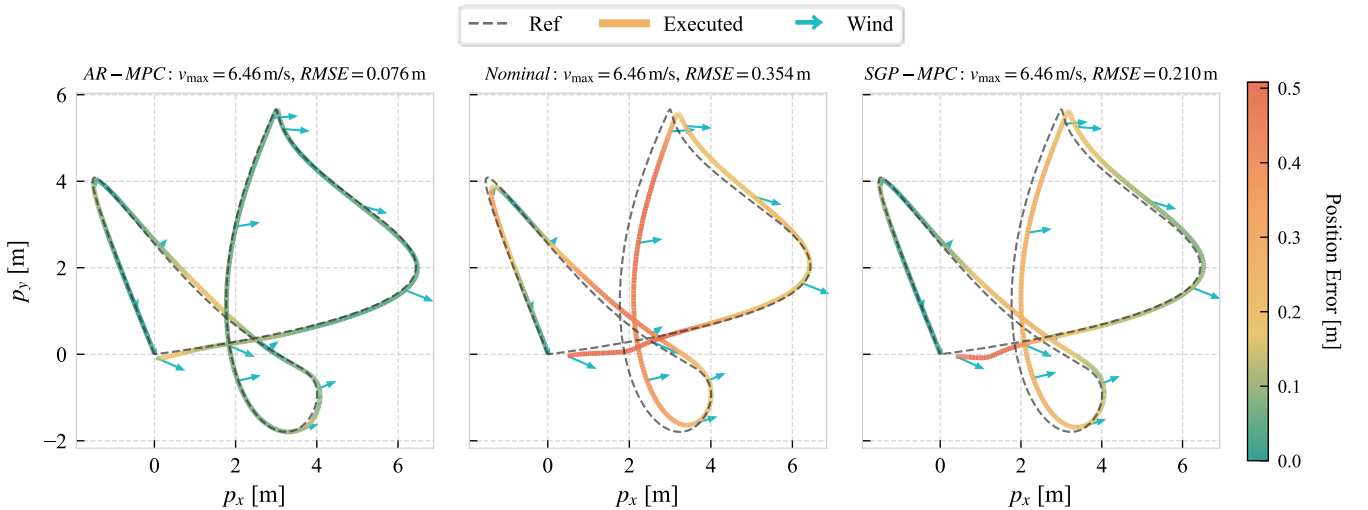


Fig. 4. Comparison of trajectory tracking performance in the XY-plane under time-varying wind disturbances at a maximum speed of 6.46 m/s. The executed trajectory (solid line) is colored by the positional error, with yellow indicating larger errors. The cyan arrows represent the direction and relative magnitude of the wind vector at different points along the trajectory. (Left) Our AR-MPC maintains tight tracking of the reference (dashed line). (Center) The Nominal MPC, without any model compensation, fails to follow the aggressive turns. (Right) The SGP-MPC, which only compensates for static errors, shows significant deviation.

### B. Trajectory Tracking Performance Evaluation

We quantitatively evaluated the trajectory tracking performance of different controllers in a challenging time-varying wind disturbance environment. Fig. 4 visually show the comparison of the executed trajectories with the reference trajectory in the XY plane projection for the three controllers at a maximum speed of 6.46 m/s.

- **AR-MPC:** Our method demonstrates the best performance. Its online GP module successfully learns and compensates for the time-varying wind disturbances that the SGP model cannot explain in real-time. Ultimately, the AR-MPC keeps the trajectory tightly around the reference path, with the tracking error significantly reduced to 0.0706 m.
- **SGP-MPC:** The offline-learned static GP effectively compensates for most of the predictable aerodynamic effects, showing a significant performance improvement over the Nominal MPC. However, its static model cannot adapt to the dynamically changing wind field, thus still showing noticeable tracking errors, with an RMSE of 0.2103 m.
- **Nominal MPC:** As it completely ignores model mismatch and external wind disturbances, it exhibits huge tracking deviations, almost completely deviating from the reference path in the high-speed turning sections, with a final RMSE as high as 0.3539 m.

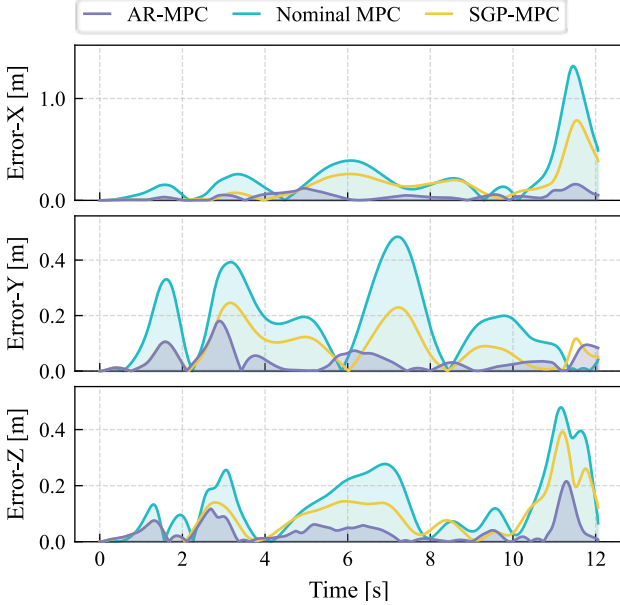


Fig. 5. Positional tracking error over time for each controller, corresponding to the experiment shown in Fig. 4. The plots show the absolute error along the X, Y, and Z axes. The purple, yellow, and cyan lines correspond to the AR-MPC, SGP-MPC, and Nominal MPC, respectively.

To further analyze the dynamic characteristics of the error, Fig. 5 shows the evolution of the three-axis position error over time. It can be clearly seen that the AR-MPC (purple curve)

suppresses the error to a very low level for most of the time. In contrast, the errors of the Nominal MPC (cyan curve) and SGP-MPC (yellow curve) show continuous large fluctuations, which directly reflects their inability to cope with time-varying disturbances. Also, initially, since the AR-MPC has not collected enough information for model fitting, its tracking performance is similar to that of the SGP-MPC.

Fig. 6 further evaluates the robustness of each controller at different maximum flight speeds. The error of the Nominal MPC deteriorates sharply with increasing speed. The performance of the SGP-MPC is relatively stable, but its error remains at a high level, indicating that its compensation capability has reached saturation. Our AR-MPC maintains the lowest error level across the entire speed range.

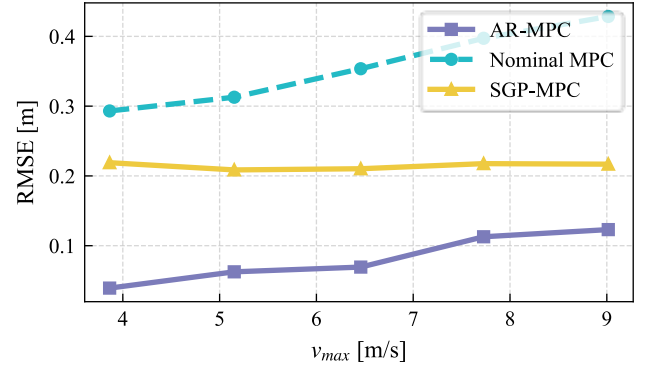


Fig. 6. Position tracking RMSE as a function of maximum flight speed. The purple, yellow, and blue dashed lines show the RMSE for the AR-MPC, SGP-MPC, and Nominal MPC controllers, respectively.

Table I summarizes the quantitative performance of the three control frameworks at different peak flight speeds under the same wind disturbance conditions. This table clearly reveals the superiority of our proposed AR-MPC framework. In all tested speed ranges, the trajectory tracking error of AR-MPC is significantly lower than the SGP-MPC benchmark.

TABLE I. QUANTITATIVE COMPARISON OF TRACKING PERFORMANCE

$v_{\text{peak}}$ [m/s]	Model		
	<i>Nominal MPC</i> RMSE [m]	<i>SGP-MPC</i> RMSE [m]	<i>AR-MPC (Ours)</i> RMSE [m]
3.86	0.2932	0.2192	0.0356
5.15	0.3128	0.2087	0.0624
6.46	0.3539	0.2103	0.0760
7.73	0.3974	0.2176	0.1077
9.02	0.4290	0.2178	0.1214

It is particularly noteworthy that the performance improvement brought by our AR-MPC framework is especially significant at low speeds. At a peak speed of about 3.9 m/s, the tracking error is reduced by nearly 83.5% compared to SGP-MPC. As the speed increases, although the absolute error increases, AR-MPC always maintains the

lowest error level, achieving a performance improvement of over 44% even at a peak speed of 9.0 m/s. This result powerfully demonstrates the superior performance of our hierarchical decoupled learning framework and its robustness in dynamic, unknown environments.

### C. Online Learning Performance

To investigate the model regression effect of the online learning model within the AR-MPC framework, Fig. 7 shows a visualization analysis of the learning state of the dynamic residual GP at the 8th second of the simulation.

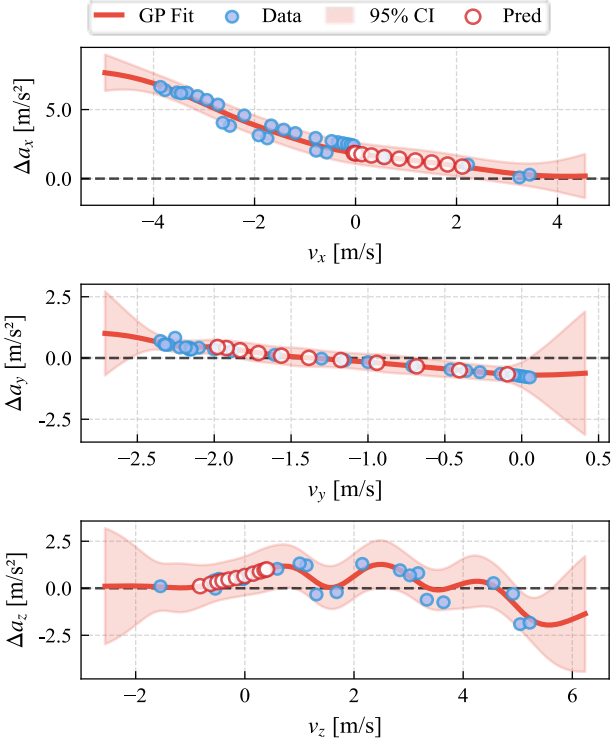


Fig. 7. The learned dynamic residual model at  $t = 8s$ . The plots show the mapping from body velocity ( $v_x, v_y, v_z$ ) to acceleration residuals ( $\Delta a_x, \Delta a_y, \Delta a_z$ ). The red line is the GP mean prediction, the shaded area is the 95% confidence interval, blue dots are the training data points, and open red circles are the MPC predictions over its horizon.

The GP's mean prediction curve accurately passes through the dense regions of the training data points selected by the data buffer, indicating that it has successfully captured the nonlinear relationship between the body velocity and the unmodeled acceleration residual. In regions where data points are sparse, the GP's 95% confidence interval expands significantly. A larger confidence interval increases the magnitude of the uncertainty penalty term in the cost function. This incentivizes the optimizer to generate trajectories that avoid regions of high predictive uncertainty, favoring control actions for which the model is more confident and thereby promoting safer, more reliable flight. Our asynchronous learning architecture can continuously learn from flight data without interrupting the main control loop and provide critical risk assessment information for cautious control.

Fig. 8 shows the analysis of computational efficiency. Despite the introduction of the online learning module, the average optimization time of the entire AR-MPC control loop remains stable at around 5ms, meeting the 50Hz real-time control requirement. Although slightly higher than the optimization times of SGP-MPC (approx. 1.8ms) and Nominal MPC (approx. 1ms), this increase in computational cost is entirely acceptable given the huge performance improvement it brings, illustrating that our method strikes a good balance between theoretical innovation and engineering practicality.

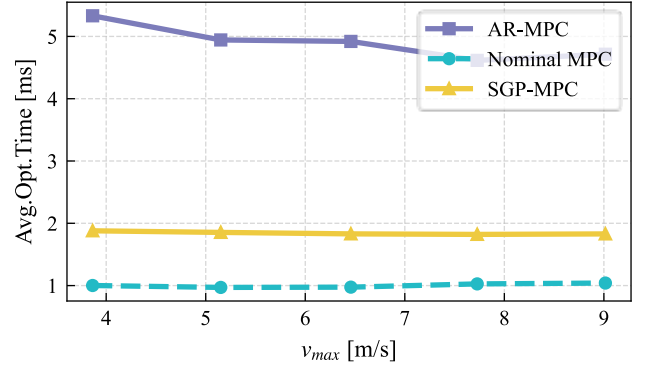


Fig. 8. Mean optimization time per control step as a function of maximum flight speed. The purple, yellow, and blue dashed lines show the computation time for the AR-MPC, SGP-MPC, and Nominal MPC controllers, respectively.

## VI. CONCLUSION

This paper proposed an Adaptive Risk-aware Model Predictive Control (AR-MPC) framework for high-precision UAV trajectory tracking in unknown dynamic environments. The core of this framework lies in a novel online adaptive learning and decision-making mechanism, which features an information-value-driven multi-level data management strategy. By intelligently assessing the novelty and recency of data points, this strategy ensures that the online GP learning module can efficiently utilize data while avoiding catastrophic forgetting. Concurrently, an asynchronous learning architecture decouples the computationally intensive optimization task from the high-frequency control loop. By constructing a risk-sensitive optimization framework that internalizes the uncertainty from online learning into the controller's decision-making, a trade-off between performance and risk is achieved.

High-fidelity simulation experiments demonstrated that, in complex time-varying wind fields, the AR-MPC can reduce trajectory tracking error by up to 44% compared to a static GP-MPC baseline, with its computational overhead fully meeting the 50Hz real-time control requirement. This validates the excellent balance our method strikes a balance between theoretical innovation and engineering practicality.

Future work can further explore more advanced uncertainty-aware control strategies based on this foundation,

such as actively avoiding hazardous areas with obstacles, to achieve safer and more intelligent autonomous flight.

## REFERENCES

- Agha-mohammadi, A.-a., Ure, N. K., How, J. P., & Vian, J. (2014). Health aware stochastic planning for persistent package delivery missions using quadrotors. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems,
- Choo, W., & Kayacan, E. (2023, 2023-05-31). Computationally Efficient Data-Driven MPC for Agile Quadrotor Flight. 2023 American Control Conference (ACC),
- Gedefaw, E. A., Abera, N. B., & Abdissa, C. M. (2025). A Review of Modeling and Control Techniques for Unmanned Aerial Vehicles. *Engineering Reports*, 7(6), e70215.
- Hanover, D., Foehn, P., Sun, S., Kaufmann, E., & Scaramuzza, D. (2021). Performance, precision, and payloads: Adaptive nonlinear mpc for quadrotors. *IEEE Robotics and Automation Letters*, 7(2), 690-697.
- Hewing, L., Kabzan, J., & Zeilinger, M. N. (2019). Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, 28(6), 2736-2743.
- Hewing, L., Wabersich, K. P., Menner, M., & Zeilinger, M. N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1), 269-296.
- Kulathunga, G., Hamed, H., & Klimchik, A. (2024). Residual dynamics learning for trajectory tracking for multi-rotor aerial vehicles. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-51822-0>
- Langston, D. (2024). A Comprehensive Review of Trajectory Tracking Control Technologies for Rotary-Wing UAVs: Challenges and Prospects. *Journal of Computer Science and Software Applications*, 4(8), 1-5.
- Liu, Y., & Toth, R. (2021, 2021-12-14). Learning Based Model Predictive Control for Quadcopters with Dual Gaussian Process. 2021 60th IEEE Conference on Decision and Control (CDC),
- Liu, Y., Wang, P., & Tóth, R. (2022). Learning for predictive control: A dual gaussian process approach. *arXiv preprint arXiv:2211.03699*.
- Ma, T. (2023). Constrained tracking control of stochastic multivariable nonlinear systems via Gaussian process predictions. *International Journal of Control*, 96(11), 2787-2798. <https://doi.org/10.1080/00207179.2022.2112622>
- Martinez-Vasquez, A. H., Castro-Linares, R., & Sira-Ramirez, H. An equivalence of ADRC and flat filters: an application to a quadrotor UAV with a spherical inverted or suspended pendulum. *International Journal of Control*, 1-22. <https://doi.org/10.1080/00207179.2025.2489564>
- Miranda-Colorado, R., Domínguez, I., & Aguilar, L. T. (2024). Variable-gain sliding mode control for quadrotor vehicles: Lyapunov-based analysis and finite-time stability. *International Journal of Control*, 97(11), 2565-2583. <https://doi.org/10.1080/00207179.2023.2285412>
- Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning* (pp. 63-71). Springer.
- Salzmann, T., Kaufmann, E., Arrizabalaga, J., Pavone, M., Scaramuzza, D., & Ryll, M. (2023). Real-time neural mpc: Deep learning model predictive control for quadrotors and agile robotic platforms. *IEEE Robotics and Automation Letters*, 8(4), 2397-2404.
- Schmid, N., Gruner, J., Abbas, H. S., & Rostalski, P. (2022, 2022-06-08). A real-time GP based MPC for quadcopters with unknown disturbances. 2022 American Control Conference (ACC),
- Torrente, G., Kaufmann, E., Fohn, P., & Scaramuzza, D. (2021). Data-Driven MPC for Quadrotors. *IEEE Robotics and Automation Letters*, 6(2), 3769-3776. <https://doi.org/10.1109/lra.2021.3061307>
- Ventura Diaz, P., & Yoon, S. (2018). High-fidelity computational aerodynamics of multi-rotor unmanned aerial vehicles. 2018 AIAA Aerospace Sciences Meeting,
- Wei, M., Li, M., Zou, B., & Zheng, L. (2024, 2024-06-21). Online Learning-based Model Predictive Control for Quadrotors Under Uncertain Wind Disturbances. Proceedings of the 2024 4th International Conference on Control and Intelligent Robotics,
- Zhu, J., Li, Y., Wen, G., Lee, S., & Veluvolu, K. C. Neural network-based simplified optimised backstepping control for attitude tracking of a 2-DOF helicopter system. *International Journal of Control*, 1-16. <https://doi.org/10.1080/00207179.2025.2540542>