ORIGINAL RESEARCH **Open access**

Adaptive Dual-Loop Model Predictive Control for Variable-Mass Aquaculture Feeding UAVs

Ahmad Razak[1*], Norazizah Mohamad[1], Fairuz Ismail[2], Zainal Abidin[3]

Abstract

The operation of feeding unmanned aerial vehicles (UAVs) within aquaculture environments encounters fundamental obstacles arising from temporally fluctuating mass properties, pronounced cross-coupling dynamics, and ambient perturbations, all of which undermine the performance of traditional control methodologies. The present work introduces a resilient dual-loop model predictive control (MPC) scheme refined through an adaptive niche radius genetic algorithm (ANRGA). Position regulation is handled by the external MPC loop using virtual acceleration signals, while the internal MPC loop provides attitude stabilization with real-time inertia compensation. By adaptively tuning the weighting coefficients, ANRGA circumvents the drawbacks inherent to manual parameter adjustment, inhibiting premature convergence whilst amplifying global exploration capabilities. Theoretical guarantees of system stability are established through Lyapunov-based arguments and the small-gain theorem, holding firm even when mass characteristics vary with time. MATLAB-based simulation trials spanning spiral, figure-eight, and feeding cruise trajectories reveal that the formulated ANRGA-MPC-MPC architecture confines position-tracking deviations to below 0.5 m, accelerates transient response by approximately 58% compared to standard MPC, and consistently surpasses reference controllers across metrics of precision, disturbance rejection, and convergence rate. These outcomes substantiate the feasibility of the proposed strategy for accurate, energy-conscious UAV feeding missions, thereby delivering a compelling control paradigm for intelligent aquaculture systems.

Keywords Model Predictive Control (MPC), Feeding UAV, Dual-loop control, Adaptive Niche Radius Genetic Algorithm (ANRGA), Aquaculture

*Correspondence:
Ahmad Razak
ahmad.razak@um.edu.my
[1] Department of AI for Palm Oil Yield Estimation and Carbon Sequestration, Faculty of Agriculture, University of Malaya, Kuala Lumpur, Malaysia
[2] Department of Smart Soil Health and pH Mapping, Faculty of Agriculture, Universiti Sains Malaysia, Penang, Malaysia
[3] Department of Machine Learning for Precision Weed Control, Faculty of Agriculture, Universiti Putra Malaysia, Serdang, Malaysia

Introduction

Fisheries, together with aquaculture, serve an indispensable function in nourishing the planet's expanding human population. Over the preceding several decades, worldwide aquaculture output has surged at a remarkable pace, supplying a significant share of the fish destined for human consumption [1]. Said contribution materializes across numerous realms, encompassing the reduction of impoverishment [2-4], the generation of employment opportunities [5-7], the stimulation of economic activity, and the bolstering of both income streams and nutritional adequacy. Within the spectrum of aquacultural priorities, meticulous feed administration occupies a position of primacy. Deficient feeding as well as excessive feed provision not only retard piscine development but additionally inflate operational expenditures and intensify the degradation of aquatic habitats [8]. Established aquacultural approaches, characterized by their heavy dependence on manual effort, frequently prove inefficient, financially burdensome, and subject operators to non-negligible

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which permits non-commercial use, distribution, reproduction, and adaptation in any medium or format, provided that appropriate credit is given to the original author(s) and source, a link to the license is provided, and any changes made are indicated. Adapted material must be distributed under the same license.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

hazards [9, 10]. It follows that engineering intelligent feeding apparatuses is essential for ensuring uniform dispersion of feed, maximizing nutrient assimilation, and curtailing aquatic contamination throughout feeding cycles. Unmanned Aerial Vehicle (UAV) platforms have exhibited considerable promise in satisfying these technological specifications [11]. Hence, harnessing UAV-centric solutions to deliver feeding control that is simultaneously high-precision, energy-sparing, and intelligent emerges as a pivotal scientific and engineering pursuit within present-day aquaculture.

The functional capabilities of UAVs have improved exponentially in recent years, enabling these aerial platforms to execute multiple assignments concurrently [12]. Unmanned Aerial Systems (UAS) have seen widespread adoption across marine scientific inquiry, with De Lima *et al.* [13] furnishing a methodical exposition of their conservation-oriented deployments within oceanographic exploration. Agricultural UAVs, leveraging precision surveillance, phytosanitary interventions, and mechanized sowing, have substantially increased the operational throughput of contemporary agricultural enterprises [14, 15]. Such systems are customarily fielded at high altitudes, demonstrating an aptitude for rapidly disseminating agricultural substances. That said, the integration of UAV technologies into aquaculture feeding practices remains at a nascent stage [16]. Feeding UAVs exhibit dynamic behavioral traits analogous to those observed in agricultural counterparts, given that both categories operate within low-altitude envelopes proximate to terrestrial or aquatic surfaces, where flight equilibrium remains susceptible to topographic anomalies, micro-scale atmospheric turbulence, progressive battery depletion, and fluctuations in payload mass [17]. Temporal variation in mass is a foundational determinant of the flight dynamics of feeding UAVs [18]. As feeding operations unfold, the monotonic reduction in mass demands adaptable control policies to safeguard flight equilibrium; without such provisions, uncontrolled mass attrition inexorably erodes operational performance. Accordingly, the incorporation of sophisticated attitude regulation architectures is imperative for ensuring both stabilized flight and precise feed deposition.

A diverse assortment of quadrotor attitude regulation frameworks has been scrutinized, including Incremental Nonlinear Dynamic Inversion (INDI) [19], intelligent control paradigms [20], and sliding-mode regulation [21]. Be that as it may, classical PID-type regulation remains the predominant paradigm governing small-scale UAVs deployed in precision agriculture contexts, most notably in the fragmented agrarian landscapes characteristic of southern China [22, 23]. Kapnopoulos and Alexandridis [24] put forward a controller calibration methodology anchored in Cooperative Particle Swarm Optimization (COOP-PSO), customized for quadrotor trajectory-following, employing an arrangement in which “outer-loop error-based MPC addresses position regulation” is paired with “inner-loop PID delivers attitude stabilization.” Although the dual-loop controller construct imparts meaningful conceptual stimulus, its straightforward transposition to the authentic operating milieu of aquacultural feeding remains circumscribed. Ben Abdi *et al.* [25] introduced a hybrid P-PID regulator whose amplification parameters were adaptively calibrated via Ant Colony Optimization (ACO), thereby enhancing quadrotor resilience to exogenous disturbances and dynamical indeterminacies.

Notwithstanding, attributable to the reliance upon a lone proportional element within the altitude channel, said configuration exhibits susceptibility to pronounced tracking discrepancies under variable-mass conditions. Wu *et al.* [26] formulated a Distributed Model Predictive Control (DMPC) framework in which the Adaptive Grasshopper Optimization Algorithm (AGOA) served as the foundational solver, facilitating trajectory refinement for multiple Solar UAVs engaged in cooperative target-shadowing assignments in metropolitan settings. Furthermore, documented in the literature are deep reinforcement learning methodologies grounded in the Simulated Annealing—Twin Delayed Deep Deterministic Policy Gradient (SA-TD3) architecture for quadrotor attitude equilibrium and positional trajectory tracking [27]; a comparative appraisal of eight model-free reinforcement learning algorithms applied to quadrotor attitude regulation and trajectory tracing, together with a benchmarking endorsement favoring the TD3 algorithm [28]; the Differential Evolution (DE) approach directed at optimizing PID coefficients to ameliorate quadrotor energetic economy [29]; the TD3-TPR technique incorporating Tangential Path Reward (TPR) that factors in attitude dynamics to refine quadrotor self-guided traversal [30]; as well as the tuning of Model Predictive Control (MPC) coefficients predicated upon the Improved Black-Wing Kite Algorithm (IBKA) aimed at sharpening heading regulation and depth-following fidelity of Autonomous Underwater Vehicles (AUVs) [31]. Although this particular framework resists seamless reconfiguration to accommodate the aquacultural feeding scenario under examination, its optimization philosophy nonetheless imparts instructive conceptual direction.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Propelled by accelerated advances in the semiconductor sector, embedded computational substrates have seen significant increases in processing capability, thereby enhancing their ability to solve Quadratic Programming (QP) formulations. Consequently, Model Predictive Control (MPC) methodologies now exhibit greater tractability for implementation on nonlinear, tightly coupled UAV dynamical systems, affording enhanced regulatory performance amidst such intricate dynamics.

Model Predictive Control (MPC) is a control philosophy that relies on predictive representations of system dynamics. Through the instrumentality of a Rolling Horizon Optimization (RHO) tactic, MPC forecasts the effects of control actions on the system's future state evolution [32, 33]. By way of optimization, MPC concurrently reconciles the imperatives of positional and orientational governance, thereby yielding optimal thrust–acceleration and angular-rate directives. MPC's potency hinges critically upon the fidelity of the underpinning model. In practical settings, however, an exact mathematical characterization of quadrotor UAVs proves formidable—most acutely in time-varying configurations characterized by substantial payload mass and pronounced load transience. It follows that divergences may materialize between MPC-forecast trajectories and the UAV's veridical states, ultimately impairing attitude-control exactitude. To address this predicament, a dual-loop MPC structure is proposed herein, in which position regulation is entrusted to the outer loop whilst attitude stabilization is delegated to the inner loop. The cooperative dynamic interplay between the two loops empowers the feeding UAV under study to achieve higher control fidelity. Simulated outcomes substantiate that the proposed regulator capably attenuates perturbations originating from modeling imprecision.

Notwithstanding the above, MPC-MPC architectures exhibit acute sensitivity to regulatory parameters, with heuristic calibration typically necessitating arduous iterative experimentation, thereby rendering the tuning enterprise both temporally protracted and resource-inefficient. In light of this, metaheuristic optimization offers a viable approach to identifying optimal amplification values in attitude governance. Time-honored techniques—exemplified by Genetic Algorithms (GA) [34], Particle Swarm Optimization (PSO) [35], and Simulated Annealing (SA)—tend to exhibit premature convergence, often becoming entrapped in local extrema [36, 37]. To transcend this constraint, the present inquiry refines the orthodox Genetic Algorithm and integrates it into the MPC-MPC construct, yielding an Adaptive Niche Radius Genetic Algorithm-based MPC-MPC (ANRGA-MPC-MPC). The formulated regulator was corroborated through MATLAB 2023 b simulations of quadrotor UAVs traversing three disparate motion regimes.

Dynamic Modeling and Dual-Loop MPC Design

Quadrotor UAVs epitomize a canonical class of underactuated, time-varying dynamical systems characterized by multivariable nonlinearities and pronounced cross-coupling [38], as shown in **Figure 1**. As conveyed by **Figure 2**, the feed receptacle suspended beneath the quadrotor airframe may be legitimately idealized as a temporally varying lumped mass parameter, admissible within a stipulated error bound.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

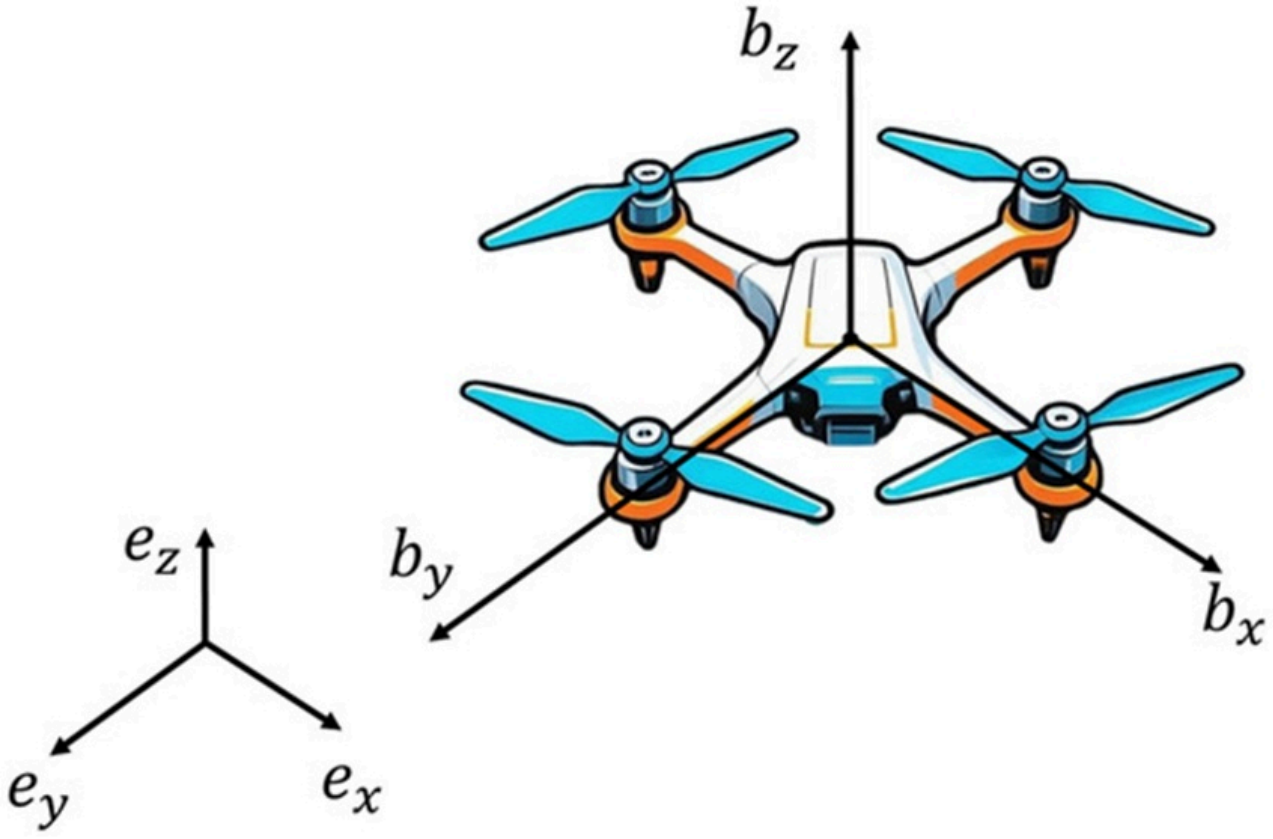


Figure 1. Configuration of the quadrotor in the right-handed body-fixed frame. bx forward, by right, bz down. Attitude (ϕ, ψ) follows the $Z{-}Y{-}X$ convention. Positions (x, z) are defined in the inertial frame used in the model and in the wind-disturbance description.

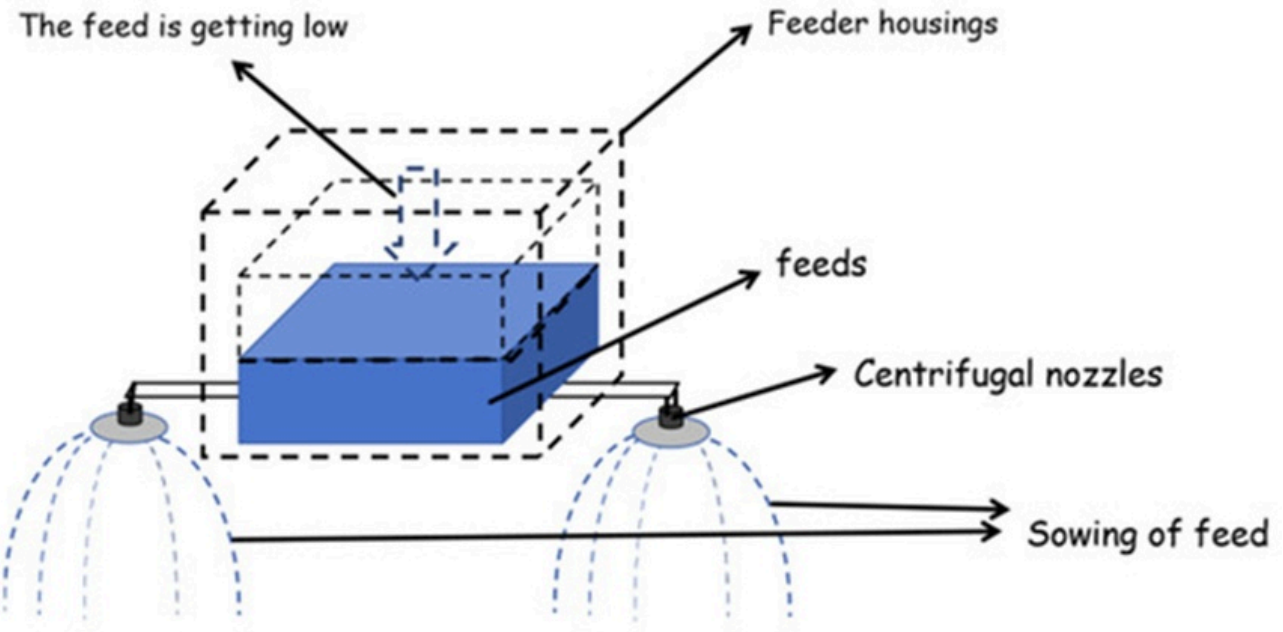


Figure 2. Operational illustration of the feed container.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

The formulation of the model and the synthesis of the control system rest upon right-handed coordinate conventions. The inertial reference triad (ex, ey, ez) is utilized to express translational displacements (x, y, z) and to characterize wind-induced perturbations. The vertical direction in the inertial frame is taken along ez, with the gravitational vector oriented opposite to this axis. The body-attached frame (bx, by, bz) follows a right-handed arrangement: bx is aligned forward, by extends rightward, and bz points downward, coinciding with the thrust axis. The airframe's orientation is described through a set of Euler angles (ϕ, θ, ψ) obeying the $Z{-}Y{-}X$ succession, with positive sense determined by the right-hand convention. The coordinate transformation that maps quantities from the body frame to the inertial frame adheres to the same rotational sequence, and the Newton–Euler dynamical relationships, along with their small-angle linear approximations, are interpreted under the same set of conventions. The above choices remain consistent with the deployment of an inertial reference system within the dynamical description and with the disturbance formulation, in which wind-field variables are expressed exclusively in inertial coordinates.

Owing to the progressive expenditure of feed and the accompanying temporal evolution of the inertia tensor, the dynamic parameters characterizing the UAV experience continual modification within the governing equations. To capture this behavior, the instantaneous mass is formally prescribed as:

$$m(t) = m_0 + \zeta \cdot t \quad (1)$$

Within this definition, m0 stands for the UAV's fully burdened mass, (<0) quantifies the rate at which feed is discharged per unit of time (kg/s), and t indicates the accumulated flight time. Correspondingly, the temporal derivative of the moment of inertia is expressed through:

$$\frac{dI}{dt} = \frac{d}{dt}\left(I_{cm} + m(t)d^2\right) \quad (2)$$

Here, Icm denotes the container's moment of inertia computed about its native center of mass, whereas d measures the spatial separation between the container's mass center and the aggregate centroid of the UAV. The symbolic conventions introduced at this point are maintained uniformly throughout the entire manuscript and shall not be redefined in subsequent passages.

The mathematical description of the UAV originates from the Newton–Euler framework, culminating in the dynamical representation that follows [39]:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\begin{aligned}
\ddot{x} &= \frac{(cos\phi sin\theta cos\psi + sin\phi sin\psi)u_1}{m(t)} + \xi_x \\
\ddot{y} &= \frac{(cos\phi sin\theta sin\psi - sin\phi cos\psi)u_1}{m(t)} + \xi_y \\
\ddot{z} &= \frac{(cos\phi cos\theta)u_1}{m(t)} - g + \xi_z \\
\ddot{\phi} &= \frac{(I_y(t) - I_z(t))\dot{\theta}\dot{\psi}}{I_x(t)} + \frac{u_2}{I_x(t)} + \xi_\phi \\
\ddot{\theta} &= \frac{(I_z(t) - I_x(t))\dot{\phi}\dot{\psi}}{I_y(t)} + \frac{u_3}{I_y(t)} + \xi_\theta \\
\ddot{\psi} &= \frac{(I_x(t) - I_y(t))\dot{\phi}\dot{\theta}}{I_z} + \frac{u_4}{I_z} + \xi_\psi
\end{aligned} \quad (3)$$

In these relations, x, y, and z designate the translational degrees of freedom of the UAV's mass center expressed in the inertial frame; ϕ, θ, and ψ signify the roll, pitch, and yaw orientations, in that order; $u1$, $u2$, $u3$, and $u4$ represent the control effector signals; g captures the gravitational constant, fixed at 9.81 m/s²; ξx, ξy, and ξz account for external perturbations impinging upon the translational motion of the UAV's mass center; and $\xi\phi$, $\xi\theta$, and $\xi\psi$ reflect external perturbations acting on the rotational dynamics referenced to the UAV's mass center.

As previously highlighted, the native nonlinear characteristics, strong cross-channel couplings, and underactuated configuration inherent to quadrotor platforms necessitate linearizing their dynamic equations. During low-velocity, fixed-altitude operation over aquaculture basins, the criterion $|\phi|$, $|\theta| \leq 10°$ holds t, under which small-angle linearization substantially curtails computational load while safeguarding the real-time tractability of receding-horizon optimization. For moderate angular displacements ($|\phi|$, $|\theta| = 20° \sim 40°$) occasioned by aggressive maneuvering, wind gusts, or swift trajectory reconfigurations, Section 2.1 elaborates upon nonlinear augmentation strategies together with constraint enforcement via a Reference Governor (RG), thereby widening the domain of controllable flight conditions.

Controller design

Every matrix and variable appearing in this section is governed by the coordinate frames and notational rules established in Section 2, which secures a uniform interpretation of (x, y, z, ϕ, θ, ψ) and of the linearization performed in the vicinity of hover. Under the premise of small angular deviations (ϕ, $\theta \approx 0$), with the heading angle ψ regarded as approximately invariant ($\psi \approx \psi_0$), and upon linearizing around the near-hover trim state ($u1 \approx mg$), the following linearized system description emerges:

$$\ddot{x} \approx g\left(\theta\cos\psi_0 + \phi\sin\psi_0\right),\ \ddot{y} \approx g\left(\theta\sin\psi_0 - \phi\cos\psi_0\right),\quad \ddot{z} \approx \frac{u_1}{m} - g \quad (4)$$

To empower the outer regulation layer with the capability to directly command the "acceleration reference," a synthetic control variable is introduced via:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$u_p := a_c = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (5)$$

In the above, up embodies the commanded acceleration triplet of the UAV's mass center. By intervening directly at the acceleration level, the outer loop invokes a double-integrator surrogate to project acceleration directives onto positional trajectories, thereby diminishing the intermediary nonlinear interplay between "position" and "attitude" and rendering the outer-loop predictive computations more transparent.

Captured through an idealized double-integrator abstraction:

$$\dot{x} = v_x, \dot{v}_x = a_x; \dot{y} = v_y, \dot{v}_y = a_y; \dot{z} = v_z, \dot{v}_z = a_z \quad (6)$$

From the foregoing, the state vector and input vector about the outer loop may be assembled as:

$$x_p = \begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix}^{\mathsf{T}}, u_p = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^{\mathsf{T}} \quad (7)$$

The resultant continuous-time dynamical representation can thus be cast in the form:

$$\dot{x}_p = A_{p,c} x_p + B_{p,c} u_p, A_{p,c} = \begin{bmatrix} 0_{3\times3} & I_3 \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, B_{p,c} = \begin{bmatrix} 0_{3\times3} \\ I_3 \end{bmatrix} \quad (8)$$

Discretization of the system is accomplished by means of a Zero-Order Hold (ZOH) procedure:

$$x_{p,k+1} = A_p x_{p,k} + B_p u_{p,k}, y_{p,k} = C_p x_{p,k} \quad (9)$$

A Zero-Order Hold (ZOH) discretization strategy is employed since both the airborne flight controller and the ground-based supervisory processor are clocked at an identical sampling period Ts. The ZOH construct effectively replicates the actuator behavior of maintaining a constant command value across successive sampling epochs, while remaining consistent with the discrete prediction framework intrinsic to receding-horizon optimization.

Within the above expressions,

$$A_p = \begin{bmatrix} I_3 & T_p I_3 \\ 0 & I_3 \end{bmatrix}, B_p = \begin{bmatrix} \frac{1}{2} T_p^2 I_3 \\ T_p I_3 \end{bmatrix}, C_p = \begin{bmatrix} I_3 & 0 \end{bmatrix} \quad (10)$$

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Let stands for the prediction horizon length and for the control horizon length, with the restriction ≤ enforced. These quantities are specified by:

$$\mathrm{X}_p = \begin{bmatrix} x_{p,k+1} \\ \vdots \\ x_{p,k+N_p^p} \end{bmatrix}, \mathrm{U}_p = \begin{bmatrix} u_{p,k} \\ \vdots \\ u_{p,k+N_c^p-1} \end{bmatrix}, \mathrm{Y}_p = \begin{bmatrix} y_{p,k+1} \\ \vdots \\ y_{p,k+N_p^p} \end{bmatrix} \quad (11)$$

From which it ensues that:

$$\begin{aligned} \mathrm{X}_p &= \Phi_p x_{p,k} + \Gamma_p \mathrm{U}_p \\ \mathrm{Y}_p &= O_p x_{p,k} + \mathcal{J}_p \mathrm{U}_p \end{aligned} \quad (12)$$

Where:

$$\Phi_p = \begin{bmatrix} A_p \\ A_p^2 \\ \vdots \\ A_p^{N_p^p} \end{bmatrix}, \Gamma_p = \begin{bmatrix} B_p & 0 & \dots & 0 \\ A_p B_p & B_p & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_p^{N_p^p-1} B_p & A_p^{N_p^p-2} B_p & \dots & B_p \end{bmatrix} \quad (13)$$

$$C_p = I_{N_p^p} \otimes C_p, O_p = C_p \Phi_p, \mathcal{J}_p = C_p \Gamma_p$$

In an analogous manner, performing linearization around the small-angle trim condition (ϕ, θ, ψ, p, q, r) ≈ 0, and omitting the quadratic coupling contributions pq, pr, and qr, leads to:

$$\dot{\phi} = p, \dot{\theta} = q, \dot{\psi} = r, \dot{p} \approx \frac{u_2}{I_x(t)}, \dot{q} \approx \frac{u_3}{I_y(t)}, \dot{r} \approx \frac{u_4}{I_z(t)} \quad (14)$$

The state vector and input vector associated with the inner loop are consequently identified as:

$$x_a = [\, \phi \quad \theta \quad \psi \quad p \quad q \quad r \,]^\top, u_a = [\, u_2 \quad u_3 \quad u_4 \,]^\top \quad (15)$$

Where:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$A_{a,c} = \begin{bmatrix} 0_{3\times3} & I_3 \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, B_{a,c}(t) = \begin{bmatrix} 0_{3\times3} \\ \operatorname{diag}\left(I_x^{-1}(t), I_y^{-1}(t), I_z^{-1}(t)\right) \end{bmatrix} \quad (16)$$

Once discretized, the inner-loop dynamics assume the representation:

$$x_{a,k+1} = A_a x_{a,k} + B_a(t_k) u_{a,k}, y_{a,k} = C_a x_{a,k} \quad (17)$$

Where:

$$A_a = \begin{bmatrix} I_3 & T_a I_3 \\ 0 & I_3 \end{bmatrix}, B_a(t_k) = \begin{bmatrix} \frac{1}{2} T_a^2 \operatorname{diag}\left(I_x^{-1}, I_y^{-1}, I_z^{-1}\right) \\ T_a \operatorname{diag}\left(I_x^{-1}, I_y^{-1}, I_z^{-1}\right) \end{bmatrix}_{t=t_k}, \quad (18)$$
$$C_a = [I_3 \quad 0]$$

Following identical reasoning, the following expression is derived:

$$\begin{aligned} \mathrm{X}_a &= \Phi_a x_{a,k} + \Gamma_a \mathrm{U}_a \\ \mathrm{Y}_a &= O_a x_{a,k} + \mathcal{J}_a \mathrm{U}_a \end{aligned} \quad (19)$$

Where:

$$\Phi_a = \begin{bmatrix} A_a \\ A_a^2 \\ \vdots \\ A_a^{N_p^a} \end{bmatrix}, \Gamma_a = \begin{bmatrix} B_a & 0 & \dots & 0 \\ A_a B_a & B_a & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_a^{N_p^a-1} B_a & A_a^{N_p^a-2} B_a & \dots & B_a \end{bmatrix} \quad (20)$$
$$C_a = I_{N_p^a} \otimes C_a, O_a = C_a \Phi_a, \mathcal{J}_a = C_a \Gamma_a.$$

Based on the linearized discrete-time models elaborated above, the performance indices governing both the outer- and inner-loop regulators can be formulated. The outer-loop cost functional is given by:

$$J_p = \left(\mathrm{Y}_p - \mathrm{R}_p\right)^{\mathrm{T}} \bar{Q}_p \left(\mathrm{Y}_p - \mathrm{R}_p\right) + \mathrm{U}_p^{\mathrm{T}} \bar{R}_{0p} \mathrm{U}_p \quad (21)$$

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Within this expression, $\bar{Q}p = blkdiag(Qp, \ldots, Qp, f) \in \mathbb{S}_{+}^{3N_p}$, and $\bar{R}0p = blkdiag(R0p, \ldots) \in \mathbb{S}_{+}^{3N_c^p}$ respectively denote the state-penalty weighting matrix and the control-effort weighting matrix.

Reformulated as a Quadratic Programming (QP) instance:

$$J_p = \frac{1}{2}\mathrm{U}_p^T H_p \mathrm{U}_p + f_p^T \mathrm{U}_p + const$$
$$H_p = 2\left(\mathcal{J}_p^{\,T} \bar{Q}_p \mathcal{J}_p + \bar{R}_{0p}\right) \quad (22)$$
$$f_p = 2\mathcal{J}_p^{\,T} \bar{Q}_p \left(O_p x_{p,k} - \mathrm{R}_p\right)$$

The inner-loop cost functional is structured as:

$$J_a = \left(\mathrm{Y}_a - \mathrm{R}_a\right)^{\mathrm{T}} \bar{Q}_a \left(\mathrm{Y}_a - \mathrm{R}_a\right) + \Delta \mathrm{U}_a^{\mathrm{T}} \bar{R}_{\Delta a} \Delta \mathrm{U}_a + \mathrm{U}_a^{\mathrm{T}} \bar{R}_{0a} \mathrm{U}_a \quad (23)$$

Reformulated as a Quadratic Programming (QP) instance:

$$J_a = \left(\mathrm{Y} - \mathrm{R}\right)^{\mathrm{T}} \bar{Q}_a \left(\mathrm{Y} - \mathrm{R}\right) + \mathrm{U}_a^T \bar{R}_{0a} \mathrm{U}_a$$
$$H_a = 2\left(\mathcal{J}_a^{\,T} \bar{Q}_a \mathcal{J}_a + \bar{R}_{0a}\right) \quad (24)$$
$$f_a = 2\mathcal{J}_a^{\,T} \bar{Q}_a \left(O x_{a,k} - \mathrm{R}_a\right)$$

At this stage, the mathematical formulations underpinning both the outer- and inner-loop controllers are fully specified, with the overarching dual-loop MPC schematic provided in **Figure 3**.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

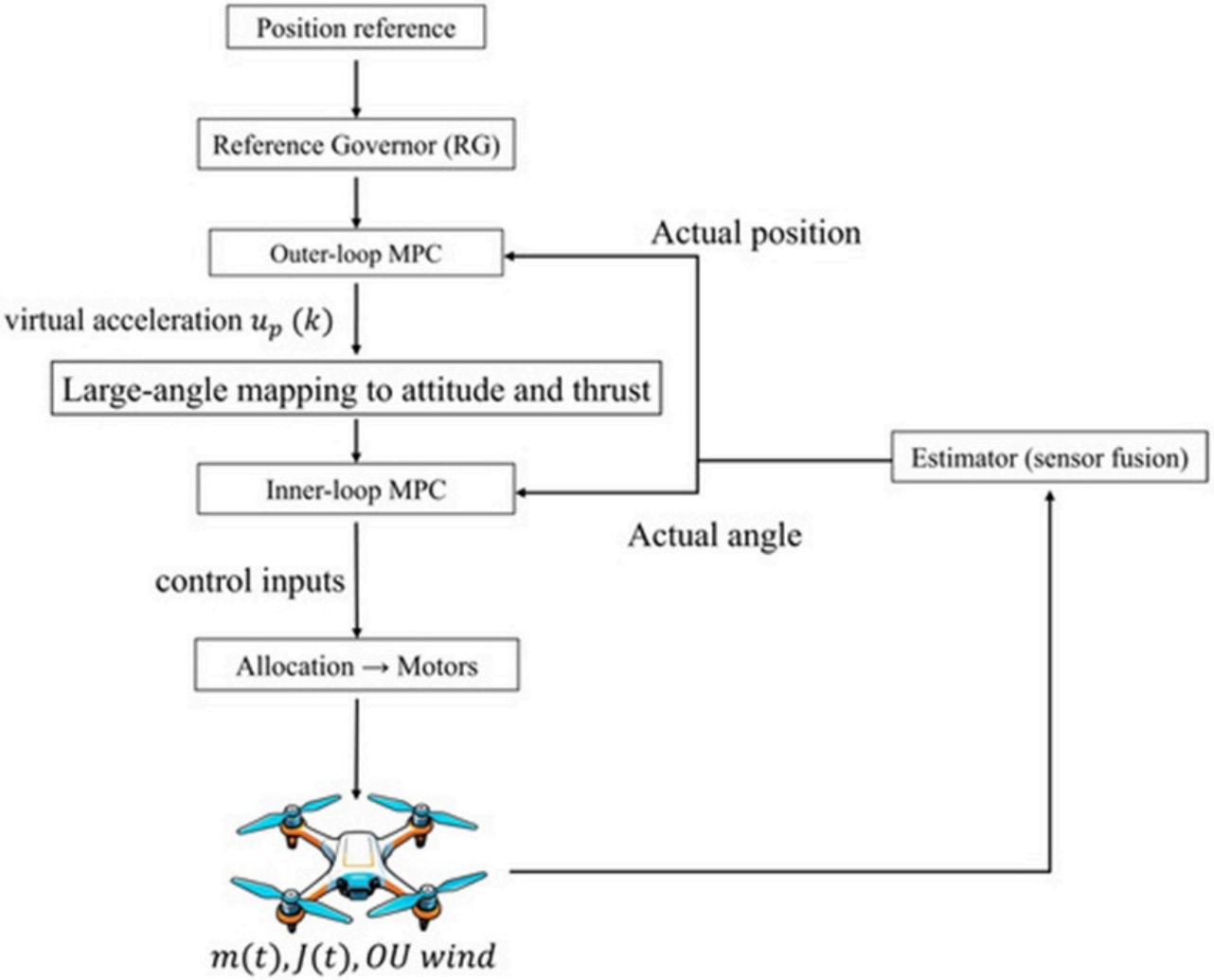


Figure 3. Schematic of the dual-loop MPC architecture.

As depicted in **Figure 3**, the outer loop ingests $(r - x)$ as its incoming signal and synthesizes ac, into which the gravity-offset component is assimilated; the inner loop converts ac into the commanded attitude (and the associated aggregate thrust), which is subsequently apportioned among the four rotors via a thrust–torque mapping. The separation of bandwidths enforces a dynamical hierarchy in which "the outer loop operates at lower frequencies while the inner loop responds at higher frequencies," with a customary bandwidth ratio of at least 5.

Accommodating large-angle flight regimes

The controller formulated in the prior section relies on small-angle and near-hover idealizations, which are sufficient for steady, fixed-altitude aquacultural sorties under nominal conditions. In operational reality, however, UAV platforms are routinely compelled to execute evasive turns, counteract wind gusts, and transition between waypoints with agility—maneuvers during which roll and pitch excursions may reach 20 °- 40 °. Clinging to the small-angle approximation within such flight envelopes leads to substantial misrepresentations of gravitational projection and inter-axis coupling, thereby diminishing regulation fidelity and endangering closed-loop stability. To remedy this shortcoming, the present section broadens the scope of both the outer and inner regulatory loops to encompass large-angle operational domains while retaining the dual-loop MPC architecture.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Incorporating the temporally evolving mass (t) and inertia tensor (t), the translational and rotational equations of motion for the UAV assume the form:

$$\begin{aligned} m\dot{\mathrm{v}} &= mg\mathrm{e}_3 - fR\mathrm{e}_3 - \mathrm{D}(\mathrm{v}) + \mathrm{d}_f \\ J\dot{\omega} &= \tau - \omega \times (J\omega) + \mathrm{d}_\tau \\ \dot{R} &= R\hat{\omega}, \hat{\omega}x = \omega \times x \end{aligned} \tag{25}$$

Within these expressions, $R \in SO(3)$ stands for the attitude rotation operator, ω captures the angular rate resolved in body axes, f conveys the net thrust produced, and τ denotes the torque command vector.

As introduced previously, the outer-loop synthetic control is taken as $\overset{ac}{=} [ax, ay, az]^\top$, which, when the airframe operates at sizable tilt angles, must transform the requisite attitude setpoint and thrust demand. The quantity is characterized by:

$$\mathrm{a}_{\mathrm{tot}} := g\mathrm{e}_3 - \mathrm{a}_c, \mathrm{b}_{3d} := \frac{\mathrm{a}_{\mathrm{tot}}}{\|\mathrm{a}_{\mathrm{tot}}\|} \tag{26}$$

Here, b3d signifies the intended alignment of the body-frame +z direction. Jointly with the prescribed yaw orientation ψd, the following relationships emerge:

$$\begin{aligned} \mathrm{b}_{1c} &= [\cos\psi_d, \sin\psi_d, 0]^\top \\ \mathrm{b}_{2d} &= \frac{\mathrm{b}_{3d} \times \mathrm{b}_{1c}}{\|\mathrm{b}_{3d} \times \mathrm{b}_{1c}\|} \\ \mathrm{b}_{1d} &= \mathrm{b}_{2d} \times \mathrm{b}_{3d} \end{aligned} \tag{27}$$

The requisite thrust amplitude can be written as:

$$f_d = m\|\mathrm{a}_{\mathrm{tot}}\| \tag{28}$$

Adopting the inner-loop state description ξ = (R, ω) together with the actuation vector u = (τ, f), Equation (27) can be recast into the discretized representation below:

$$\xi_{k+1} = \Phi(\xi_k, \mathrm{u}_k) \tag{29}$$

In this formulation, Φ(·) is evaluated via numerical quadrature. The performance index for the inner-loop NMPC is constructed as:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$J_{\text{in}} = \sum_{k=0}^{N_p-1} \left(\| e_{R,k} \|_{Q_R}^2 + \| e_{\omega,k} \|_{Q_\omega}^2 + \| \Delta\tau_k \|_{R_\tau}^2 + \| \Delta f_k \|_{R_f}^2 \right) \quad (30)$$

The restrictions are codified through:

$$0 \le f_k \le f_{max}, \| \tau_k \| \le \tau_{max}, |\theta| \le \theta_{max}, \| \omega \| \le \omega_{max} \quad (31)$$

Given the constrained processing capability of the UAV's embedded processor, a Linear Parameter-Varying (LPV) surrogate model is substituted. Let the scheduling quantities be identified as:

$$\sigma = \left[\sin\phi, \sin\theta, m, J_x, J_y \right] \quad (32)$$

The matrix pairs (Ai, Bi) are evaluated offline across a grid of operating points, and barycentric interpolation performed in real time furnishes:

$$A(\sigma) = \sum_t w_t(\sigma) A_t, B(\sigma) = \sum_t w_t(\sigma) B_t \quad (33)$$

Consequently, the incremental linear MPC is articulated as:

$$\Delta x_{k+1} = A(\sigma_k)\Delta x_k + B(\sigma_k)\Delta u_k \quad (34)$$

The optimization target is expressed as:

$$min \sum \left(\| \Delta x \|_Q^2 + \| \Delta u \|_R^2 \right) \quad (35)$$

Subject to limitations imposed on orientation angles, angular rates, and propulsive force.

To safeguard the realizability of outer-loop directives, the synthetic command a_c is regulated accordingly.

Lateral acceleration restriction:

$$\| a_c^{xy} \| \le g \tan\theta_{max} \quad (36)$$

With $a_c^{xy} = [ax, ay, 0]^\top$.

Path curvature restriction:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\kappa \leq \frac{g\tan\theta_{max}}{v^2} \quad (37)$$

In the above relation, κ represents the trajectory curvature, while v denotes the horizontal speed component.

Whenever these thresholds are breached, the Reference Governor (RG) either smooths the path geometry or attenuates the reference acceleration, thereby preserving the feasibility of the inner-loop attitude response. The RG is embedded to underwrite the geometric executability of the outer-loop reference trajectory, certifying that the synthetic input generated by the outer loop never exceeds the physical envelope of the inner-loop attitude dynamics.

Throughout MATLAB simulation trials, the mean computational duration per optimization step was 0.02 s (Intel i5 CPU, sampling period of 10 ms), substantiating that the proposed control scheme is real-time viable on contemporary embedded platforms.

This chapter examined the time-varying dynamical characteristics of the feeding UAV by proposing a linearized dual-loop MPC formulation grounded in small-angle assumptions, within which state and control penalization terms were incorporated into the objective functional to balance tracking precision with energetic prudence. Building on this groundwork, an augmentation tailored to large-angle flight conditions was proposed, incorporating nonlinear dynamics, an LPV surrogate representation, and a reference governor, thereby ensuring feasibility and robust stability across demanding maneuvering profiles. The foregoing synthesis delivers a unified regulatory framework that paves the way for integrating optimization algorithms in the ensuing chapter.

Adaptive Niche Radius Genetic Algorithm for MPC Optimization

The work at hand targets a safety-critical dual-loop MPC tailored for aquaculture feeding UAVs that grapple with variable mass and ground-proximal wind disturbances. The weighting matrices (Q, R) are tuned offline, ensuring that the fielded controller remains a conventional constraint-obeying MPC with the stability guarantees derived in this paper, while imposing a foreseeable onboard load of a single QP solves per control cycle. The genetic algorithm runs exclusively before flight deployment and imposes no additional real-time processing demand. The weight-selection problem assumes a multimodal landscape due to the interplay between outer and inner loops and time-varying inertial properties. To curb the risk of early stagnation, an Adaptive Niche Radius Genetic Algorithm is applied. Fitness sharing upholds population heterogeneity across the initial generational stages, whereas a gradually contracting niche radius systematically bolsters exploitative refinement during later epochs. The observed convergence histories and the time-domain figures of merit tabulated in the Results section remain consistent with this evolutionary pattern.

Reinforcement learning represents a compelling prospective direction. However, attaining a comparable level of explicit constraint governance, certifiable stability, and deployment transparency would require extensive experience with interaction corpora, dedicated safety-filtering mechanisms, and additional computational and storage overhead for policy inference and verification. Considering the immediate deployment priorities—namely, stability guarantees, interpretability, and ease of sustainment—an offline-optimized MPC driven by ANRGA stands as the most fitting choice.

Genetic-algorithm-driven adaptive tuning of weighting coefficients

As highlighted earlier, the practical impossibility of obtaining a precise quadrotor UAV model often results in reduced control fidelity. Accordingly, the weighting coefficients embedded within the Model Predictive Controller (MPC) demand adaptive adjustment to offset this shortcoming. The Genetic Algorithm (GA) is a biologically motivated evolutionary optimization methodology founded on the mechanics of genetic inheritance. Using genetic operators—selection, crossover, and mutation—it efficiently traverses intricate search spaces to pinpoint near-optimal solutions. Harnessing

© 2023 The Author(s). | Published by Agro Intelligence Research Press

this capability, the GA can be married to the MPC structure **(Figure 4)** to avoid manual calibration of weighting coefficients, thereby enabling rapid determination and construction of high-performance weighting matrices.

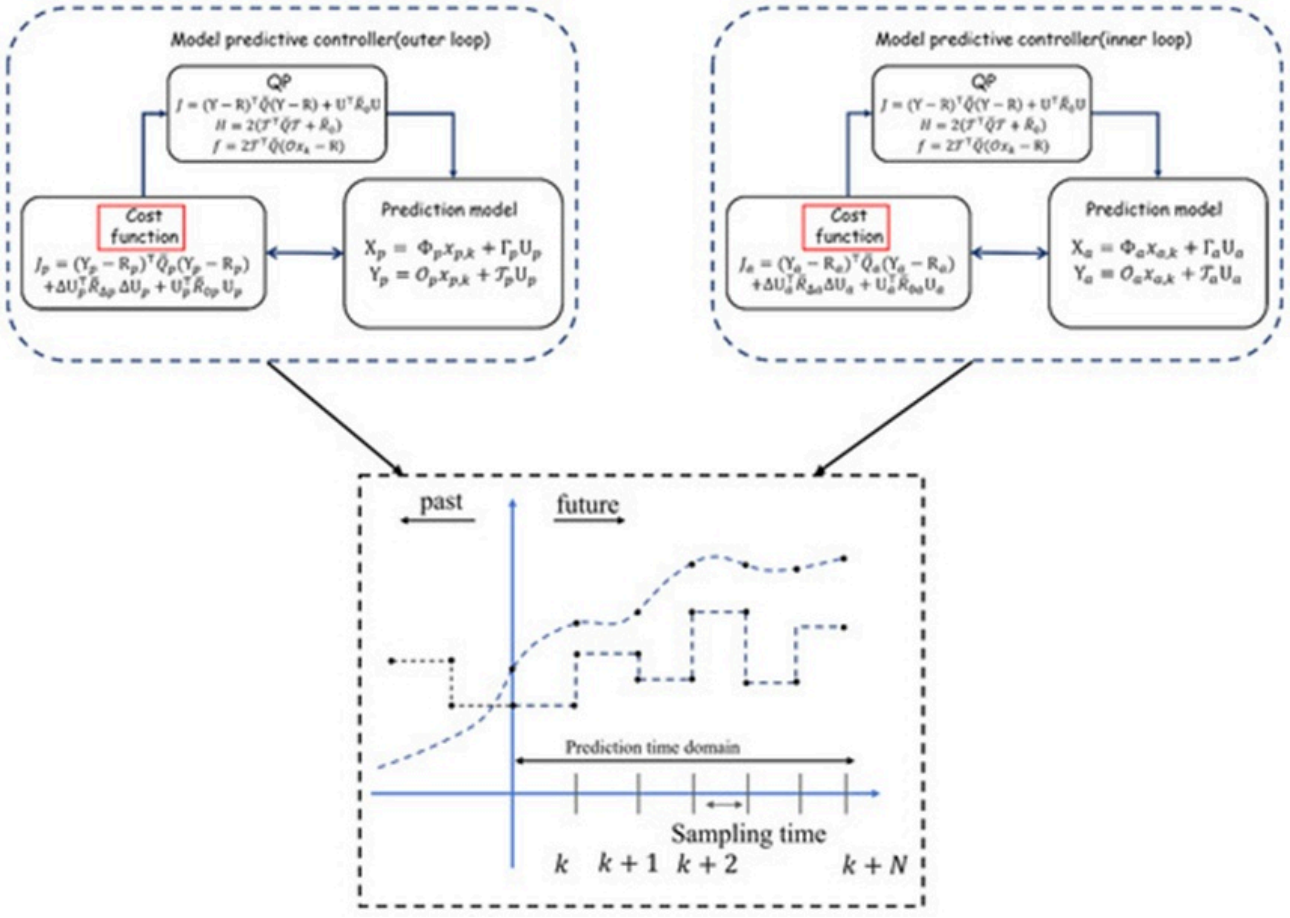


Figure 4. MPC structure integrated with the Genetic Algorithm (GA).

To begin, an initial population of weighting coefficient combinations is spawned, with chromosomes encoded in real-valued notation to signify the diagonal elements of the weighting matrix (the outer loop being taken as a representative illustration):

$$\alpha = \left[\underbrace{q_1, q_2, \cdots, q_n}_{n\ state-weighting\ factors} , \underbrace{r_1, r_2, \cdots, r_m}_{m\ control-weighting\ factors} \right] \in \mathbb{R}^{n+m} \quad (38)$$

Here, $qi=log10(Qii)$, the logarithmic transformation is adopted to enforce positive definiteness; $Ri=log10(Rii)$, employed to avert numerical overflow.

Once the initial population is established, a fitness evaluation is executed, with the associated evaluation function formulated as:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\mathcal{F}(\alpha) = \frac{1}{1 + J_{track} + \lambda J_{control}} \quad (39)$$

Where $Jtrack$ captures the tracking-error contribution, expressed as:

$$J_{track} = \frac{1}{T}\sum_{k=0}^{T} \| \tilde{x}_k \|^2_{W_x}, W_x = diag(w_{x1}, \ldots, w_{xn}) \quad (40)$$

$Jcontrol$ captures the control-energy consumption contribution, expressed as:

$$J_{control} = \frac{1}{T}\sum_{k=0}^{T} \| \Delta u_k \|^2_{W_u}, \Delta u_k = u_k - u_{k-1} \quad (41)$$

λ is the trade-off coefficient that balances the penalty imposed on control-energy consumption against that on tracking-error performance. A comparatively large λ steers the system toward minimizing energy outlay at the sacrifice of tracking precision, whereas a comparatively small λ raises tracking fidelity but incurs elevated energy consumption. In real flight environments, λ additionally reinforces the system's ability to suppress transient disturbances. Throughout the iterative progression of the Genetic Algorithm (GA), λ undergoes dynamic adaptation along the Pareto front:

$$\lambda^{(k+1)} = \lambda^{(k)} \cdot exp\left(-\beta \frac{\Delta J_{track}}{\Delta J_{control}}\right) \quad (42)$$

Here, $\begin{matrix}\Delta Jtrack \\ = J_{track}^{(k)} \\ - J_{track}^{(k-1)}\end{matrix}$ indicates the incremental shift in the tracking-error contribution; $\begin{matrix}\Delta Jcontrol \\ = J_{control}^{(k)} \\ - J_{control}^{(k-1)}\end{matrix}$ indicates the incremental shift in the control-energy contribution.

To purge dimensional mismatches that might distort the weighting, the error term is normalized to its peak value. In contrast, the energy-consumption term is normalized to the full-load power reference, and both are scaled to [0, 1]. The learning rate β=0.1 is pchosen to ensure smooth convergence while restraining pronounced parameter fluctuations triggered by rapid mass depletion during feeding, thereby achieving a compromise between flight stability and energy thrift. This choice is supported by exhaustive sensitivity studies that show that over the range [0.05, 0.2], the algorithm exhibits virtually no variation in convergence characteristics.

In the preceding formulation, $\tilde{x}\ k$ designates the tracking error at time instant k, while Wx represents the weighting coefficient attached to the tracking error at that same instant.

Following assessment using **Eq. 41**, the initial population enters the genetic iterations, commencing with population selection, which employs a tournament-based mechanism. The elite individual has a probability of 1 of winning a single tournament, conditional on its selection among the candidates. The probability of being excluded equals $1-1/N$; consequently:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$P_{\text{not selected}} = \left(1 - \frac{1}{N}\right)^s \quad (43)$$

$$P_{best} = 1 - \left(1 - \frac{1}{N}\right)^s \quad (44)$$

In actual tournament selection, the likelihood of an elite individual's selection varies with the fitness distribution. Let $p1$ denote the probability that the elite individual triumphs in a random contest:

$$p_1 = \Phi\left(\frac{\mathcal{F}_{max} - \mu_{\mathcal{F}}}{\sigma_{\mathcal{F}}}\right) \quad (45)$$

For any pair of individuals and j, the probability that i outperforms j is given by:

$$P\left(\mathcal{F}_i > \mathcal{F}_j\right) = \Phi\left(\frac{\mathcal{F}_i - \mathcal{F}_j}{\sqrt{2}\sigma_{\mathcal{F}}}\right) \quad (46)$$

Here, $\sigma_{\mathcal{F}} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\mathcal{F}\left(\theta i\right) - \mu\mathcal{F}\right)2}$ corresponds to the standard deviation of the population fitness distribution; $\mu_{\mathcal{F}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{F}\left(\theta i\right)$ corresponds to the mean fitness of the population; (x) stands for the cumulative distribution function (CDF) of the standard normal distribution. For the elite individual, αi is given as:

$$P\left(\mathcal{F}_{best} > \mathcal{F}_j\right) \approx \Phi\left(\frac{\Delta}{\sigma_{\mathcal{F}}}\right), \Delta = \mathcal{F}_{best} - \mu_{\mathcal{F}} \quad (47)$$

From this, it follows that:

$$p_1 = \Phi\left(\frac{\Delta}{\sigma_F}\right) \quad (48)$$

Additionally, the selection pressure is defined as:

$$SP = \frac{P_{best}}{P_{avg}} = \frac{1 - \left[1 - \Phi\left(\frac{\Delta}{\sigma_F}\right)\right]^s}{\frac{1}{N}} \quad (49)$$

© 2023 The Author(s). | Published by Agro Intelligence Research Press

In UAV attitude-control optimization, a tempered increase in selection pressure accelerates the discovery of stable parameter configurations. Yet excessive pressure may trap solutions in a local basin, diminishing robustness. For this reason, the present work adopts fitness-variance-based dynamic modulation of selection pressure, thereby reconciling swift convergence with robustness in controller tuning. The fitness-variance coefficient is expressed as $\begin{matrix} c \\ = \sqrt{\sigma_{\mathscr{F}}^2}/\mu\mathscr{F}. \\ = \sigma\mathscr{F}/\mu\mathscr{F} \end{matrix}$

After selection, the fittest individuals advance to chromosome crossover and mutation, yielding offspring with higher fitness. Owing to the continuous nature of the weighting coefficients, crossover is carried out via the Simulated Binary Crossover (SBX) scheme, while mutation is effected through polynomial mutation. The choice of SBX and polynomial mutation is motivated by the fact that MPC weighting parameters are continuous, real-valued variables. This approach safeguards the physical feasibility of the generated parameters. It curtails the appearance of unstable extreme control gains—a consideration of particular significance for the tightly coupled outer and inner loops within the dual-loop MPC architecture.

Let the two parent chromosomes be expressed as:

$$\begin{cases} p_1 = \left[p_{1,1}, p_{1,2}, \cdots, p_{1,D}\right] \\ p_2 = \left[p_{2,1}, p_{2,2}, \cdots, p_{2,D}\right] \end{cases} \quad (50)$$

Here, D=n+m specifies the chromosome dimensionality.

Determine the gene disparity between the parent chromosomes:

$$\Delta_t = \left|p_{1,t} - p_{2,t}\right| \quad (51)$$

The crossover probability pc dictates whether a given parental pair experiences crossover. For each parental pair, draw a random number r∼(0,1):

When crossover is executed, the expansion factor βi is first constructed:

$$\beta_t = \begin{cases} (\alpha \cdot u_t)^{\frac{1}{\eta_c + 1}} & u_t \le 0.5 \\ \left(\frac{1}{2 - \alpha \cdot u_t}\right)^{\frac{1}{\eta_c + 1}} & u_t > 0.5 \end{cases} \quad (52)$$

Here, ui∼(0,1), and ηc indicate the distribution index of the crossover operator.

The offspring chromosomes can be formulated through the expansion factor as:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\begin{bmatrix} c_{1,t} \\ c_{2,t} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1+\beta_t & 1-\beta_t \\ 1-\beta_t & 1+\beta_t \end{bmatrix}\begin{bmatrix} p_{1,t} \\ p_{2,t} \end{bmatrix} \quad (53)$$

Following expansion, the following holds:

$$\begin{cases} c_{1,t} = \frac{1}{2}\left[(1+\beta_t)p_{1,t} + (1-\beta_t)p_{2,t}\right] \\ c_{2,t} = \frac{1}{2}\left[(1-\beta_t)p_{1,t} + (1+\beta_t)p_{2,t}\right] \end{cases} \quad (54)$$

Boundary conditions must also be prescribed.

The parental interval is expressed as:

$$\left[p_{min,t}, p_{max,t}\right] = \left[min\left(p_{1,t}, p_{2,t}\right), max\left(p_{1,t}, p_{2,t}\right)\right] \quad (55)$$

The permissible offspring interval is expressed as:

$$\left[p_{min,t} - 0.5\Delta_t, p_{max,t} + 0.5\Delta_t\right] \quad (56)$$

Ultimately, it must be confined within the search space [$xmin$,]:

$$c_{k,t} = min\left(max\left(c_{k,t}, x_{min}\right), x_{max}\right) \quad (57)$$

By deriving the probability density function (PDF) of βi, the impact of the crossover distribution index ηc from Equation (52) on parental crossover can be delineated, the PDF being given by:

$$f(\beta) = \frac{1}{2}(\eta_c + 1)\beta^{\eta_c},\ \beta \in [0,1] \quad (58)$$

As $\eta c \to 0$, (β) ≈ 1/2, yielding a roughly uniform distribution and shifting the algorithm into a strongly exploratory mode; As $\eta c \to \infty$, (β) → (1), causing the offspring to converge toward the parental midpoint.

In contrast to crossover, the mutation operator is applied exclusively to a single parent chromosome. Offspring emerge through independent perturbation of every gene locus along the chromosome. Polynomial mutation is utilized in this context, the parent chromosome being written as x=[x1,x2,...,xD], bounded by the search space $xmin \leq x_i \leq xmax$, $i \in$ {1,2,…,D}. The normalized proximity to the search-space boundaries is defined as:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\begin{cases} \delta_L = \dfrac{x_t - x_{min}}{x_{max} - x_{min}} \\ \delta_R = \dfrac{x_{max} - x_t}{x_{max} - x_{min}} \\ \delta_L + \delta_R = 1 \end{cases} \quad (59)$$

Here, δL designates the relative gap to the left boundary, while δR designates the relative gap to the right boundary.

For every gene position i, mutation materializes with probability pm, and the mutation indicator is given by:

$$m_t = \begin{cases} 1 & p_m \\ 0 & 1 - p_m \end{cases} \quad (60)$$

Here, mi represents a Bernoulli random variable.

Should a gene position be flagged for mutation, the perturbation magnitude is dictated by a random number $r \sim U(0,1)$ sampled from the parent, thereby permitting mutations of varying intensities. Let the perturbation magnitude be denoted by σi; accordingly:

$$\sigma_t = \begin{cases} \left[2u_t + (1-2u_t)(1-\delta_{L,t})^{\eta_m+1}\right]^{\frac{1}{\eta_m+1}} - 1 & u_t \le 0.5 \\ 1 - \left[2(1-u_t) + 2(u_t-0.5)(1-\delta_{R,t})^{\eta_m+1}\right]^{\frac{1}{\eta_m+1}} & u_t > 0.5 \end{cases} \quad (61)$$

Here, ηm denotes the mutation distribution index, which governs the aggregate strength of the mutation operator. This characteristic is reflected in the probability density function (PDF) of σi, defined as:

$$f_\sigma(\sigma) = \frac{\eta_m + 1}{2}(1 - |\sigma|)^{\eta_m}, \sigma \in [-1,1] \quad (62)$$

As $\eta m \to 0$:

$$\lim_{\eta_m \to 0} f(\sigma) = \frac{1}{2} \quad (63)$$

It degenerates into a uniform distribution, producing comparatively large mutation amplitudes.

As $\eta m \to \infty$:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\lim_{\eta_m \to \infty} f(\sigma) = \delta(\sigma) \quad (64)$$

It collapses toward the Dirac delta function, mutation amplitudes dwindling to zero. The (x) function is specified as:

$$\delta(x) = \begin{cases} +\infty & x = 0 \\ 0 & x \neq 0 \end{cases} \quad (65)$$

Resting on the perturbation magnitude obtained from Equation (64), the offspring generated by the mutation operator are computed as:

$$x_l' = \begin{cases} x_l + \sigma_l \cdot \Delta_{max} & if\ m_l = 1 \\ x_l & if\ m_l = 0 \end{cases} \quad (66)$$

Here, $\Delta max = xmax - xmin$ signifies the span of the search space. The probability density function (PDF) of the offspring gene value is given by:

Here, $(1-pm)\cdot\delta(y-xi)$ constitutes the exploitative component within the mutation process; $pm\cdot f\sigma(\frac{y-xi}{\Delta max})\cdot\frac{1}{\Delta max}$ makes up the exploratory component. The expected value of the offspring is analyzed as:

$$f_{x_i'}(y) = (1 - p_m) \cdot \delta(y - x_l) + p_m \cdot f_\sigma\left(\frac{y - x_l}{\Delta_{max}}\right) \cdot \frac{1}{\Delta_{max}} \quad (67)$$

Here, $(1-pm)\cdot\delta(y-xi)$ constitutes the exploitative component within the mutation process; $pm\cdot f\sigma(\frac{y-xi}{\Delta max})\cdot\frac{1}{\Delta max}$ makes up the exploratory component. The expected value of the offspring is analyzed as:

$$\mathbb{E}[x_l'] = \int y f_{x_i'}(y) dy = x_l \quad (68)$$

It can consequently be demonstrated that the mutation operator imparts no systematic bias, thus preserving solution quality. This attribute proves especially vital for the MPC-based UAV attitude controller, as it prevents the emergence of infeasible weighting coefficients. Upon concluding selection, crossover, and mutation, the Genetic Algorithm (GA) spawns a fresh population. This cycle repeats until the designated maximum generation count is reached, at which juncture the optimal weighting coefficients are harvested. The overarching procedure is portrayed in **Figure 5**.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

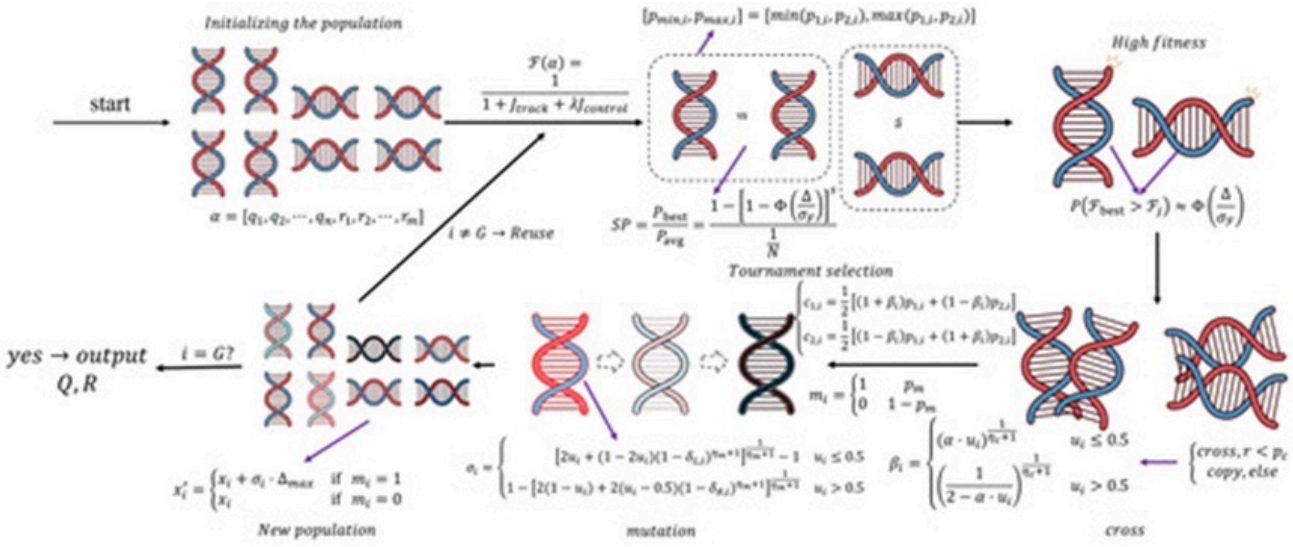


Figure 5. Flowchart of the Genetic Algorithm (GA).

Niching technique integration

By deploying the genetic algorithm introduced above, the weighting factors in the Model Predictive Controller (MPC) can be adaptively tuned. Yet, traditional genetic algorithms (GAs) remain vulnerable to premature convergence to suboptimal solutions. As evolution enters its later stages, the population commonly succumbs to early convergence, eroding variety—individuals end up congregated around local peaks, unable to break free and thus failing ever to locate the global best. Adjusting exploratory strength through ηc and ηm brings only minimal gains.

To combat this loss of population variety, the present work integrates a niching strategy **(Figure 6)**. The term niching draws from ecology's "niche," referring to the habitat space a species occupies in a given setting. Classical genetic algorithms (GAs) struggle considerably with multimodal optimization landscapes, habitually converging too soon or losing diversity entirely. Drawing on how natural species coexist and coevolve by occupying distinct niches, investigators have developed a family of niching techniques to sustain population diversity, prevent a single peak from monopolizing the search, and concurrently discover and retain multiple optima. The foundational principle involves mimicking niche segregation, resource apportionment, and competitive pressures within the algorithm's population, while using mechanisms such as fitness-sharing penalization to guide the emergence and maintenance of multiple stable subpopulations (niches), thereby addressing the core shortcomings of standard GAs.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

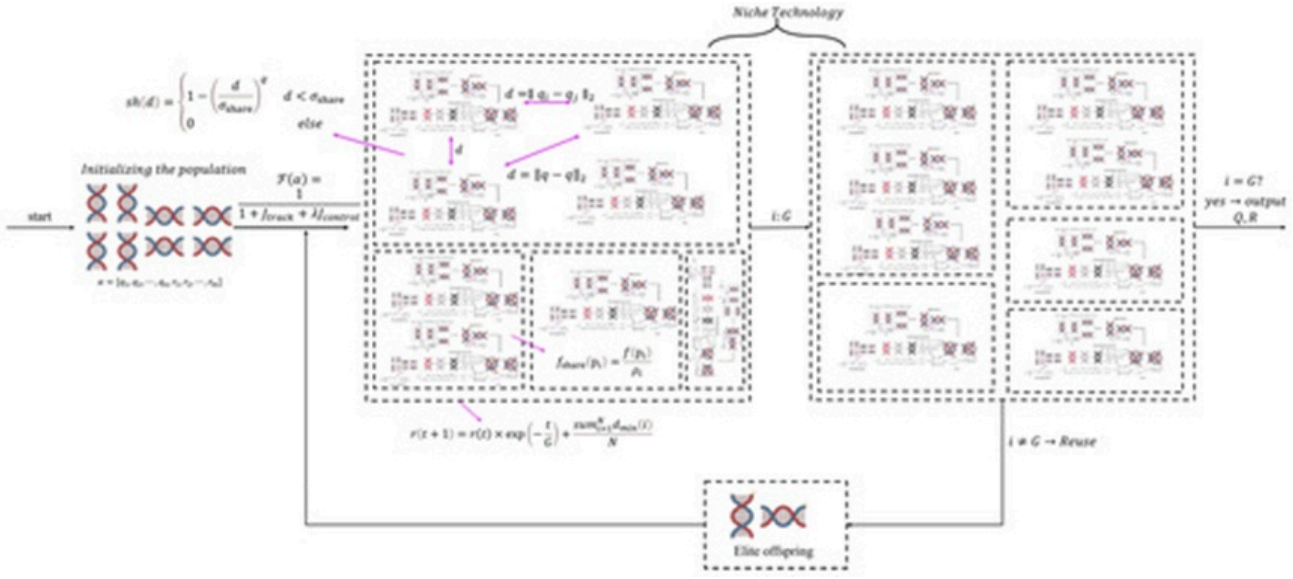


Figure 6. Structural overview of the Niching Genetic Algorithm (NGA).

Initially, the population is split into several isolated niches (subpopulations). Shared fitness values are then computed through the Euclidean distance dij between individuals alongside niche density estimates, which enables the clustering process. For two individuals, their Euclidean distance reads:

$$d = \|q_i - q_j\|_2 \quad (69)$$

Accordingly, the niche radius $\sigma share$ takes the form:

$$\sigma_{share} = \beta \cdot \frac{d}{\sqrt[p]{N}} \quad (70)$$

From this, the sharing function $sh(x)$ is constructed as:

Here, ϱ stands for the shape parameter. Individuals residing within a distance smaller than $\sigma share$ are bundled into a common niche.

Niche density for a given individual qi is assessed using the sharing function together with the niche radius:

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^{\varrho} & d < \sigma_{share} \\ 0 & else \end{cases} \quad (71)$$

Shared fitness is then expressed in terms of ϱi:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$\rho_i = \sum_{j=1}^{N} sh\,(d) \quad (72)$$

Left unchecked within the solution space, densely packed groups would otherwise overwhelm the search, dragging the algorithm into local traps, dismissing possibly superior candidates, and impoverishing population diversity. By performing shared-fitness calculations independently within each niche, crowded individuals incur a penalty. In contrast, those in sparse regions earn a competitive edge, thereby defending diversity and curbing early stagnation at local optima.

Seeking a compromise between swift convergence and maintained diversity, this investigation employs an adaptive niche radius. In the early generations, a larger niche radius fosters wide exploration and guards variety; as the run progresses, the radius is steadily narrowed, permitting intensive exploitation around regions inhabited by top individuals, thereby accelerating final convergence.

The adaptive radius update at step (t+1) follows:

$$r\,(t+1) = r(t) \times exp\left(-\frac{t}{G}\right) + \frac{sum_{i=1}^{N} d_{min}\,(i)}{N} \quad (73)$$

In this expression, G denotes the total iteration budget, and (i) captures the shortest distance from individual iii to every other member of the population (itself excluded). In UAV attitude control, if the search prematurely locks onto a confined region, the weighting parameters may be unable to accommodate dynamic phenomena such as rapid mass loss. The adaptive radius maintains a generous search envelope in early phases to avoid local-optimum entrapment, then steadily tightens later to accelerate the approach to the global optimum. Crucially, the Genetic Algorithm (GA) featured here is used exclusively for offline weight optimization before flight tasks. During live airborne operations, the controller merely draws on these pre-optimized weights for receding-horizon prediction and actuation, placing zero extra demands on the embedded hardware's real-time capacity.

System stability verification

Upon completing the design of the dual-loop MPC controller, theoretical stability verification becomes indispensable. As the system is discretized into a linear description around the small-angle hovering equilibrium and the cost is formulated quadratically, the Lyapunov method, combined with the feasibility property of receding-horizon optimization, can be applied within the predictive control paradigm to certify the closed-loop system's asymptotic stability. The assessment begins with a single-loop examination. For the outer-loop discrete model, one obtains:

$$x_{k+1} = A_{out}x_k + B_{out}u_k \quad (74)$$

The predictive cost function is given as:

$$J_{out} = x_N^{\top} Q_{f,out} x_N + \sum_{k=0}^{N-1} \left(x_k^{\top} Q_{out} x_k + u_k^{\top} R_{out} u_k\right) \quad (75)$$

With $Q_{out} \geq 0, t > 0$. Denote the optimal receding-horizon cost by:

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$V(x) = J_N^*(x) \quad (76)$$

Through the “shifted feasible sequence” construction technique, one can demonstrate:

$$V(x_{k+1}) - V(x_k) \leq -\left(x_k^{\mathrm{T}} Q_{out} x_k + u_k^{\mathrm{T}} R_{out} u_k\right) < 0 \quad (77)$$

Consequently, (x) serves as a Lyapunov function for the closed-loop system, confirming that the outer loop is asymptotically stable. By analogy, for the inner loop:

$$x_{k+1} = A_{in} x_k + B_{in} u_k \quad (78)$$

Introducing a quadratic cost function yields a parallel result.

Under the dual-loop arrangement, the outer loop’s generated attitude reference becomes the inner loop’s command signal. Provided the inner loop operates at a sampling rate at least 3–5 times quicker than the outer loop, and its tracking error exhibits input-to-state stability (ISS) with respect to reference signal fluctuations, then from the outer loop’s standpoint, the overall setup can be viewed as an MPC scheme under mild perturbations.

According to the small-gain theorem, the following criteria must hold:

$$\| G_{out} \|_\infty \cdot \| G_{in,err} \|_\infty < 1 \quad (79)$$

Under these, the complete dual-loop system retains asymptotic stability.

During feed dispensing, the overall mass (t) and moment of inertia (t) diminish progressively, causing the system matrices (Ak, Bk) to drift slowly over time. Should the rate of parameter change ϱ remain small enough that:

$$V_{k+1}(x^+) - V_k(x) \leq -\alpha \| x \|^2 + c\rho \| x \|^2 \quad (80)$$

Then, provided $\alpha - c\varrho > 0$, stability endures.

Here, the GA’s solution space Ω is defined as the collection of weight factor configurations obeying the subsequent strict constraints:

(1) $Q \geq Qmin \geq 0, \geq Rmin > 0$;

(2) The terminal control gain K is calculated from (A, Q,), with the terminal weight P given as the solution to the discrete algebraic Riccati equation (DARE):

© 2023 The Author(s). | Published by Agro Intelligence Research Press

$$(A+BK)^{\top}P(A+BK)-P=-(Q+K^{\top}RK) \quad (81)$$

(3) The terminal invariant set χf is positively invariant under closed-loop evolution x+=(A+BK)x and respects all imposed constraints;

(4) The dual-loop system's small-gain or bandwidth-separation condition is satisfied.

Define =A+BK. The DARE implies:

$$x_k^{\top}Px_k-x_{k+1}^{\top}Px_{k+1}=x_k^{\top}(Q+K^{\top}RK)x_k \quad (82)$$

Summation then yields:

$$\sum_{k=0}^{\infty}x_k^{\top}(Q+K^{\top}RK)x_k\le x_0^{\top}Px_0 \quad (83)$$

Letting (Q,):=(Q+K⊤RK), one arrives at:

$$\sum_{k=0}^{\infty}\|x_k\|^2\le\frac{\lambda_{max}(P)}{\alpha(Q,R)}\|x_0\|^2 \quad (84)$$

In other words, a greater (Q,) coupled with a diminished (P) produces swifter convergence along with reduced overshoot. Within Ω, ANRGA probes diverse pairings of Q and R. By exploiting multimodal search to sidestep local optima, it is more likely to uncover weight configurations that yield larger α and smaller P, thereby strengthening the Lyapunov decay property.

In brief, Section 3 presents an Adaptive Niche Radius Genetic Algorithm (ANRGA) that efficiently tunes MPC weighting parameters and, through stability analysis, establishes the closed-loop system's asymptotic stability. The ensuing chapter corroborates the proposed approach's effectiveness through comparative simulations with alternative controllers and optimization strategies.

Simulation Results and Analysis

Comparative evaluation across controllers

With the controller's mathematical formulation established, simulation trials for distinct controllers are executed within MATLAB (Intel i5 CPU, 16 GB RAM). Each of the four controllers is heuristically tuned. To ensure parity, their settings are refined under identical reference trajectories and wind disturbance profiles, and simulations are iterated until each one's error and energy-use indices plateau within acceptable thresholds. **Table 1** collates the outcomes. The reference path dictates a vertical climb to 4 m, a hover phase, followed by emulation of the authentic feeding sequence. Under a full payload, feeding initiates at 4 m altitude, with mass and inertial moments varying progressively throughout the maneuver.

Table 1. Wind disturbance simulation parameters.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

Variable	Description	Value
$Vmean$	Mean wind velocity	5.0 m/s
τ	Characteristic time constant	60.0 s
σ	Magnitude of large-scale fluctuations	1.0 m/s
$\sigma turb$	Small-scale turbulence intensity	0.3 m/s
ϱ	Air density parameter	1.225 kg/m3

To achieve a more faithful reproduction of the aerodynamic forces a UAV encounters during actual flight, a nonlinear Ornstein–Uhlenbeck (OU) process is embedded into the dynamic simulation environment in this work. The chosen stochastic model captures both the persistent temporal structure of the wind field and the fine-grained turbulent fluctuations, all while avoiding numerical instability. Expressed in continuous time, the model takes the form:

$$dv_w(t) = -\frac{1}{\tau}\left(v_w(t) - \mu(v_w)\right)dt + \sigma dW_t + \sigma_{turb} dB_t \quad (85)$$

In the above, $v(t)\in\mathbb{R}3$ signifies the wind velocity resolved in the inertial reference frame; τ sets the characteristic relaxation timescale of the OU dynamics; σ governs the amplitude of broadband stochastic forcing; $\sigma turb$ controls the strength of the turbulent component; and Wt together with Bt represent mutually independent standard Wiener processes. Diverging from the textbook OU formulation that imposes a constant mean, the variant applied here employs a drifting mean that continuously aligns with the instantaneous wind bearing:

$$\begin{aligned} \mu_k &= V_{mean} \cdot \frac{v_{w,k}}{\|v_{w,k}\| + \varepsilon} \\ \Delta v_k &= -\frac{v_{w,k} - \mu_k}{\tau}\Delta t + \sigma\sqrt{\Delta t}\,\xi_{1,k}, \xi_{1,k} \sim \mathcal{N}(0, I_3) \\ v_{w,k+1} &= v_{w,k} + \Delta v_k + \sigma_{turb}\xi_{2,k}, \xi_{2,k} \sim \mathcal{N}(0, I_3) \end{aligned} \quad (86)$$

To guard against a complete collapse of aerodynamic drag when the wind magnitude dwindles toward zero, a slight random nudge is applied at any step where $\|vw,+1\|$ drops below 0.5 m/s, thereby keeping the value inside a physically plausible envelope.

Shifting attention to the aerodynamics, the velocity of the oncoming airstream relative to the airframe is computed as:

$$v_{rel} = v_{uav} - v_w \quad (87)$$

The resulting drag force vector acting on the vehicle is modeled via a component-wise quadratic dependency:

$$F_{d,i} = -\frac{1}{2}\rho C_d A_i v_{rel,i}|v_{rel,i}|, i \in \{x, y, z\} \quad (88)$$

© 2023 The Author(s). | Published by Agro Intelligence Research Press

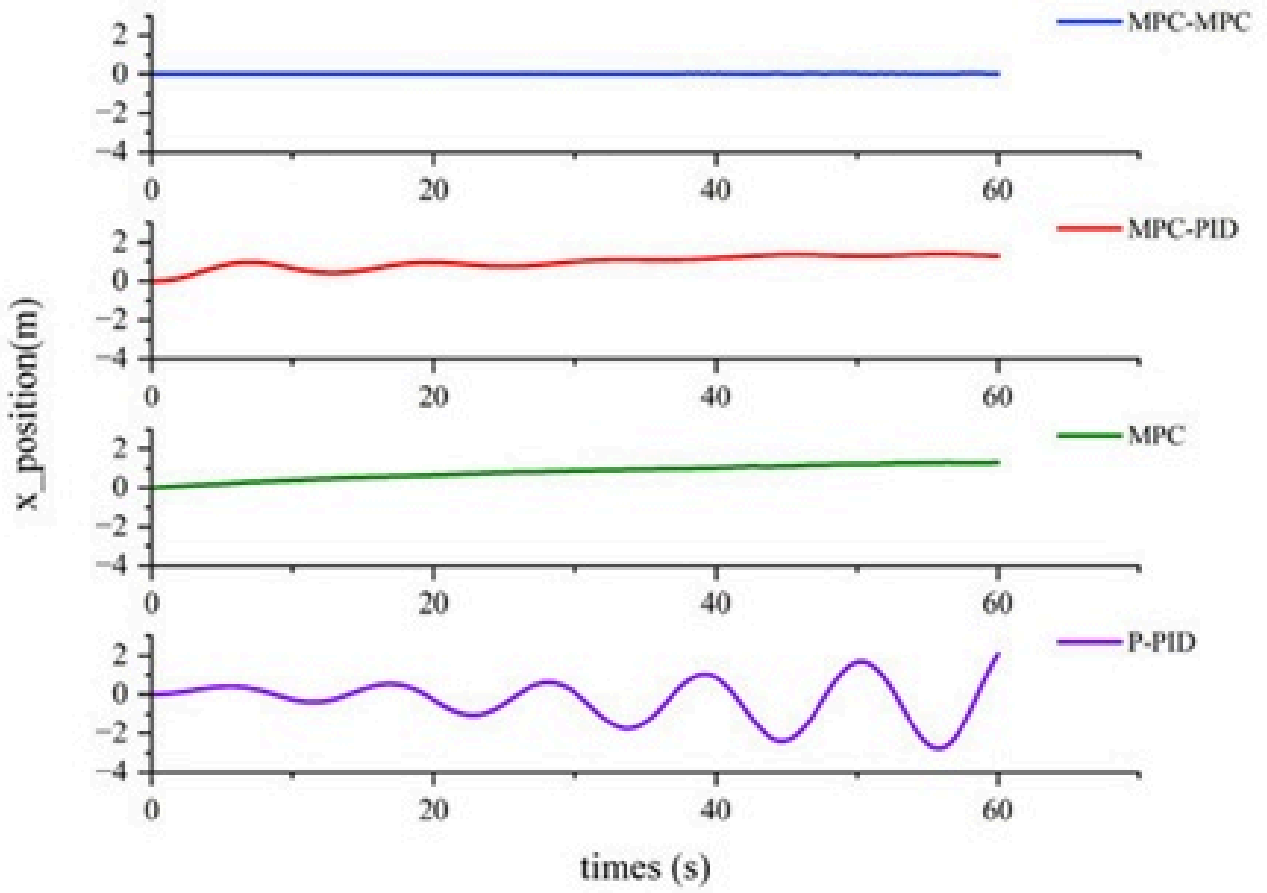
Within this expression, the ambient air density is fixed at ϱ=1.225 kg/m3; the effective reference area along the two lateral body axes is set to $CdAx$,=0.25; and along the vertical body axis to $CdAz$=0.30. The drag contribution Fd is subsequently combined with the propulsive thrust and the gravitational force to drive the translational state integration.

To trade off between physical realism and numerical efficiency, the parameter values listed in **Table 1** are used across all simulations reported here. The resulting disturbance model retains the low-frequency wandering behavior typical of mesoscale wind structures while also resolving the high-frequency content characteristic of microscale turbulence, offering a credible emulation of the perturbed flight environment a UAV would experience under a nominal 5 m/s wind.

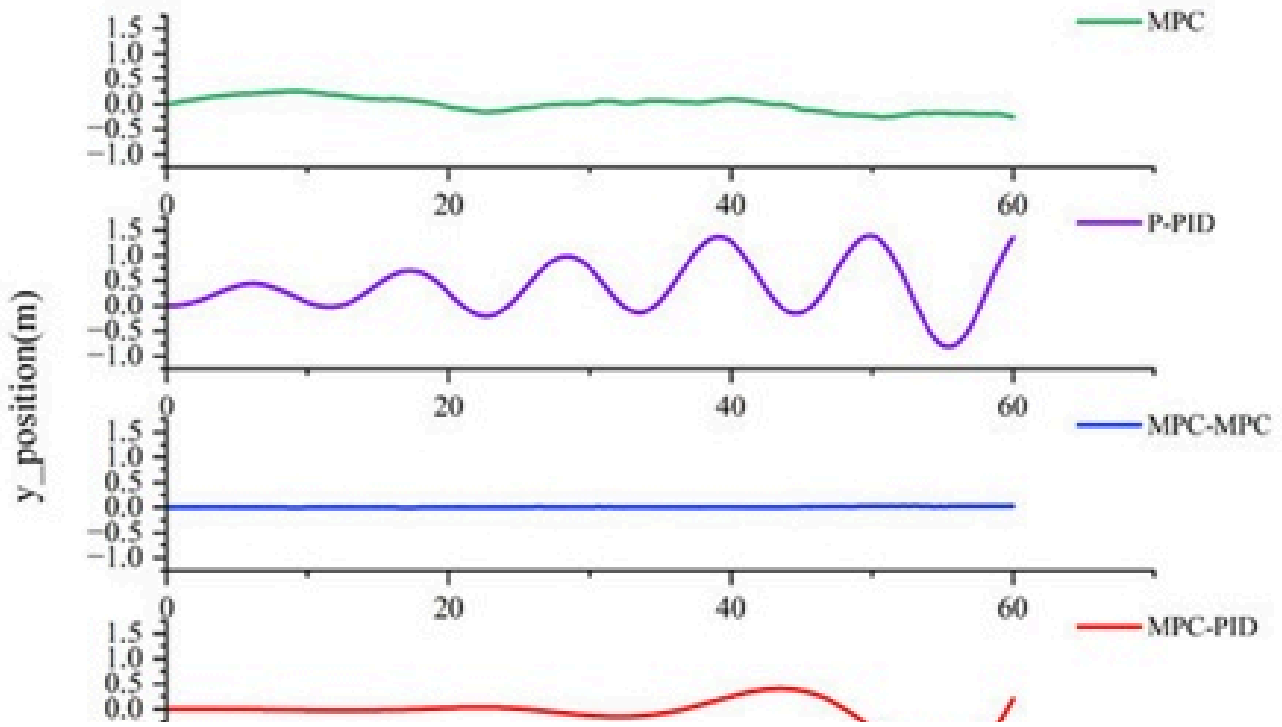
To safeguard against statistical artifacts in the experimental findings, each test configuration is independently repeated at least 30 times and run across three separate computing platforms to verify the consistency of controller behavior. In the context of aquaculture feed dispersal, the UAV's ability to hold its intended position in all three spatial dimensions directly determines how uniformly the feed is spread. Large tracking lapses translate into patchy distribution patterns. Hence, the analysis presented here centers on the triaxial positional accuracy of each control approach.

As shown in **Figure 7**, panel 7a reveals that the dual-loop MPC achieves the tightest x-axis regulation, with a worst-case excursion of only 0.018 m. The MPC-PID scheme ranks second with a peak deviation of 1.139 m, while the baseline MPC and P-PID arrangements register peak deviations of 1.278 m and upward of 3 m, respectively. It is worth noting that the P-PID controller exhibits substantial oscillatory behavior along the x-axis. Turning to panel 7b, the response patterns diverge across the spatial axes, yet the dual-loop MPC consistently secures the finest precision, confining its largest errors to beneath 0.5 m. The MPC-PID combination yields inferior y-axis tracking compared to its x-axis performance, with deviations lingering near 1 m. The classical MPC keeps control errors bounded within 0.5 m along the y-axis, whereas the P-PID controller continues to exhibit sizeable oscillatory swings. Looking at the climb phase (panel 7c), the P-PID controller generates the most severe overshoot, with the maximum deviation climbing close to 1.5 m. By contrast, the dual-loop MPC achieves rapid ascent with almost imperceptible overshoot; the remaining two controllers ascend briskly yet produce clearly visible overshoot.

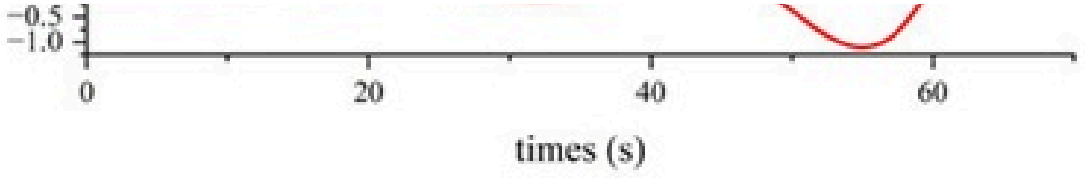
© 2023 The Author(s). | Published by Agro Intelligence Research Press



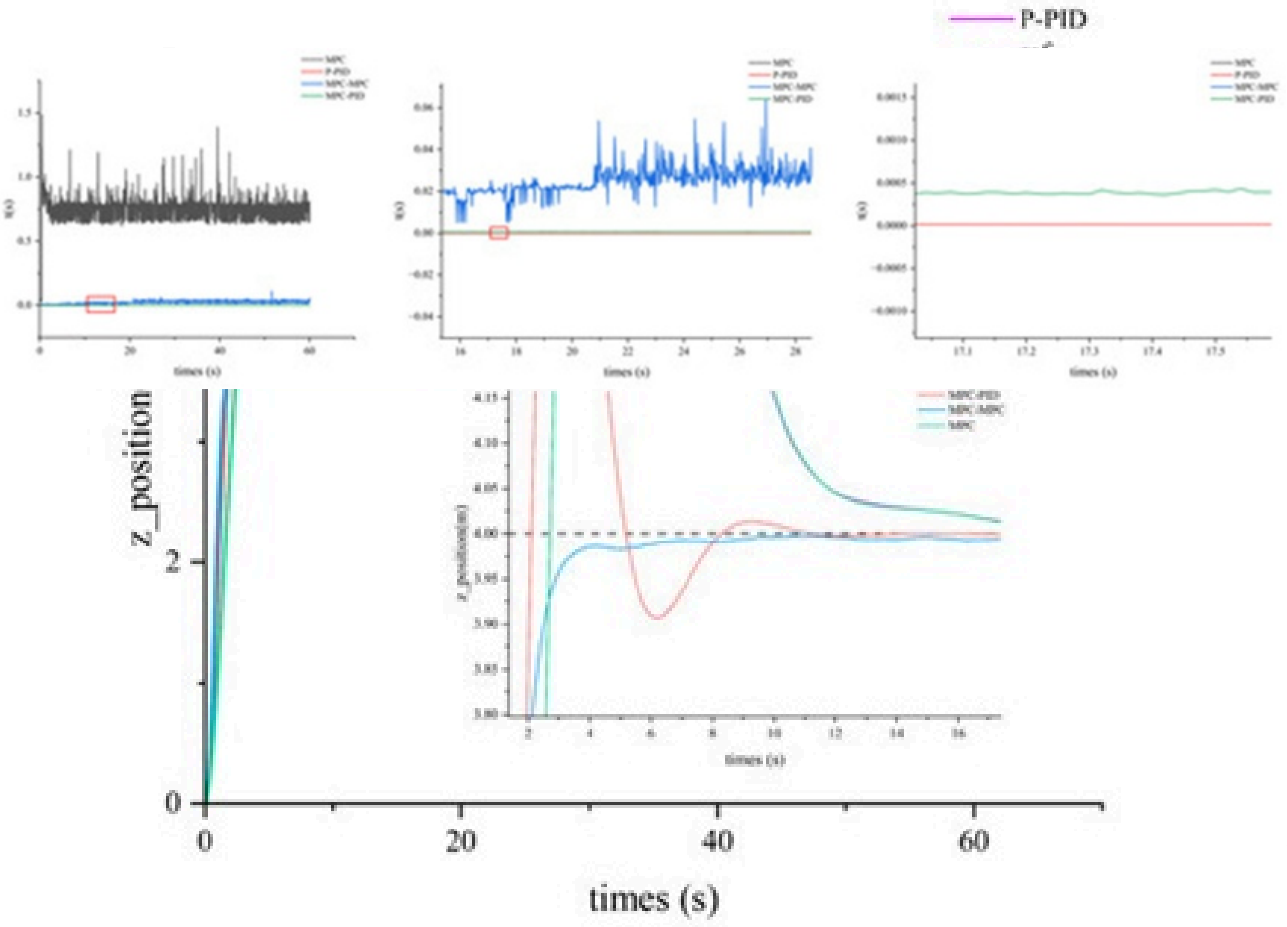
(a)



© 2023 The Author(s). | Published by Agro Intelligence Research Press



(b)



(c)

study.

Through simulating six candidate optimization routines and logging the resulting trajectories, the flight scenario is once again defined as a climb to and station-keeping at 4 m under the influence of wind disturbances. The integral of squared error (ISE), integral of absolute error (IAE), integral of time-weighted squared error (ITSE), and integral of time-weighted absolute error (ITAE) are chosen as quantitative yardsticks for assessing optimizer quality. The positional histories along x, y, and z are captured, the four indices are evaluated, and the findings are summarized in **Table 2**.

Table 2. Comparison of various errors in different optimization algorithms.

Algorithms	Average fitness	Position	ITAE	ITSE	IAE	ISE

© 2023 The Author(s). | Published by Agro Intelligence Research Press

GA	10.3654	x	0.305870191	0.001866838	0.038871587	0.000215173
		y	0.255337156	0.001201626	0.03098914	0.00014437
		z	1.774057728	4.011823729	3.332748304	10.12536554
PSO	10.3656	x	0.287396506	0.001826505	0.037749126	0.000209744
		y	0.219257914	0.000930779	0.027695836	0.000116342
		z	1.758141291	4.012470412	3.334065471	10.12476443
GOA	10.3652	x	0.516707276	0.005760842	0.062688509	0.000610794
		y	0.19827557	0.00087144	0.023615554	0.000108194
		z	1.843790076	4.012388443	3.344654568	10.12476715
POA	10.3654	x	0.237062265	0.001201294	0.03938835	0.000191671
		y	0.217212464	0.000914783	0.027202972	0.000113357
		z	1.76787313	4.01277184	3.33610989	10.1254051
ACOR	10.3660	x	0.454516737	0.00498513	0.054400741	0.000520045
		y	0.135319912	0.00040151	0.017003337	0.000132314
		z	1.813609035	4.016482238	3.349032802	10.1257645
ANRGA	10.3648	x	0.292830781	0.001807915	0.037597793	0.000177635
		y	0.228636622	0.000467757	0.018234861	0.000106948
		z	1.76519455	4.01196195	3.33351663	10.1247436

A review of the compiled metrics indicates that the dual-loop MPC controller, whose weights are determined via the adaptive niche radius technique, yields the lowest mean fitness and outperforms the alternatives across every error-based benchmark. Regarding convergence dynamics **(Figure 9)**, the fitness reduction trajectories produced by the different optimizers are plotted side by side.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

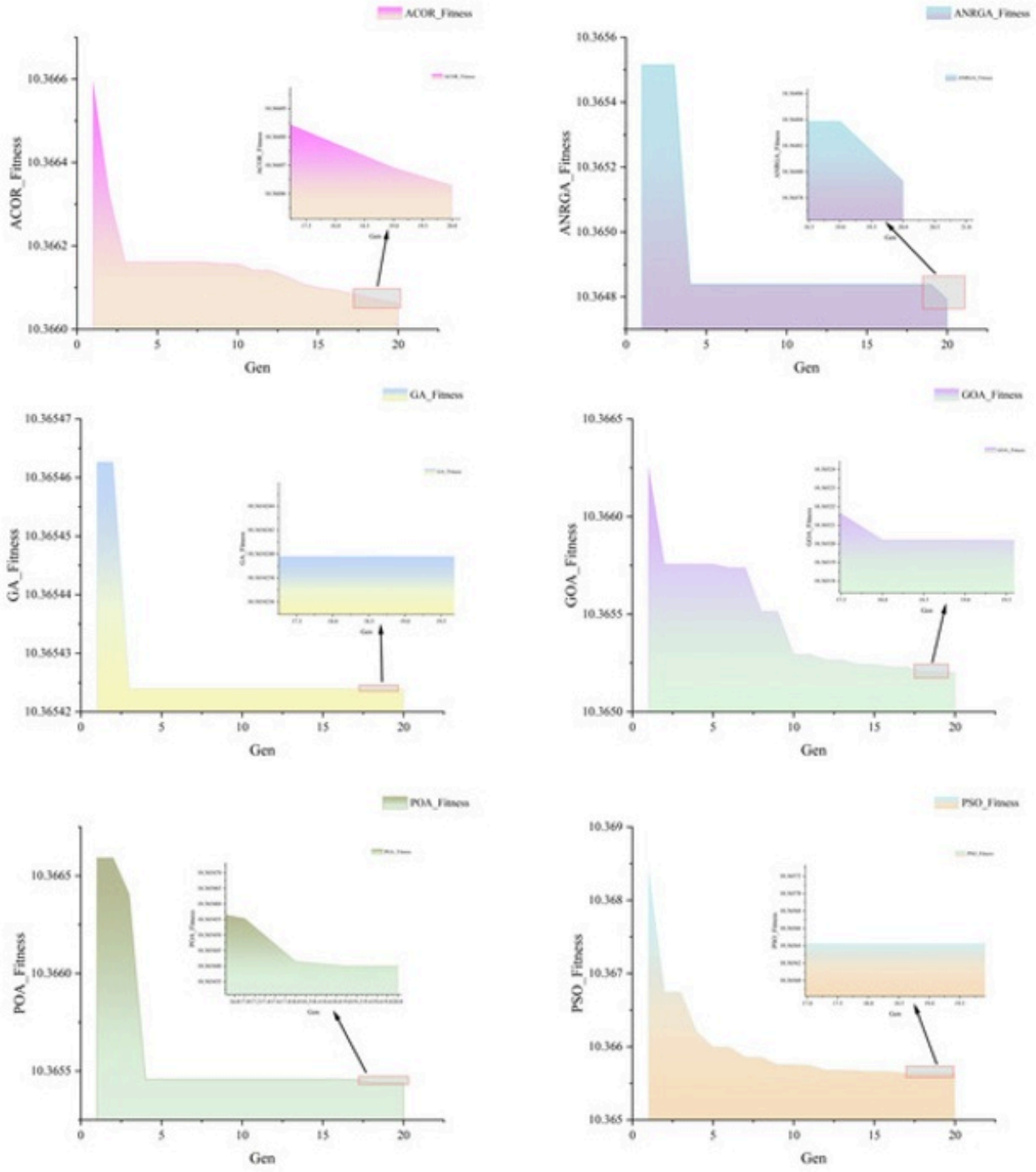


Figure 9. Fitness values corresponding to different algorithms. The sustained downward trend in later generations, alongside visible population spread, mirrors the use of shared fitness and an adaptive niche radius that initially promotes exploratory behavior and subsequently transitions into focused exploitation, aligning with the mechanism detailed in Section 4.2.

Cross-algorithm comparisons indicate that the Genetic Algorithm, Pelican Algorithm, and the Adaptive Niche Radius Genetic Algorithm all converge with the greatest swiftness; the Grasshopper Algorithm does not embark on meaningful convergence until past the 16th generation; the Ant Colony Algorithm undergoes an abrupt fitness drop at generation 3 but thereafter drifts into sluggish convergence beyond generation 10; and Particle Swarm Optimization remains mired in

© 2023 The Author(s). | Published by Agro Intelligence Research Press

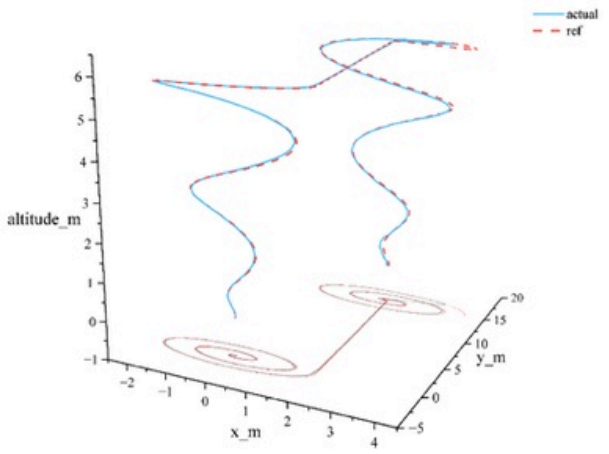
slow convergence through the first 17 generations. Benchmarked against the Ant Colony Algorithm, the introduced method achieves a broadly comparable rate of convergence yet attains lower final fitness values and continues to exhibit a pronounced downward trend in the closing generations. For UAV flight control—even for heavily loaded, comparatively sluggish, feeding-oriented platforms—rapid optimizer convergence is a principal performance requirement. Crucially, however, speed of convergence must not be obtained by sacrificing diversity; otherwise, premature arrest at local optima becomes a genuine risk. Grounded in this reasoning, the proposed Adaptive Niche Radius Genetic Algorithm successfully navigates the trade-off between fast convergence and the imperative of avoiding premature stagnation.

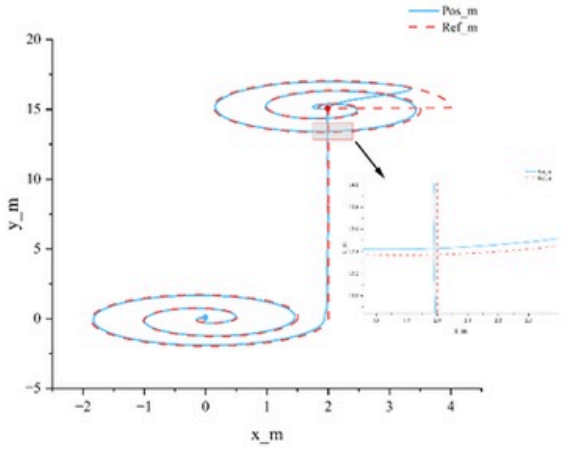
This investigation employs a dual-loop MPC structure in which the outer loop handles translational position or velocity regulation. In contrast, the inner loop maintains attitude and angular rate stabilization despite mass variations and near-ground wind effects. The search over the Q and R weighting matrices yields a multimodal surface owing to cross-loop coupling and the gradual time-dependence of inertial properties during the feeding phase. The offline ANRGA MPC framework is a natural fit for this setup because the evolutionary search is entirely conducted before flight, and the airborne controller reduces to a conventional constrained dual-loop MPC that solves a deterministic quadratic program at each control cycle. The manuscript explicitly states that the GA is reserved exclusively for pre-flight weight optimization and thus contributes no additional runtime overhead. At the same time, the online computational footprint is confined to the receding-horizon QP solve. Stability is separately analyzed for the outer and inner loops and their coupled interaction, drawing on Lyapunov theory, a small-gain condition, terminal constraints, and a terminal weight matrix derived from the DARE, thereby providing an analytical foundation for closed-loop asymptotic stability under the dual-loop arrangement. Within this admissible parameter domain, the ANRGA maintains population diversity via niching and progressively contracts the niche radius to accelerate convergence, thereby countering premature convergence in the multimodal search—constituting the specific methodological contribution advanced in the manuscript. The results already presented confirm that the ANRGA-tuned dual-loop MPC attains the lowest average fitness and consistently superior error-based indices relative to alternative metaheuristic optimizers under an equivalent evaluation budget. At the same time, the convergence profiles exhibit a steeper descent without erosion of diversity, lending empirical support to the final design selection for initial deployment. By contrast, a reinforcement learning pipeline would require a distinct training phase that involves interaction data or explicit domain randomization, along with additional mechanisms for constraint satisfaction and policy certification. Yet, the manuscript does not adopt such an architecture. In view of the present deployment priorities favoring analytical stability guarantees, explicit constraint enforcement, and deterministic execution timing on resource-constrained avionics hardware, the offline ANRGA MPC configuration aligns squarely with the requirements articulated earlier in the paper.

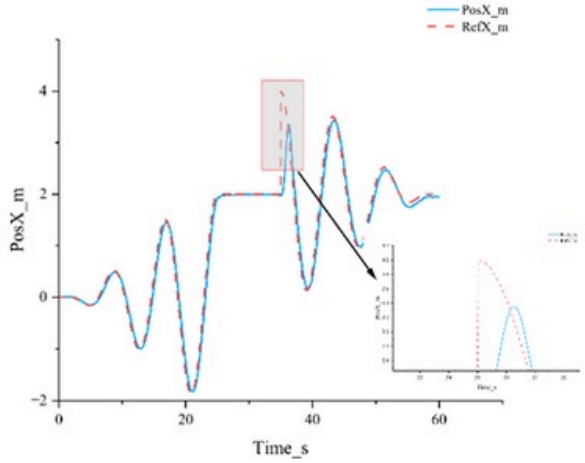
The cost criterion blends the normalized path-following error with the accumulated actuation expenditure over the receding window. Every candidate weight configuration is required to meet the admissibility conditions stemming from the terminal controller, the terminal penalty derived via the DARE, the terminal positively invariant region, and the small-gain criterion. Weight sets that fall outside these constraints are either discarded outright or assigned a heavy penalty. The initial gene pool is seeded within physically sensible ranges and may contain individuals carried over from earlier calibration sessions. Conventional selection, crossover, and mutation operations are applied, all confined within a fixed allotment of generations. Variety across the population is preserved through a shared-fitness mechanism. The niche radius is governed by a schedule that decreases monotonically as generations advance, so that wide-ranging exploration dominates in the early phase and gradually yields to focused refinement. During flight, only the deterministic quadratic programming core of the dual-loop MPC is executed in real time.

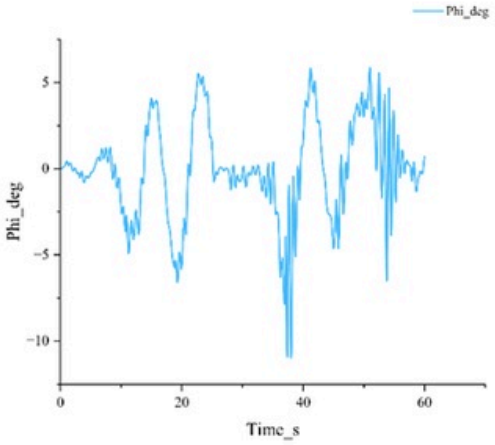
To furnish additional evidence of the proposed algorithm’s edge, the UAV is evaluated on several distinct flight paths. **Figure 10** shows the path synthesized by Algorithm A1, which forms a helix. This curve couples changes in elevation with continuous turning, placing amplified demands on the coordination between translational motion and rotational orientation. The outcomes are displayed in the figure as follows: the red dashed curve denotes the commanded route, the blue solid curve traces the actual flown route, the blue dot marks the starting point, and the red dot indicates the terminal point.

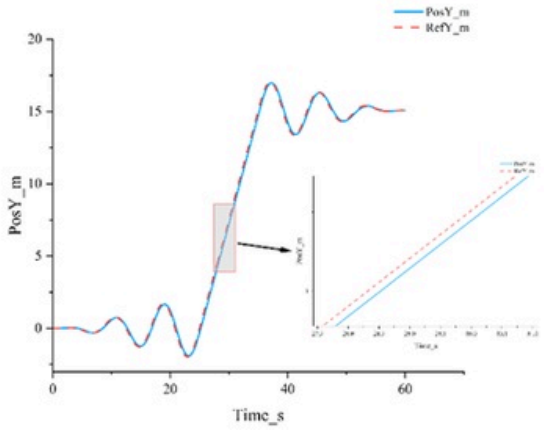
© 2023 The Author(s). | Published by Agro Intelligence Research Press

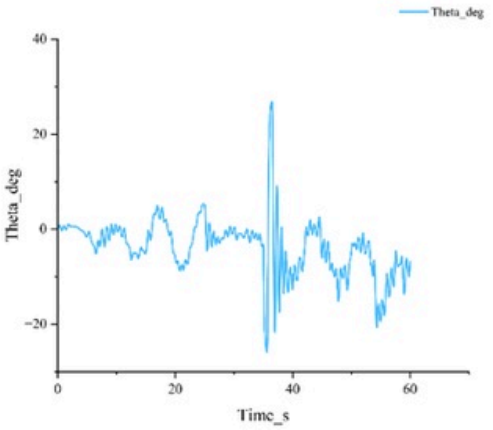


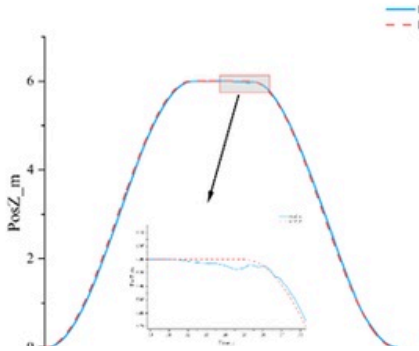


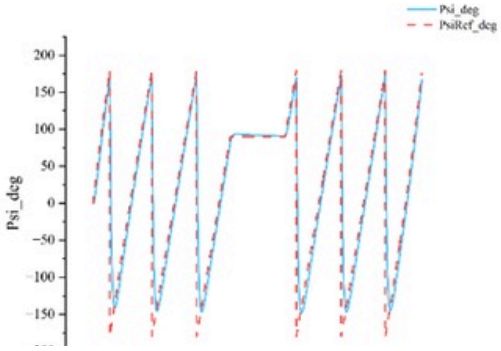












© 2023 The Author(s). | Published by Agro Intelligence Research Press

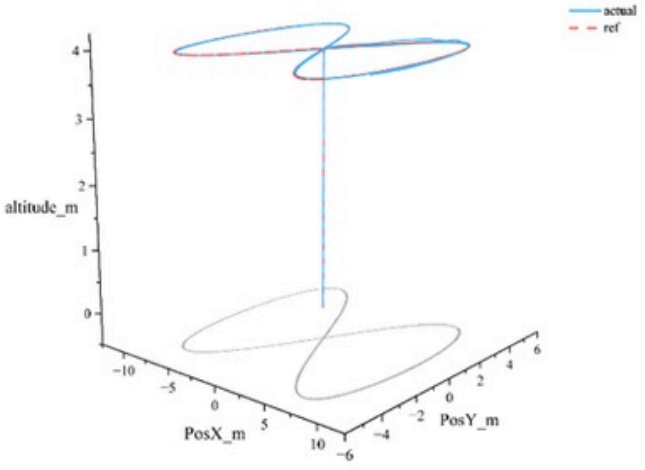


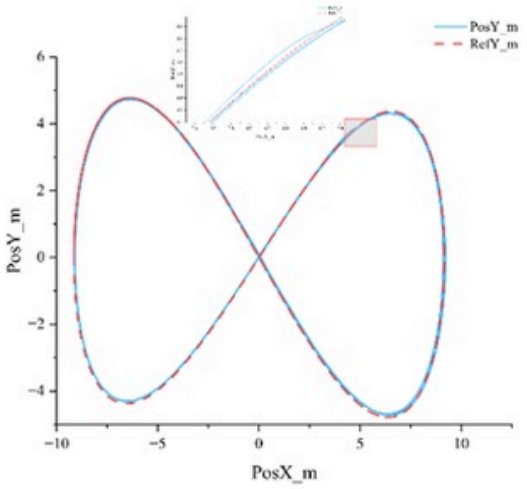
During the helix-tracking experiment, both the full spatial curve and its planar x–y footprint reveal strong agreement between the vehicle's realized motion and the prescribed trajectory. Although faint phase lags are discernible at segments of tight curvature, such mismatches stay small in magnitude, recede quickly, and do not accrue into an unbounded drift. This outcome underscores the controller's capacity to cope with the pronounced nonlinearities and strong channel interdependence typical of helical flight, delivering both favorable transient characteristics and precise steady-state holding. In parallel, the altitude channel proceeds smoothly through climbs and descents with virtually zero overshoot, reflecting the equilibrium maintained between thrust apportionment and total lift generation.

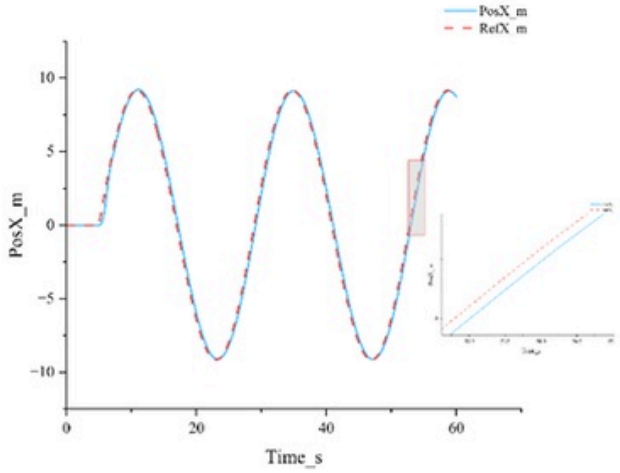
The rotational responses further highlight the coordinated operation of the control framework under demanding trajectories. Through the helical maneuver, roll and pitch oscillate periodically within bounded ranges, registering only momentary spikes at points of acceleration or deceleration change that subsequently subside without delay. The yaw channel tracks the commanded heading tightly, ensuring the airframe continuously orients itself along the tangent direction. Considering the position and attitude results together, the system exhibits substantial resilience and convergent behavior in helical path following, consequently lending reliable support for UAV tasks conducted in convoluted settings.

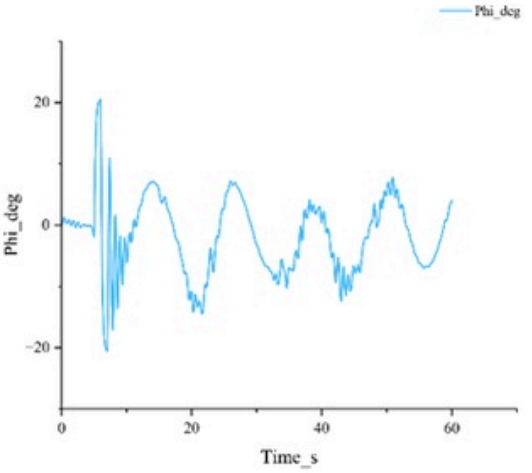
Figure 11 corresponds to an aerial figure-eight trajectory. The full spatial trace and its x–y projection show that the actual flown path conforms geometrically to the reference curve around both lobes and the central crossing zone, with no indication of progressive drift. The time evolutions of the x and y channels reveal slight phase advancement (or retardation) along with modest undershoot (or overshoot) at curvature inflection points and where the path intersects itself; nevertheless, these deviations are corrected promptly, matching the transient centripetal requirement $ac = v2/R$ dictated by the figure-eight geometry. The vertical channel reaches the commanded operating altitude within seconds and thereafter remains stable, showing only minute ripples at the uppermost point and no appreciable overshoot, indicating correct thrust balancing and suitable damping settings in the altitude loop. Regarding the attitude channels, the roll angle ϕ and pitch angle θ increase moderately at the loop extremities and high-curvature segments to supply the lateral centripetal component demanded by the turn, with any spikes and ripples proving short-lived and kept within manageable amplitudes. The yaw angle ψ stays close to the zero reference, exhibiting only insignificant variations, thereby confirming the decoupling performance of the heading-hold strategy. As observed in the helical scenario, tracking errors concentrate mainly within the brief intervals of "elevated curvature or speed change." Their amplitudes remain modest and decline rapidly, reinforcing the conclusion that the system exhibits strong convergence and robustness.

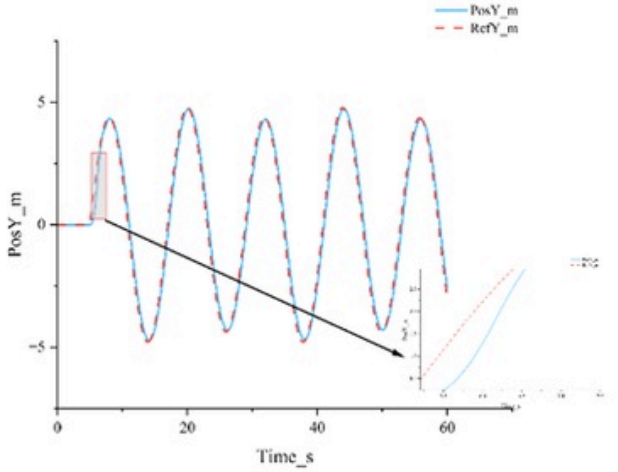
© 2023 The Author(s). | Published by Agro Intelligence Research Press

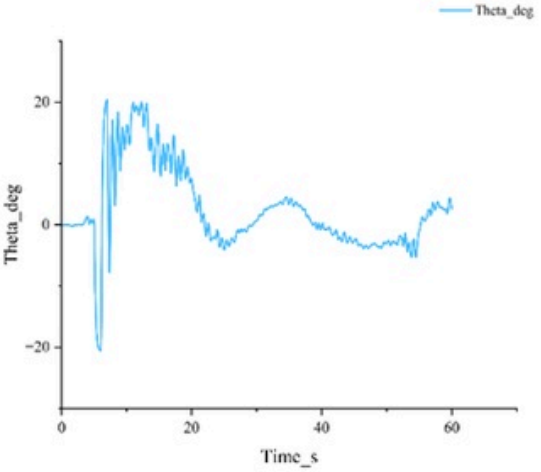


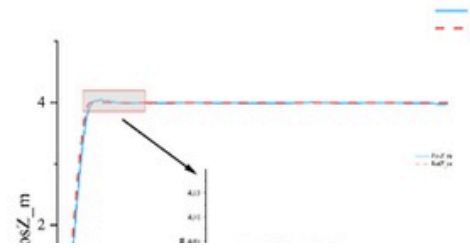












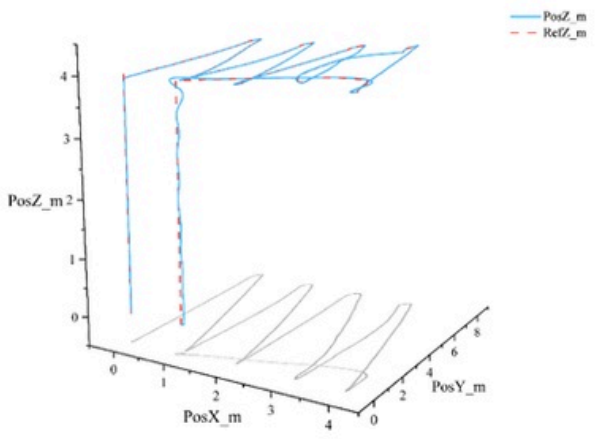


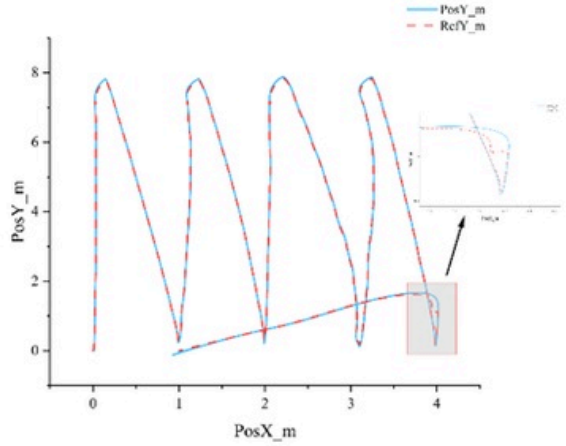
© 2023 The Author(s). | Published by Agro Intelligence Research Press

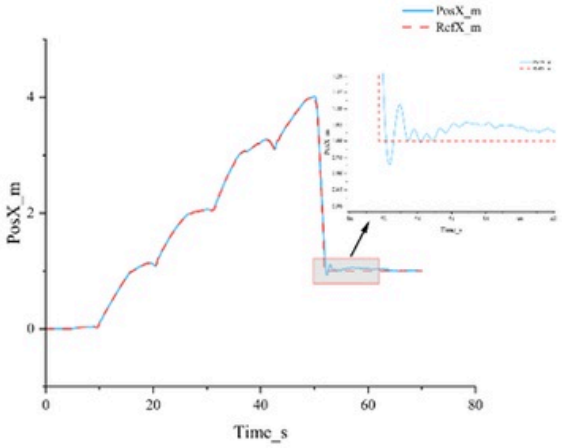


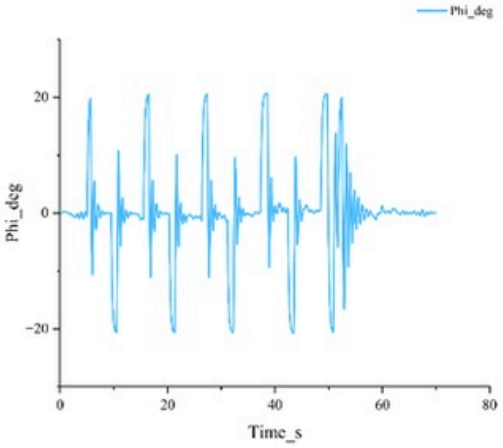
In parallel, the yaw channel responds to discrete heading-step commands at each reversal point and settles quickly thereafter. In contrast, the roll signal generates matched peak–valley excursions around the corners to deliver the lateral centripetal force indispensable for turning. The altitude channel locks onto a plateau after reaching the designated service layer. It reacts distinctly to mid-maneuver small-amplitude altitude tweaks, thereby confirming that the operational rhythm of “constant-altitude traversal interspersed with localized altitude corrections” is faithfully enacted. The associated three-axis positional profiles **(Figure 12)** exhibit transient, low-amplitude oscillations following turns that attenuate over time and return the platform to the prescribed lane.

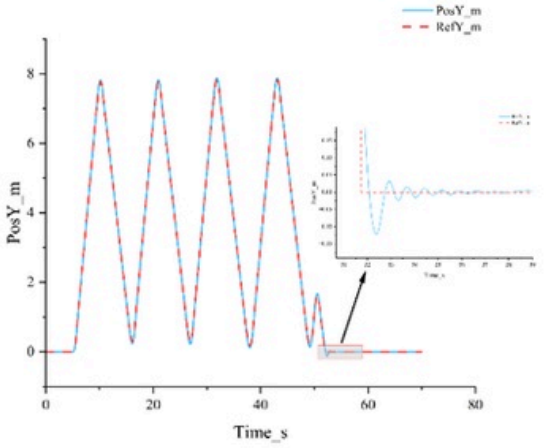
© 2023 The Author(s). | Published by Agro Intelligence Research Press

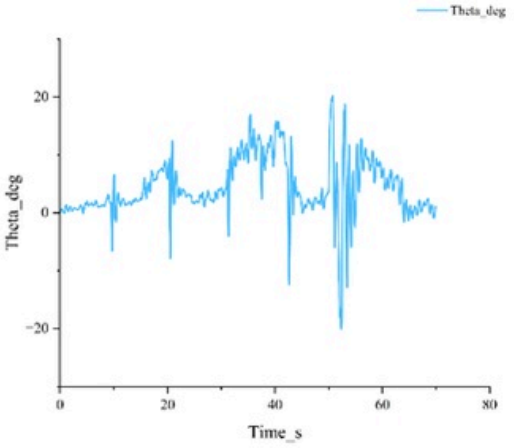


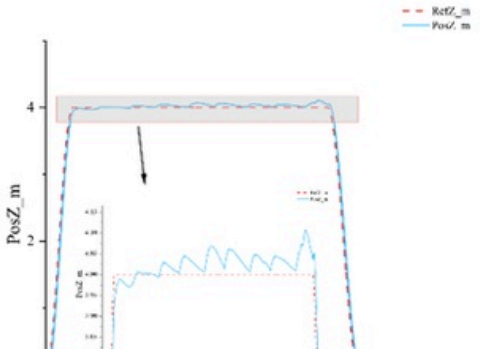


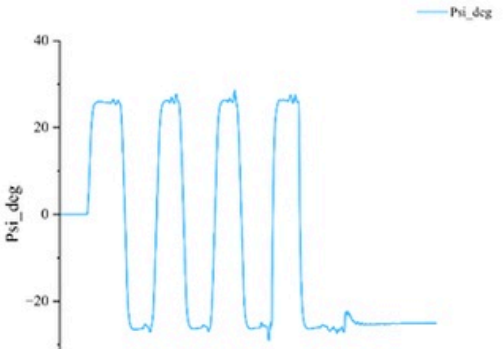












© 2023 The Author(s). | Published by Agro Intelligence Research Press



forward displacement along straight corridor sections. In contrast, around corners, it dispatches heading-step directives, coupled with saturated roll (and pitch) calls, to execute the “decelerate—pivot—reaccelerate” sequence. The inner-loop attitude peaks are bracketed within the cornering episodes and decay soon after the turn finishes, without triggering notable altitude excursions, which testifies that the thrust distribution policy successfully isolates attitude regulation from altitude maintenance. The zoomed-in view indicates that the residual oscillations in x and y after a turn endure for several periods but gradually fade, in accordance with the decay of the yaw step command.

In brief, the multi-scenario simulation evidence presented in Section 4 establishes that the proposed scheme significantly surpasses the reference methods in both positional fidelity and convergence properties, thereby laying a robust foundation for subsequent conclusions and practical deployment.

Results and Discussion

This research addresses the control challenges inherent to feeding-category UAVs operating under time-varying mass, time-varying inertia, and wind-induced perturbations. It advances a dual-loop Model Predictive Control (MPC) methodology integrated with an Adaptive Niche Radius Genetic Algorithm (ANRGA). By embedding a time-varying dynamical description that captures payload diminution together with low-altitude wind disturbances, a decoupled structure consisting of an outer-loop position MPC and an inner-loop attitude MPC is erected, with ANRGA deployed to calibrate the controller weight matrices.

Simulation outputs confirm that ANRGA–MPC yields remarkable lateral trajectory-tracking quality: the RMSEs along the x- and y-axes are 0.00172 m and 0.00133 m, respectively, corresponding to an accuracy enhancement of roughly 45% compared with the worst-performing algorithm. Evaluated via the IAE metric, ANRGA scores 0.0182 along the y-axis, substantially surpassing GA (0.0310) and POA (0.0272). Along the vertical z-axis, the RMSE remains at 0.411 m. Despite the influence of diminishing mass, ANRGA upholds robust performance.

A theoretical treatment resting on the small-gain theorem confirms the closed-loop system’s asymptotic stability, further validating the feasibility of the proposed scheme. On balance, the advocated approach exceeds the competing algorithms in path fidelity, robustness, and convergence characteristics, delivering a viable solution for precision feed dispensing and complex-environment tasks undertaken by aquaculture UAVs, while also showing potential for extension to agricultural crop spraying and coordinated multi-UAV operations.

Conclusion

This study is conducted entirely within a simulation framework, and the genetic search is performed strictly offline. At the same time, the controller running on board is the constrained dual-loop MPC that solves a deterministic quadratic program at each sampling interval. Toward eventual hardware deployment, we will first freeze the control period. The solver time quota, then pick prediction and control horizons that match the airframe bandwidth and the available cycle duration, formulate the optimization using sparse or block-structured matrix representations, warm-start from the preceding optimal trajectory, produce target-customized code employing statically allocated memory workspaces, and supervise execution timing through a watchdog that invokes a safe contingency—for instance, holding the last feasible actuator command or reverting to a previously certified backup law—should a solver exceed its allotted time on rare occasions. This procedure keeps the control algorithm intact while making cycle-to-cycle timing and memory footprint predictable on resource-constrained avionics. A graduated verification sequence will be pursued, moving from software-in-the-loop simulations integrated with the flight software stack, to processor-in-the-loop timing assessments on the target embedded device, then to bench-level validation of actuator and sensor latencies, and ultimately to limited outdoor flight

© 2023 The Author(s). | Published by Agro Intelligence Research Press

tests, with acceptance gauged by per-step solver completion within the prescribed time, the absence of any deadline violations over prolonged runs, and tracking accuracy consistent with the simulated wind disturbance models. For multi-agent configurations the architecture can be extended by augmenting the outer loop with separation and collision-avoidance constraints or by incorporating a reference governor that accounts for neighbor vehicle states, while the inner attitude loop stays unchanged; practical fielding then contends with constrained communication bandwidth, dropped data packets, and unsynchronized updates, addressed through either decentralized MPC with locally exchanged state information or a hierarchical scheme wherein a coordinating node disseminates feasible waypoints. Each platform solves its own MPC problem, retaining genetic optimization as an offline process executed per vehicle class. Moving beyond the linearized flight envelope, we plan to investigate nonlinear MPC or an LPV-based alternative to relax the small-angle approximations for aggressive maneuvering and large attitude displacements, adhering to the same deployment discipline of fixed time budget, structure-exploiting numerical solvers, warm starts, and a watchdog with safe fallback to trade off accuracy against computational cost. Finally, to boost adaptivity under mass variation and wind disturbances, we aim to incorporate vision-based or other sensor-driven feedback for estimating states, disturbances, and model parameters—for example, visual odometry, airspeed sensing, or wind field estimation, combined through a Kalman-type filter with event-triggered measurement updates—so that model coefficients and disturbance estimates are refreshed online without any modification to the optimization formulation. This roadmap does not modify the reported results, but it clearly illustrates how the current design can transition from simulation to embedded implementation and scale to coordinated multi-agent missions.

Acknowledgements

None

Conflict of interest

None

Financial support

Construction Projects funded this research in Specific Universities, grant number 2023B10564002; Jiangsu Provincial Key Research and Development Program, grant number BE2022366; The Open Fund of Key Laboratory of Smart Agricultural Technology (Yangtze River Delta), Ministry of Agriculture and Rural Affairs, grant number KSAT-YRD2024008.

Ethics statement

None

Received: 17 Jan 2022 Revised: 24 Mar 2022 Accepted: 11 Apr 2022
Published online: 25 May 2023

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at https://creativecommons.org/licenses/by-nc-sa/4.0/.

References

1. Subasinghe R, Soto D, Jia J. Global aquaculture and its role in sustainable development. Rev Aquac. 2009;1:2-9.

2. Béné C, Arthur R, Norbury H, Allison EH, Beveridge M, Bush S, et al. Contribution of fisheries and aquaculture to food security and poverty reduction: Assessing the current evidence. World Dev. 2016;79:177-96.

3. Genschick S, Kaminski AM, Kefi AS, Cole SM. Aquaculture in Zambia: An Overview and Evaluation of the Sector's Responsiveness to the Needs of the Poor. Lusaka, Zambia: CGIAR Research Program on Fish Agri-Food Systems; 2017.

4. Kassam L. Assessing the Contribution of Aquaculture to Poverty Reduction in Ghana. Ph.D. Thesis, SOAS University of London, London, UK; 2013.

5. Cai J, Leung P, Hishamunda N. Commercial Aquaculture and Economic Growth, Poverty Alleviation and Food Security. Assessment Framework. Rome, Italy: FAO; 2009.

6. Shamsuzzaman MM, Mozumder MMH, Mitu SJ, Ahamad AF, Bhyuian MS. The Economic Contribution of Fish and Fish Trade in Bangladesh. Aquac Fish. 2020;5:174-81.

7. Sribhibhadh A. Role of aquaculture in economic development within southeast asia. J Fish Board Can. 1976;33:899.

8. Zhang L, Li B, Sun X, Hong Q, Duan Q. Intelligent Fish Feeding Based on Machine Vision: A Review. Biosyst Eng. 2023;231:133-64.

9. Bao J, Li D, Qiao X, Rauschenbach T. Integrated Navigation for Autonomous Underwater Vehicles in Aquaculture: A Review. Inf Process Agric. 2020;7:139-51.

10. Wei Y, Wei Q, An D. Intelligent Monitoring and Control Technologies of Open Sea Cage Culture: A Review. Comput Electron Agric. 2020;169:105119.

11. Wu Y, Duan Y, Wei Y, An D, Liu J. Application of Intelligent and Unmanned Equipment in Aquaculture: A Review. Comput Electron Agric. 2022;199:107201.

12. Ma Y, Xu Y, Sun L, Chen Z, Feng J, Wu X. Development Status and Prospect of Aviation Plant Protection in China. Agrochemicals. 2022;61:469-77.

13. De Lima RLP, Paxinou K, Boogaard FC, Akkerman O, Lin F-Y. In-Situ Water Quality Observations under a Large-Scale Floating Solar Farm Using Sensors and Underwater Drones. Sustainability. 2021;13:6421.

14. Hu P, Zhang R, Yang J, Chen L. Development status and key technologies of plant protection uavs in China: A review. Drones. 2022;6:354.

15. Liu H, Chen J, Xiang Y, Geng H, Yang X, Yang N, et al. Improving UAV hyperspectral monitoring accuracy of summer maize soil moisture content with an ensemble learning model fusing crop physiological spectral responses. Eur J Agron. 2024;160:127299.

16. Wang Y, Zhang Y, Ma F, Tian X, Ge S, Man C, et al. Research on precise feeding strategies for large-scale marine aquafarms. J Mar Sci Eng. 2024;12:1671.

17. Chen H, Lan Y, Fritz BK, Clint Hoffmann W, Liu S. Review of agricultural spraying technologies for plant protection using unmanned aerial vehicle (UAV). Int J Agric Biol Eng. 2021;14:38-49.

18. Ijaz S, Shi Y, Khan YA, Khodaverdian M, Javaid U. Robust adaptive control law design for enhanced stability of agriculture uav used for pesticide spraying. Aerosp Sci Technol. 2024;155:109676.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

19. Ahmadi K, Asadi D, Nabavi-Chashmi S-Y, Tutsoy O. Modified adaptive discrete-time incremental nonlinear dynamic inversion control for quad-rotors in the presence of motor faults. Mech Syst Signal Process. 2023;188:109989.

20. Sun R, Ahn CK, Liu D, Wang W, Zhang C. Near-Asteroid spacecraft formation control with prescribed-performance: A dynamic event-triggered reinforcement learning control approach. Aerosp Sci Technol. 2025;161:110138.

21. Xu R, Ozguner U. Sliding Mode Control of a Quadrotor Helicopter. In: Proceedings of the 45th IEEE Conference on Decision and Control (CDC); 2006; San Diego, CA, USA. New York, NY, USA: IEEE; 2006. p. 4957-62.

22. Gautam D, Ha C. control of a quadrotor using a smart self-tuning fuzzy pid controller. Int J Adv Robot Syst. 2013;10:380.

23. Pounds PEI, Bersak DR, Dollar AM. Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control. Auton Robot. 2012;33:129-42.

24. Kapnopoulos A, Alexandridis A. A cooperative particle swarm optimization approach for tuning an mpc-based quadrotor trajectory tracking scheme. Aerosp Sci Technol. 2022;127:107725.

25. Ben Abdi S, Debilou A, Guettal L, Guergazi A. Robust trajectory tracking control of a quadrotor under external disturbances and dynamic parameter uncertainties using a hybrid p-pid controller tuned with ant colony optimization. Aerosp Sci Technol. 2025;160:110053.

26. Wu J, Wang H, Li N, Yao P, Huang Y, Su Z, et al. Distributed trajectory optimization for multiple solar-powered uavs target tracking in urban environment by adaptive grasshopper optimization algorithm. Aerosp Sci Technol. 2017;70:497-510.

27. Trad TY, Choutri K, Lagha M, Meshoul S, Khenfri F, Fareh R, et al. Real-Time implementation of quadrotor uav control system based on a deep reinforcement learning approach. Comput Mater Contin. 2024;81:4757-86.

28. Caffyn Yuste P, Iglesias Martínez JA, Sanchis De Miguel MA. Simulation-Based evaluation of model-free reinforcement learning algorithms for quadcopter attitude control and trajectory tracking. Neurocomputing. 2024;608:128362.

29. Gün A. Attitude control of a quadrotor using pid controller based on differential evolution algorithm. Expert Syst Appl. 2023;229:120518.

30. Luo Q, Li Y, Zeng J, Wu G, Wang Y. Quadrotor navigation considering attitude: a deep reinforcement learning method using tangent path rewards. Expert Syst Appl. 2026;298:129762.

31. Wang J, Zhang Y, Ren S, Wang H. Parameter optimization design of mpc controller in auv motion control based on improved black-winged kite algorithm. J Mar Sci Eng. 2025;13:2018.

32. Zhao J, Liu S, Xue W, Ding Y, Li D. Disturbance-Rejection guaranteed mpc design for mimo systems with application to a 2-dof helicopter. Aerosp Sci Technol. 2025;168:110757.

33. Jiang J, Tang J, Liu M, Li Y, Chen H, Cao D. A Data-Driven neural model predictive controller for multi-layer nonlinear vibration isolation system. Aerosp Sci Technol. 2025;166:110583.

34. Roberge V, Tarbouchi M, Labonte G. Fast genetic algorithm path planner for fixed-wing military uav using gpu. IEEE Trans Aerosp Electron Syst. 2018;54:2105-17.

35. Shao S, He C, Zhao Y, Wu X. Efficient trajectory planning for uavs using hierarchical optimization. IEEE Access. 2021;9:60668-81.

36. Jiao X, Ren J, Hua Y, Li Q. Online Nonparametric Identification Modeling of Ship Maneuvering Motion Based on PSO-Optimized Incremental Gaussian Mixture Model. Eng Appl Artif Intell. 2025;160:111962.

© 2023 The Author(s). | Published by Agro Intelligence Research Press

37. Lu Q, Zeng H, Yin L. Dynamic distributed multi-objective mantis search algorithm based on transformer hybrid strategy for novel power system dispatch. Energy. 2025;332:136075.

38. Raffo GV, Ortega MG, Rubio FR. An integral predictive/nonlinear h∞ control structure for a quadrotor helicopter. Automatica. 2010;46:29-39.

39. Zheng E-H, Xiong J-J, Luo J-L. Second Order Sliding Mode Control for a Quadrotor UAV. ISA Trans. 2014;53:1350-6.

© 2023 The Author(s). | Published by Agro Intelligence Research Press