Radar-Avoidant Quadrotor Flight via RRT-Guided LPV–MPC with CBFs, Feedback Linearization, and Integral-Terminal- Adaptive Sliding mode control

1st Dat Vu Tien
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City (VNU-HCM)
Ho Chi Minh City, Viet Nam
dat.vuv@hcmut.edu.vn

2nd Phuong Tung Pham
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City (VNU-HCM)
Ho Chi Minh City, Viet Nam
pptung@hcmut.edu.vn

Abstract: *The paper presents a navigation architecture for quadcopter radar evasion in environments where ground-based radar stations/coverage zones are known in advance from mission maps. At the planning layer, we employ Rapidly-Exploring Random Tree Star (RRT★) on a "radar exposure" cost field (assuming known coverage radii), generating waypoints from which a cubic spline trajectory is constructed. The time parameterization and discretization of the spline are carefully handled to avoid instability at junction nodes. At the control layer, the trajectory is tracked by an LPV–MPC, in which a Control Barrier Function (CBF) acts as a safety filter at the output, preventing control commands from driving the UAV into radar detection zones. Simultaneously, in a cascade architecture, the position channel is made robust through a Feedback Linearization branch (decoupling and nominal linearization of translational dynamics) combined with Adaptive Sliding-Mode Control (compensating for model mismatch and external disturbances), ensuring stable trajectory tracking under uncertainties. The proposed method is validated in MATLAB simulations with multiple radar layouts and terrain occlusions, demonstrating accurate tracking, strict radar-evasion compliance, and clear advantages over traditional controllers (PID, LQR, …).*

Keywords—LPV-MPC; Control barrier function; Adaptive control;Adaptive sliding mode control; safe control; quadrotor/UVA; Real-time optimization, collision avoidance.

I. INTRODUCTION

Unmanned aerial vehicle (UAV) missions in radar-surveilled areas impose two key requirements: (i) trajectories and control inputs must satisfy dynamic and actuator constraints, and (ii) radar detection probability must be minimized. This work assumes that radar positions are known a priori from mission maps or intelligence sources, treating them as external threats. The main challenge lies in embedding radar-evasion constraints—represented by exposure or SNR fields—into both trajectory planning and real-time control under the limited computational resources of small UAVs.

Related works have addressed UAV navigation and control via classical PID, LQR, or Sliding Mode Control (SMC) laws, which offer satisfactory nominal performance but lack mechanisms to systematically enforce input/state or safety constraints. Model Predictive Control (MPC) provides a principled approach to constraint handling through finite-horizon optimization. However, direct nonlinear MPC is computationally demanding and prone to infeasibility on embedded UAV hardware. Simplified global linearization reduces complexity but compromises model fidelity and robustness. Recent hybrid strategies—such as LPV-based MPC formulations and CBF-augmented control—have improved real-time feasibility and safety certification, yet few integrate these with robust adaptive outer loops for mission-level radar avoidance.

To address this gap, We propose a unified two-layer planning – control framework for real-time radar avoidance. At the planning layer, radar zones are mapped into an exposure-based cost field where RRT★ generates safe waypoints smoothed into time-parameterized splines. The outer control layer combines Feedback Linearization with an Integral–Terminal Adaptive Sliding Mode Controller (IT-ASMC) for robust, finite-time tracking. A Control Barrier Function (CBF) acts as a safety filter, ensuring forward invariance of radar-free zones. The inner loop employs a Linear Parameter-Varying MPC (LPV–MPC) for high-frequency attitude regulation with guaranteed feasibility and actuator constraint handling.

This architecture integrates safety, robustness, and real-time feasibility. The main contributions of this work are: (i) a unified radar-evasion framework combining planning and control, (ii) an exposure-based RRT★ planner with spline time-parameterization, (iii) an LPV–MPC inner loop ensuring constraint-satisfied attitude regulation, and (iv) an adaptive IT-ASMC outer loop with CBF-based safety filtering for finite-time, disturbance-rejected trajectory tracking.

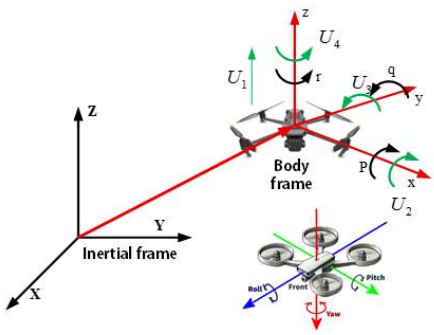

Figure 1 Dynamic reference frame

II. QUADCOPTER DYNAMIC MODEL

Quadcopter dynamic equations are built in the body frame for Attitude control strategy and built in the inertial frame for external position control layer. The above dynamic equation is constructed using the Newton-Euler method. In this context $[X \quad Y \quad Z]$ represents the position vector of the UAV in the inertia frame. $[\phi \quad \theta \quad \psi]$ are the Euler angles describing the orientation of the body frame relative to the inertia frame, also referred to as roll, pitch, and yaw. $[p \quad q \quad r]$ are the angular velocity components (rad/s) of the body frame relative to the inertia frame, expressed in the body frame, $[u \quad v \quad w]$ are the linear velocity components of the UAV in the body frame, and $g$is the gravitational acceleration. The inertia tensor of the UAV with respect to its body axes is simplified to a diagonal matrix $I = diag(I_{xx} \quad I_{yy} \quad I_{zz})$. Ω is the sum of the velocities of the motors, i.e., $\Omega = \Omega_1 + \Omega_3 - \Omega_2 - \Omega_4$. $J_{TP} = I(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)$ is the total angular momentum of the UAV's rotors. The control inputs of the UAV $[U_1 \quad U_2 \quad U_3 \quad U_4]$ defined as in Fig. 1 are:

$$\begin{cases} U_1 = C_T(\Omega_1^{\,2} + \Omega_2^{\,2} + \Omega_3^{\,2} + \Omega_4^{\,2}) \\ U_2 = C_{T.L}(\Omega_2^{\,2} - \Omega_4^{\,2}) \\ U_3 = C_{T.L}(\Omega_3^{\,2} - \Omega_1^{\,2}) \\ U_4 = C_Q(\Omega_1^{\,2} + \Omega_2^{\,2} - \Omega_3^{\,2} - \Omega_4^{\,2}) \end{cases} \quad (1)$$

From (9) it follows that vector $\vec{\Omega}$ represents the angular velocities of the corresponding motors. In this paper, the dynamic equations used to model the UAV motion are derived accordingly. The dynamic equations are discretized with respect to the sampling time T. The solution method adopted in this work is based directly on the Euler method.
The dynamic equations in the body frame is obtain as follows.

$$\dot{u} = vr - wr + g\sin(\theta) \quad (2)$$

$$\dot{v} = wp - ur - g\cos(\theta)\sin(\phi) \quad (3)$$

$$\dot{w} = uq - vp - g\cos(\theta)\cos(\phi) \quad (4)$$

$$\dot{p} = \left[(I_{yy} - I_{zz})qr - J_{TP}q\Omega + U_2 \right] / I_{xx} \quad (5)$$

$$\dot{q} = \left[(I_{zz} - I_{xx})pr - J_{TP}p\Omega + U_3 \right] / I_{yy} \quad (6)$$

$$\dot{r} = \left[(I_{xx} - I_{yy})pq + U_3 \right] / I_{zz} \quad (7)$$

The dynamic equations in the inertial frame is described as follows:

$$\ddot{X} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)U_1 / m \quad (8)$$

$$\ddot{Y} = (\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)U_1 / m \quad (9)$$

$$\ddot{Z} = -g + \cos\phi\cos\theta U_1 / m \quad (10)$$

III. CONTROLLER DESIGN

3.1. Control structure

In UAV control or flight control systems in general, the control tasks can be divided into two main categories: (i) **attitude control** of the UAV, and (ii) **position control** of the UAV. Assuming that the reference trajectory is predefined, a position controller (velocity, displacement) is employed to generate commands in the inertial frame. The outcome of this process is then transmitted to the attitude controller, which transforms them into attitude/thrust commands for the UAV. The overall control process as well as the details of each control block are presented as follows:

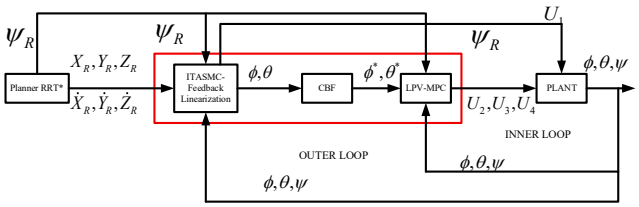

Figure 2 Control architecture block diagram

The cascade architecture separates the fast LPV–MPC inner loop and the IT–ASMC–FBL outer loop, ensuring robustness and constraint satisfaction.

3.2. Attitude control

For attitude control, the UAV is controlled in the body frame. The main state variables to be controlled are the Euler angles. Based on (4), (5), and (6), the differential equations can be expressed as:

$$\dot{p} = \frac{(I_{yy} - I_{zz})}{I_{xx}}qr - \frac{J_{TP}}{I_{xx}}q\Omega + \frac{U_2}{I_{xx}} \quad (11)$$

$$\dot{q} = \frac{(I_{zz} - I_{xx})}{I_{yy}}pr - \frac{J_{TP}}{I_{yy}}p\Omega + \frac{U_3}{I_{yy}} \quad (12)$$

$$\dot{r} = \frac{(I_{xx} - I_{yy})}{I_{zz}}pq + \frac{U_3}{I_{zz}} \quad (13)$$

In UAV control, attitude is regulated through the following dynamic equations:

$$\begin{cases} \ddot{\phi} = \dot{p} = \dfrac{(I_{yy} - I_{zz})}{I_{xx}}qr - \dfrac{J_{TP}}{I_{xx}}q\Omega + \dfrac{U_2}{I_{xx}} & (14) \\ \ddot{\theta} = \dot{q} = \dfrac{(I_{zz} - I_{xx})}{I_{yy}}pr - \dfrac{J_{TP}}{I_{yy}}p\Omega + \dfrac{U_3}{I_{yy}} & (15) \\ \ddot{\psi} = \dot{r} = \dfrac{(I_{xx} - I_{yy})}{I_{zz}}pq + \dfrac{U_4}{I_{zz}} & (16) \end{cases}$$

From (14), (15), and (16), the affine state-space equations are obtained as:

$$\begin{cases} \dot{\vec{x}} = A(\sigma)\vec{x} + B(\sigma)\vec{u} \\ \vec{y} = C\vec{x} \end{cases} \xrightarrow{Euler} \begin{cases} \vec{x}_{k+1} = A(\sigma)\vec{x}_k + B(\sigma)\vec{u}_k \\ \vec{y}_k = C\vec{x}_k \end{cases} \quad (17)$$

To ensure stable control, we redefine the control input as the deviation from the nominal input. Specifically, instead of directly controlling $\vec{u}_k$ we define $\Delta\vec{u}_k$ such that: $\vec{u}_k = \vec{u}_{k-1} + \Delta\vec{u}_k$. Using a state augmentation technique, the dynamics can be expressed as:

$$\begin{cases} \vec{x}_{k+1} = A(\sigma)\vec{x}_k + B(\sigma)\vec{u}_k \\ \vec{u}_k = \vec{u}_{k-1} + \Delta\vec{u}_k \\ \vec{y}_k = C\vec{x}_k \end{cases} \rightarrow \begin{cases} \vec{x}_{k+1} = A(\sigma)\vec{x}_k + B(\sigma)(\vec{u}_{k-1} + \Delta\vec{u}_k) \\ \vec{u}_k = \vec{u}_{k-1} + \Delta\vec{u}_k \\ \vec{y}_k = C\vec{x}_k \end{cases}$$

$$\rightarrow \begin{cases} \underbrace{\begin{bmatrix} \vec{x}_{k+1} \\ \vec{u}_k \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} A(\sigma) & B(\sigma) \\ 0 & I \end{bmatrix}}_{\tilde{A}(\sigma)} \underbrace{\begin{bmatrix} \vec{x}_k \\ \vec{u}_{k-1} \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} B(\sigma) \\ I \end{bmatrix}}_{\tilde{B}(\sigma)} \Delta\vec{u}_k \\ \vec{\tilde{y}}_k = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{\tilde{C}} \begin{bmatrix} \vec{x}_k \\ \vec{u}_{k-1} \end{bmatrix} \end{cases} \rightarrow \begin{cases} \tilde{x}_{k+1} = \tilde{A}(\sigma)\tilde{x}_k + \tilde{B}(\sigma)\Delta\vec{u}_k \\ \vec{\tilde{y}}_k = \tilde{C}\tilde{x}_k \end{cases} \quad (18)$$

*a) **Model predictive control for tracking problem.***

At each sampling instant, the controller optimizes a sequence of **incremental control inputs** $\{\Delta u_k\}_{k=0}^{N-1}$ over the discrete-time LPV model to track the reference trajectory generated by the planner. The tracking error is defined as

$$\vec{e}_k = \vec{y}_k - \vec{y}_k^{ref}, \vec{y}_k = \tilde{C}\vec{x}_k$$

where $\vec{y}_k^{ref}$ comes from the time-parameterized spline. The cost function penalizes both the stage tracking error and the terminal error.

$$J = \frac{1}{2}\vec{e}_4^T Q_f \vec{e}_4 + \frac{1}{2}\sum_{k=0}^{3}(\vec{e}_k^T Q \vec{e}_k + \Delta\vec{u}_k^T R \Delta\vec{u}_k) \quad (19)$$

$$\text{s.t} \begin{cases} \tilde{x}_{k+1} = \tilde{A}(\sigma)\tilde{x}_k + \tilde{B}(\sigma)\Delta\vec{u} \\ \vec{\tilde{y}}_k = \tilde{C}\tilde{x}_k \\ \Delta U_{G\min} \le \Delta U_G \le \Delta U_{G\max} \end{cases}$$

We denote J' as the transformed and simplified version of the original cost function J, yielding an equivalent QP formulation with the same optimal solution but reduced computational complexity.

$$J' = \Delta U_G^T \underbrace{(G^T H G + R)}_{\bar{T}} \Delta\vec{U}_G + 2\begin{bmatrix} \vec{x}_0^T & \vec{R}_G^T \end{bmatrix} \underbrace{\begin{bmatrix} \hat{A}^T H G \\ \vec{R}_G^T \end{bmatrix}}_{\bar{F}} \vec{U}_G \quad (20)$$

In which:

$$G = \begin{bmatrix} \tilde{B} & 0 & 0 & 0 \\ \tilde{A}\tilde{B} & \tilde{B} & 0 & 0 \\ \tilde{A}^2\tilde{B} & \tilde{A}\tilde{B} & \tilde{B} & 0 \\ \tilde{A}^3\tilde{B} & \tilde{A}^2\tilde{B} & \tilde{A}\tilde{B} & \tilde{B} \end{bmatrix},$$

$$H = \begin{bmatrix} \tilde{C}^T Q \tilde{C} & 0 & 0 & 0 \\ 0 & \tilde{C}^T Q \tilde{C} & 0 & 0 \\ 0 & 0 & \tilde{C}^T Q \tilde{C} & 0 \\ 0 & 0 & 0 & -\tilde{C}^T Q_f \tilde{C} \end{bmatrix}$$

$$F = \begin{bmatrix} Q\tilde{C} & 0 & 0 & 0 \\ 0 & Q\tilde{C} & 0 & 0 \\ 0 & 0 & Q\tilde{C} & 0 \\ 0 & 0 & 0 & Q_f\tilde{C} \end{bmatrix}, R = \begin{bmatrix} R & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ 0 & 0 & R & 0 \\ 0 & 0 & 0 & R \end{bmatrix}$$

We then have:

$$\Delta\vec{U}^* = \underbrace{\arg\min}_{\Delta\vec{U}} J = \underbrace{\arg\min}_{\Delta\vec{U}} (\Delta U_G^T \bar{T} \Delta\vec{U}_G + 2\begin{bmatrix} \vec{x}_0^T & \vec{R}_G^T \end{bmatrix} \bar{F}\vec{U}_G)$$

$$\text{s.t } \Delta U_{G\min} \le \Delta U_G \le \Delta U_{G\max}$$

$\vec{U}^* = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T$ only u_1 is applied to the plant; the values u_2, u_3, u_4 are not used. The procedure is repeated over the entire process (receding horizon).

Up to this point, after the above transformations, the objective function has been cast into a QP form. The key strength of MPC is the ability to impose constraints. In the next part we construct problem constraints in a form compatible with the QP formulation.

*b) **Constrain construction***

We have:

$$\Omega_{1,\min} = \Omega_{2,\min} = \Omega_{2,\min} = \Omega_{2,\min} = \Omega_{\min}$$

$$\Omega_{1,\max} = \Omega_{2,\max} = \Omega_{2,\max} = \Omega_{2,\max} = \Omega_{\max}$$

From (1) then we have:

$$\begin{cases} U_{1,\min} = 4C_T\Omega_{\min}^2, U_{1,\max} = 4C_T\Omega_{\max}^2 \\ U_{2,\min} = C_{T.L}(\Omega_{\min}^2 - \Omega_{\max}^2), U_{2,\max} = C_{T.L}(\Omega_{\max}^2 - \Omega_{\min}^2) \\ U_{3,\min} = C_{T.L}(\Omega_{\min}^2 - \Omega_{\max}^2), U_{3,\max} = C_{T.L}(\Omega_{\max}^2 - \Omega_{\min}^2) \\ U_{4,\min} = C_Q(-2\Omega_{\max}^2 + 2\Omega_{\min}^2), U_{4,\max} = C_Q(-2\Omega_{\min}^2 + \Omega_{\max}^2) \end{cases}$$

Since the LPV-MPC regulates only U_2, U_3, U_4 we design bounds for these variables.

We have: $\vec{U}_{G,\min} \le U_G \le U_{G,\max}$ with

$\vec{U}_G = \begin{bmatrix} U_{G,1} & U_{G,2} & U_{G,3} \end{bmatrix}$

In wich $\vec{U}_{G,k} = \begin{bmatrix} U_{2,k} & U_{3,k} & U_{4,k} \end{bmatrix}, k \in \{1,2,3,4\}$

We have:

$$\begin{bmatrix} U_{2,k} \\ U_{3,k} \\ U_{4,k} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\tilde{C}^*} \tilde{X}_k$$

$$\tilde{X}_k = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & U_{2,k} & U_{3,k} & U_{4,k} \end{bmatrix}$$

$$\vec{Y}_G^* = \underbrace{\begin{bmatrix} \tilde{C}^* & 0 & 0 & 0 \\ 0 & \tilde{C}^* & 0 & 0 \\ 0 & 0 & \tilde{C}^* & 0 \\ 0 & 0 & 0 & \tilde{C}^* \end{bmatrix}}_{\tilde{C}_G^*} \begin{bmatrix} \vec{\tilde{X}}_1 \\ \vec{\tilde{X}}_2 \\ \vec{\tilde{X}}_3 \\ \vec{\tilde{X}}_4 \end{bmatrix}$$

$\vec{Y}_G^* = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}, u_1 = \begin{bmatrix} U_{2,1} & U_{3,1} & U_{4,1} \end{bmatrix}$

$u_2 = \begin{bmatrix} U_{2,2} & U_{3,2} & U_{4,2} \end{bmatrix}, u_3 = \begin{bmatrix} U_{2,3} & U_{3,3} & U_{4,3} \end{bmatrix}$

$u_4 = \begin{bmatrix} U_{2,4} & U_{3,4} & U_{4,4} \end{bmatrix}$

From (15) and: $\vec{Y}_G^* = \tilde{C}_G^* \vec{X}_G = \tilde{C}_G^*(G\Delta\vec{U} + \hat{A}\vec{x}_0)$

$\rightarrow \vec{Y}_G^* = \tilde{C}_G^* G \Delta\vec{U} + \tilde{C}_G^* \hat{A}\vec{x}_0$

We have: $\vec{Y}_{G,\min}^* \le \vec{Y}_G^* \le \vec{Y}_{G,\max}^* \rightarrow \begin{cases} \vec{Y}_G^* \le \vec{Y}_{G,\max}^* \\ -\vec{Y}_G^* \le \vec{Y}_{G,\min}^* \end{cases}$

$$\begin{cases} \vec{Y}_G^* \le \vec{Y}_{G,\max}^* \\ -\vec{Y}_G^* \le \vec{Y}_{G,\min}^* \end{cases} \rightarrow \begin{cases} \tilde{C}_G^* G\Delta\vec{U} + \tilde{C}_G^* \hat{A}\vec{x}_0 \le \vec{Y}_{G,\max}^* \\ -(\tilde{C}_G^* G\Delta\vec{U} + \tilde{C}_G^* \hat{A}\vec{x}_0) \le \vec{Y}_{G,\min}^* \end{cases}$$

$$\rightarrow \begin{cases} \tilde{C}_G^* G\Delta\vec{U} \le (\vec{Y}_{G,\max}^* - \tilde{C}_G^* \hat{A}\vec{x}_0) \\ -\tilde{C}_G^* G\Delta\vec{U} \le (\vec{Y}_{G,\min}^* + \tilde{C}_G^* \hat{A}\vec{x}_0) \end{cases}$$

$$\rightarrow \underbrace{\begin{bmatrix} \tilde{C}_G^* G \\ -\tilde{C}_G^* G \end{bmatrix}}_{\tilde{G}^*} \Delta\vec{U} \le \underbrace{\begin{bmatrix} \vec{Y}_{G,\max}^* - \tilde{C}_G^* \hat{A}\vec{x}_0 \\ \vec{Y}_{G,\min}^* + \tilde{C}_G^* \hat{A}\vec{x}_0 \end{bmatrix}}_{h} \rightarrow \tilde{G}^* \Delta\vec{U} \le h$$

*c) **Solution approach for the constrained quadratic program***

After reformulating the tracking constraints into the compact inequality

$$\tilde{G}^* \Delta\vec{U} \le h$$

the MPC law at each sampling instant reduces to a standard quadratic program of the form.

$$J^{'} = \Delta U_G^T \underbrace{(G^T HG + R)}_{\tilde{T}} \Delta\vec{U}_G + 2\begin{bmatrix} \vec{x}_0^T & \vec{R}_G^T \end{bmatrix} \underbrace{\begin{bmatrix} \hat{A}^T HG \\ \vec{R}_G^T \end{bmatrix}}_{\tilde{F}} \vec{U}_G \quad (19)$$

$$\text{s.t. } \tilde{G}^* \Delta\vec{U} \le h$$

This convex formulation allows the use of well-established numerical methods such as active-set or interior-point algorithms. In practice, only the first control increment of the optimal sequence is applied to the plant, and the procedure is repeated in a receding-horizon fashion. The QP structure is sparse and block-banded, making it suitable for efficient real-time solvers.

3.2. **Position control**

The position controller is designed based on the translational dynamics, modeled in the inertial frame as:

$$\begin{cases} \ddot{X} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\dfrac{U_1}{m} = v_1 & (20) \\ \ddot{Y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\sin\psi)\dfrac{U_1}{m} = v_2 & (21) \\ \ddot{Z} = -g + \cos\phi\cos\theta\dfrac{U_1}{m} = v_3, g = 9.8(m.s^2) & (22) \end{cases}$$

Since modeling errors and environmental disturbances are unavoidable, we further consider the effect of disturbances in the controller design. Noise vector assumption $d = \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix}^T$ which represents matched and bounded disturbance. Then we have:

$$\begin{cases} \ddot{X} == v_1 + d_1 \\ \ddot{Y} == v_1 + d_2 \\ \ddot{Z} == v_3 + d_3 \end{cases} \quad (23)$$

Feedback Linearization

Consider a nonlinear dynamic system of the form: $\dot{x} = f(x) + g(x)u + d$ where $d$denotes matched disturbances. The system output is defined as: $y = h(x)$ is the output of the system.

The Derivative of y given by:

$$\frac{\partial y}{\partial t} = \frac{\partial h}{\partial x}\dot{x} = \frac{\partial h}{\partial x}(f(x) + g(x)u) = \frac{\partial h}{\partial x}f(x) + \frac{\partial h}{\partial x}g(x)u$$

Where $L_f h(x) = \dfrac{\partial h}{\partial x} f(x)$ is the Lie detivative of h with respect to f:

(i) if $L_g h(x) = \dfrac{\partial h}{\partial x} g(x) = 0$ then $\dot{y} = \dfrac{\partial h}{\partial x} f(x)$ is independent of u

(ii) Higher-order derivatives can be computed as:

$$L_g L_f h(x) = \frac{\partial (L_f h)}{\partial x} g(x), L^2{}_f h(x) = L_f (L_f h(x))$$

The Lie derivative measures how a vector field or function changes along the flow of another vector field.

If $L_g L^{i-1}{}_f h(x) = 0, i = 1,2,3,..., \rho - 1$ and $L_g L^{\rho-1}{}_f h(x) \neq 0$ then: $y^{\rho} = L_f^{\rho} h(x) + L_g L_f^{\rho-1} h(x) u$

$u = \dfrac{1}{L_g L_f^{\rho-1} h(x)}(-L_f^{\rho} h(x) + v)$, where ρ is the relative degree of the system.

According to (23), when the relative degree of the system equals the number of outputs to be tracked, feedback linearization can be applied without considering the zero dynamics. In this case, the outputs are directly controlled by the virtual input v, and the system errors are mapped into two strictly linear subsystems. However, in the presence of matched disturbances and modeling uncertainties, pure feedback linearization may still yield steady-state errors. Therefore, the dynamic controller is combined with the Integral–Terminal Adaptive Sliding Mode Control (IT-ASMC) technique in Section (b) to enhance robustness and guarantee finite-time convergence.

3.3. **Intergral Terminal Adaptive Sliding Mode control**

The IT-ASMC controller is applied to eliminate instantaneous errors. For nonlinear dynamic systems with bounded matched disturbances, yet without prior knowledge of the disturbance bounds, an Adaptive SMC (ASMC) law is employed. In addition, we propose combining Terminal SMC (TSMC) with Integral SMC (ISMC) in order to design both the reaching phase and a sliding manifold that guarantees finite-time convergence to zero.

$$e_{xx} = X - X_R, z = \int_0^t e(\tau)d\tau \quad (25)$$

$$s = \ddot{e}_{xx} + \lambda e_{xx} + kz + \beta|e|^{\mu}\,\text{sgn}(e) \quad (26)$$

$$u_{eq} = \ddot{X}_R - \lambda\dot{e}_{xx} - ke_{xx} - \beta|e_{xx}|^{\mu-1}\dot{e}_{xx}\,\text{sgn}(e) \quad (27)$$

$$u_{SW} = -Ksat(\frac{S}{\phi}) \quad (28)$$

$$v = u_{eq} + u_{SW} \quad (29)$$

$$\dot{K} = \alpha_K |S| - \sigma(K - K_{\min}), K \in [K_{\max}, K_{\min}] \quad (30)$$

With the sliding surface designed as above, the system is stable for a finite time.

$$\begin{cases} e_{xx} = X_R - X \\ e_{yy} = Y_R - Y \\ e_{zz} = Z_R - Z \end{cases} \rightarrow \begin{cases} \ddot{X} = \ddot{X}_R - \ddot{e}_{xx} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\dfrac{U_1}{m} & (31) \\ \ddot{Y} = \ddot{Y}_R - \ddot{e}_{yy} = (\cos\phi\sin\theta\sin\psi - \sin\phi\sin\psi)\dfrac{U_1}{m} & (32) \\ \ddot{Z} = \ddot{Z} - \ddot{e}_{zz} = -g + \cos\phi\cos\theta\dfrac{U_1}{m} & (33) \end{cases}$$

where $\vec{v} = \ddot{\vec{e}}, \vec{v} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T, \vec{e} = \begin{bmatrix} e_{xx} & e_{yy} & e_{zz} \end{bmatrix}$

$$\begin{cases} v_1 = \ddot{X}_R + \lambda_{xx}\dot{e}_{xx} + k_x e_{xx} - \beta_x \mu \left|e_{xx}\right|^{\mu-1} \dot{e}_{xx} - K_{xx} sat(\dfrac{S_x}{\phi_x}) & (34) \\ v_2 = \ddot{Y}_R + \lambda_{yy}\dot{e}_{yy} + k_y e_{yy} - \beta_y \mu \left|e_{yy}\right|^{\mu-1} \dot{e}_{yy} - K_{yy} sat(\dfrac{S_y}{\phi_y}) & (35) \\ v_3 = \ddot{Z}_R + \lambda_{zz}\dot{e}_{zz} + k_z e_{zz} - \beta_z \mu \left|e_{zz}\right|^{\mu-1} \dot{e}_{zz} - K_{zz} sat(\dfrac{S_z}{\phi_z}) & (36) \end{cases}$$

From (42) we proceed to design such that $\lim_{t\to\infty} \vec{e} = \vec{0}$ thus:

$$\begin{cases} \ddot{X}_R = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\dfrac{U_1}{m} \\ \ddot{Y}_R = (\cos\phi\sin\theta\sin\psi - \sin\phi\sin\psi)\dfrac{U_1}{m} \\ \ddot{Z}_R = -g + \cos\phi\cos\theta\dfrac{U_1}{m} \end{cases} \rightarrow \begin{cases} \theta = \tan^{-1}(ac + bd) \\ \begin{bmatrix} \tan\phi = \dfrac{\cos(a - \tan\theta c)}{d} \\ \tan\phi = \dfrac{\cos\theta(\tan\theta d - b)}{c} \end{bmatrix} \\ U_1 = \dfrac{m(\ddot{Z}_R + g)}{\cos\phi\cos\theta} \end{cases}$$

With $a = \dfrac{\ddot{X}_R}{\ddot{Z}_R + g}, b = \dfrac{\ddot{Y}_R}{\ddot{Z}_R + g}, c = \cos\psi_R, d = \sin\psi_R$

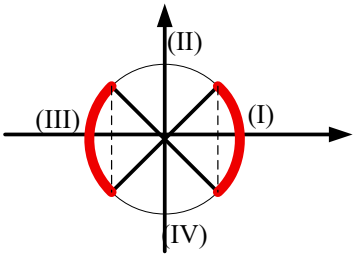

Figure 3. Behavior of Angles ψ on a Trigonometric Circle

When the value of ψ Lies in (I) và (III)

Value of ϕ is computed $\tan\phi = \dfrac{\cos(a - \tan\theta c)}{d}$

When the value of ψ Lies in (II) và (IV)

Value of ϕ is computed $\tan\phi = \dfrac{\cos\theta(\tan\theta d - b)}{c}$

V. RRT★ AND CBF

5.1. Rapidly-Exploring Random Tree Star

RRT is an asymptotically optimal sampling-based motion planning algorithm. It incrementally builds a tree by randomly sampling the configuration space and connecting each new sample to the nearest feasible node. Unlike standard RRT, the RRT* algorithm includes a rewiring step that locally updates parent nodes to minimize path cost, ensuring that the solution converges to the optimal trajectory as the number of samples increases. This makes RRT* suitable for generating globally efficient waypoints in cluttered environments such as radar-coverage maps.*

5.2. Control Barrier Function.

A Control Barrier Function (CBF) enforces forward invariance of a safety set $S = \{x \mid h(x) > 0\}$ for an affine system $\dot{x} = f(x) + g(x)u$ The CBF imposes the constraint $L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0, \alpha \in K$ class. In practice, a small quadratic program minimally adjusts the nominal control u_{nom} to satisfy the CBF and actuator limits. This guarantees that system trajectories remain inside S in real time, ensuring safety (e.g., radar-zone avoidance) without sacrificing performance.

IV. SIMULATION RESULTS

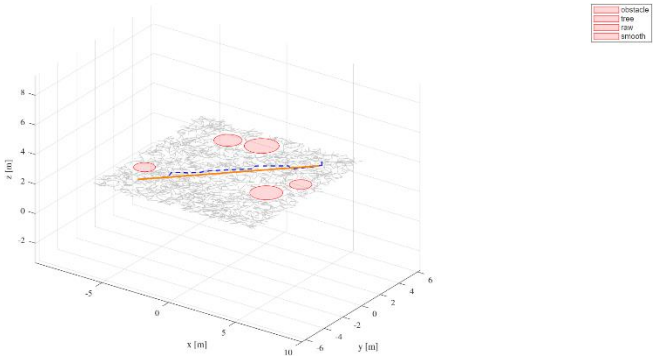

Figure 4. RRT★ algorithm simulation

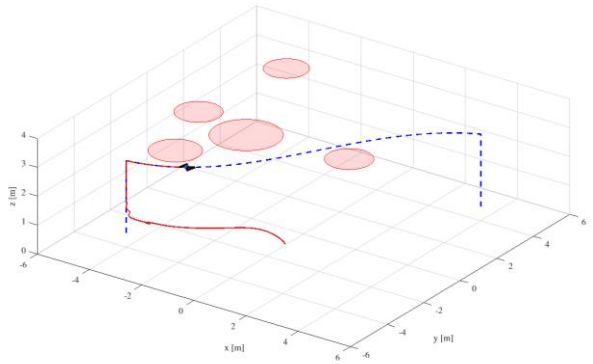

Figure 5 Simulate the tracking process

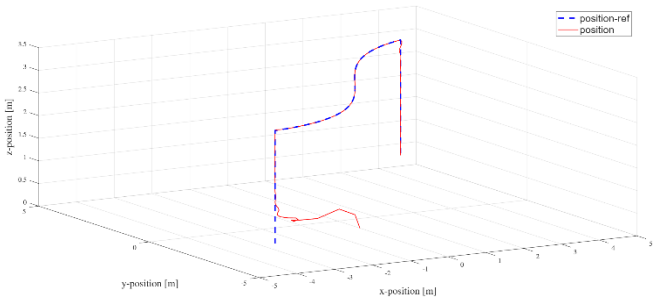

Figure 6. results after tracking

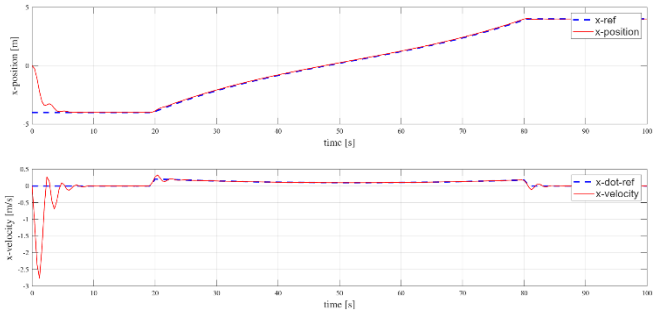

Figure 7 Tracking performance in the x-axis

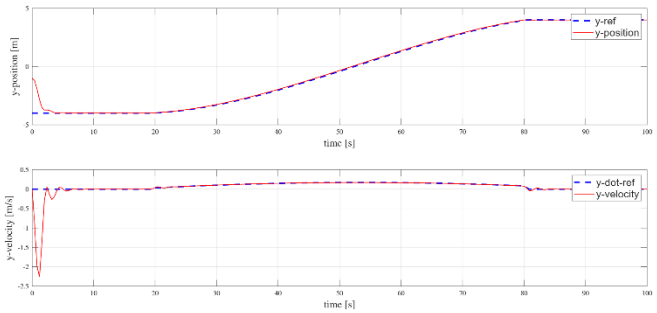

Figure 8 Tracking performance in the y-axis

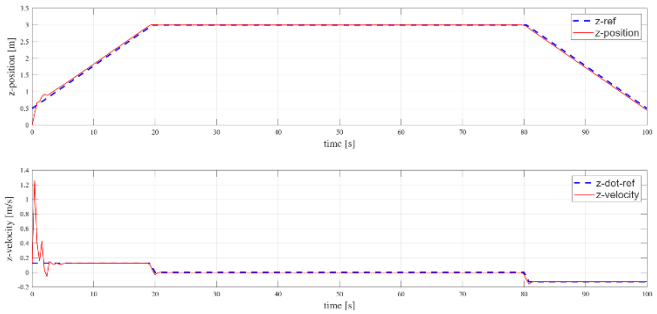

Figure 9. Tracking performance in the z-axis

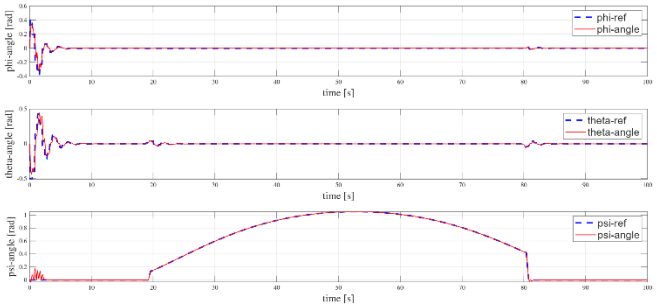

Figure 10. Attitude tracking (φ, θ, ψ)

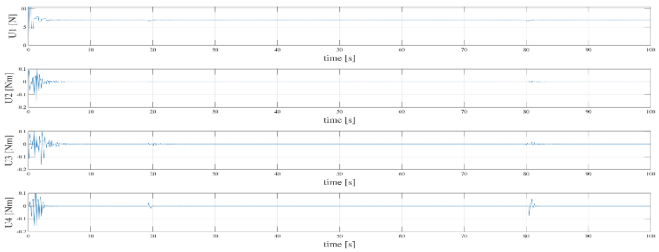

Figure 11. Control inputs (U1–U4)

V. Conclution

This paper presented an integrated radar-aware quadrotor control scheme combining RRT★ path planning, LPV–MPC guidance, and an Integral-Terminal Adaptive Sliding-Mode Controller (IT-ASMC). The proposed IT-ASMC achieves finite-time tracking without a reaching phase and maintains robustness against bounded high-frequency disturbances, as confirmed by simulation. The overall LPV–MPC-FB–CBF–IT-ASMC structure demonstrates precise, stable, and disturbance-rejected trajectory tracking, providing a strong foundation for future real-time implementation and multi-UAV cooperative extensions.

I. Acknowledment

The authors would like to express their sincere gratitude to **Dr. Phuong Tung Pham** from Ho Chi Minh City University of Technology (HCMUT) for his invaluable guidance, constructive feedback, and continuous support throughout the development of this work.

II. References

[1] A. E. H. N. T. a. D. L. K. Sidi Brahim, "Anti-saturation Finite Time Adaptive Altitude Quadrotor Control Under Unknown Disturbances," in *IFAC Papers OnLine, vol. 55, no. 22, pp. 287–292, 2022*, 2022.

[2] L. Pan, M. Catellani, L. Sabattini and N. Ayanian, "Robust Trajectory Generation and Control for Quadrotor Motion Planning with Field-of-View Control Barrier Certification," in *ICRA*, 2024.

[3] M. I. V. S. S. D. S. S. V. L. a. R. F. Zeinah Shayan, "Exponential control barrier function and model predictive control for jerk-level reactive motion planning of quadrotors," *Control Engineering Practice, vol. 164, 2025, Article no. 106489,* 2025.

[4] G. W. a. K. Sreenath, "Safety-Critical Control of a Planar Quadrotor".

[5] T. T. V. N. D. Duy Nam Bui, "Lyapunov-based Nonlinear Model Predictive Control for Attitude Trajectory Tracking of Unmanned Aerial Vehicles".

[6] V. P. M. B. Mohamed Achraf Senoussi, "Quadrotor Control using a Multilayer MPC-MHE Scheme based on LPV and Feedback Linearization Approaches".

[7] S. M. a. A. F. Omid Mofid, "Adaptive Integral-Type Terminal Sliding Mode Control for Unmanned Aerial Vehicle Under Model Uncertainties and External Disturbances," *IEEE Access, vol. 9, pp. 54497 – 54509, 2021.*

[8] C. A. IYER, "Model Predictive Control (MPC) of Quadcopters using LPV techniques".