A Dual–Layer Cascaded PID with Multi-Mixing Allocation Control with Direction Preservation for UAV Offboard Control Using ROS2 and PX4

Jacopo Dallafior
University of Trento
jacopo.dallafior@studenti.unitn.it

***Abstract*—This work presents the theory and implementation of a fully external multirotor control architecture for PX4 operating in Offboard mode. The controller follows a two–layer cascaded structure: a slower outer loop closes the position and velocity dynamics in the inertial frame, generating acceleration and attitude commands, while a high–rate inner loop tracks the desired attitude and produces body–frame torque and collective thrust requests. Motor commands are obtained through a custom multi–motor algorithm (MMA) scheme that maps the desired torque–thrust wrench into normalized actuator inputs. The allocator combines the closed form inversion with structured desaturation and slew–rate limiting with a constrained quadratic program that enforces actuator bounds and preserves the direction of the commanded roll–pitch torque, which is particularly important under strong saturation. The overall system is exercised using a three–dimensional trajectory generator with yaw–gated progression, ensuring heading coherence while probing both nominal and aggressive flight regimes in order to systematically excite saturation behaviour.**

***Index Terms*—UAV control, cascaded PID, Offboard PX4, mixer allocation, direction preservation**

I. Introduction

Cascaded PID architectures remain the de facto standard for multirotor control [1], offering clear structure and intuitive tuning. In conventional PX4 deployments[2], these layers (position, velocity, attitude, rate) are executed internally on the flight controller.

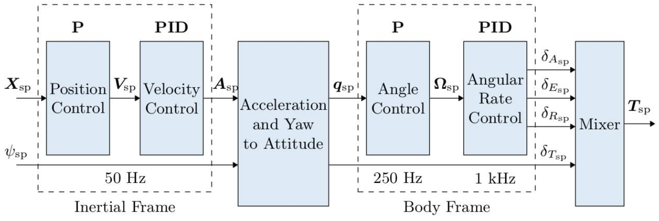

P
PID
P
PID
$\boldsymbol{X}_{\mathrm{sp}}$
Position Control
$\boldsymbol{V}_{\mathrm{sp}}$
Velocity Control
$\boldsymbol{A}_{\mathrm{sp}}$
Acceleration and Yaw to Attitude
$\boldsymbol{q}_{\mathrm{sp}}$
Angle Control
$\boldsymbol{\Omega}_{\mathrm{sp}}$
Angular Rate Control
$\delta_{A_{\mathrm{sp}}}$
$\delta_{E_{\mathrm{sp}}}$
$\delta_{R_{\mathrm{sp}}}$
$\delta_{T_{\mathrm{sp}}}$
Mixer
$\boldsymbol{T}_{\mathrm{sp}}$
ψ_{sp}
50 Hz
250 Hz
1 kHz
Inertial Frame
Body Frame

Fig. 1: Classic PID control structure proposed and used internally by PX4[2]

In contrast, the present work externally implements the full cascade, computing acceleration, attitude, torque, and actuator commands offboard and injecting them into PX4 through ROS 2 with direct–actuator Offboard modes. This bypasses PX4's internal loops and exposes the full control bandwidth to the experimenter, enabling custom allocation logic and explicit handling of saturation geometry.

The main contributions of this work are:
(i) A rigorous formulation of a cascaded PID controller in which the outer loop produces desired translational accelerations, and the inner loop tracks attitude through body–frame torque commands;
(ii) an acceleration–to–attitude mapping consistent with multirotor geometry, incorporating safety tilt limits and a normalized thrust law that matches hover equilibrium and captures lateral–tilt coupling;
(iii) a geometry–aware MMA scheme combining closed form solution with matrix inversion, enforced actuator bounds;
(iv) a constrained quadratic program that preserves roll–pitch torque direction under saturation and ensure the required wrench requested to be fully allocated;
(iv) a trajectory–generation and tracking pipeline, used to exercise both nominal and aggressive flight regimes to systematically evaluate saturation behaviour and allocation quality.

Frames, Signals, and Offboard Interface

The controller follows the same reference–frame conventions used by PX4[2]. All translational quantities (position, velocity, and the acceleration command produced by the outer loop) are expressed in the inertial North–East–Down (NED) world frame:

$$\boldsymbol{p} = [x, y, z]^\top, \qquad \boldsymbol{v} = [v_x, v_y, v_z]^\top, \qquad \boldsymbol{a}_r = [a_x, a_y, a_z]^\top.$$

Body–frame quantities use the Forward–Right–Down (FRD) convention. Angular rates are read from PX4 odometry as

$$\boldsymbol{\omega} = [p, q, r]^\top,$$

and attitude is provided as a quaternion $q = [w, x, y, z]$; ZYX yaw–pitch–roll angles are extracted only when computing attitude errors in the inner loop. Gravity is $g = 9.81\,\mathrm{m/s^2}$ along the positive NED z–axis.

The node runs entirely offboard and replaces PX4's internal cascades. At high rate it publishes:

- an *OffboardControlMode* message enabling the direct-actuator channel (and disabling position, velocity, and attitude setpoints inside PX4);
- *command setpoints* consisting of normalized motor commands via `/fmu/in/actuator_motors`, and additionally thrust and torque setpoints for debugging;
- *telemetry subscriptions* to local position/velocity (`vehicle_local_position`) and full odometry (`vehicle_odometry`) providing quaternion attitude and body rates.

All publishers and subscribers use ROS 2 QoS settings consistent with PX4 MAVLink–ROS bridges (Best Effort, Volatile durability, Keep Last with depth 1). Setpoint topics are streamed continuously and without timestamp gaps to prevent Offboard failsafe triggers and to maintain deterministic timing in the cascaded PID and QP allocation loop.

II. Control Formulation

Figure 2 illustrates the complete control architecture implemented offboard.

A. Outer–Loop PID: Position and Velocity Regulation

Given a reference trajectory $\boldsymbol{p}_d(t)$, $\boldsymbol{v}_d(t)$, and $\boldsymbol{a}_d(t)$ expressed in NED, the outer loop operates with a frequency of 50 Hz, is structures as a PID in acceleration space and following the idea proposed in [3] we plug explicit feedforward terms:

$$\boldsymbol{a}_r = \boldsymbol{a}_d + K_p(\boldsymbol{p}_d - \boldsymbol{p}) + K_d(\boldsymbol{v}_d - \boldsymbol{v}) + K_i\,\boldsymbol{I}_p, \tag{1}$$

where $\boldsymbol{I}_p$ is an integral state updated with conditional integration. The feedforward term $\boldsymbol{a}_d$ removes the nominal phase lag of a pure feedback loop and makes the feedback part responsible only for disturbances and modeling errors.

B. Acceleration→Attitude/Thrust Mapping

The translational controller outputs $\boldsymbol{a}_r$ (desired acceleration in NED). Following the work done in [4] the true multirotor translational dynamics can be expressed as:

$$\boldsymbol{a} = g\,\boldsymbol{e}_3 - \boldsymbol{f},$$

where $\boldsymbol{f}$ is the mass–normalized thrust vector expressed in NED, we form the desired specific force

$$\boldsymbol{f} = \begin{bmatrix} -a_{r,x} \\ -a_{r,y} \\ g - a_{r,z} \end{bmatrix}. \tag{2}$$

Let ψ_d be the commanded yaw. Under a ZYX decomposition, the desired tilt that aligns the body frame z axis with $\boldsymbol{f}$ is obtained in closed form as

$$\theta_d = \arctan 2\big(f_x \cos\psi_d + f_y \sin\psi_d,\ f_z\big), \tag{3}$$

$$\phi_d = \arcsin\big(-f_y\cos\psi_d + f_x \sin\psi_d\big), \tag{4}$$

If we want to sent directly the thrust command to PX4 we need to find the mass-normalized thrust required to realize (2) is

$$\frac{T}{m} = \|\boldsymbol{f}\|.$$

hence the normalized command is

$$u_{\text{th}} = u_{\text{hov}}\big(\tfrac{\|\boldsymbol{f}\|}{g}\big), \tag{5}$$

where u_{hov} is the normalized collective corresponding to stationary hover.
Otherwise the command sent to the MMA is:

$$T = \|\boldsymbol{f}\| m.$$

The resulting commands $(\phi_d, \theta_d, \psi_d)$ serve as the references for the inner attitude loop.

C. Inner-Loop PID: Attitude to Torque Regulation

The inner loop runs at higher rate (300 Hz) and regulates attitude through a PID law in Euler error [1], augmented by body-rate damping:

$$\boldsymbol{\tau} = K^p_\eta\,\boldsymbol{e}_\eta + K^i_\eta\,\boldsymbol{I}_\eta - K^d_\omega\,\boldsymbol{\omega}_{\text{lpf}}, \tag{6}$$

where $\boldsymbol{e}_\eta = [e_\phi, e_\theta, e_\psi]^\top$ is the wrapped ZYX Euler error and $\boldsymbol{I}_\eta$ is the attitude integral state.
For a ZYX parametrization, the relation between Euler–angle rates and body rates is

$$\dot{\boldsymbol{\eta}} = T(\boldsymbol{\eta})\,\boldsymbol{\omega}, \qquad T(\boldsymbol{\eta}) = \begin{pmatrix} 1 & \sin\phi\,\tan\theta & \cos\phi\,\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{pmatrix}.$$

Is important to note that for our is still possible to use the identity matrix for the mapping between Euler angles rate and body rates:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{\omega},$$

This allows me to implement the derivative action directly on the Euler-rate error,

$$\dot{\boldsymbol{e}}_\eta = \dot{\boldsymbol{\eta}}_{\text{des}} - \dot{\boldsymbol{\eta}}_{\text{meas}},$$

so that the final PID law becomes

$$\boldsymbol{\tau} = K^p_\eta\,\boldsymbol{e}_\eta + K^i_\eta\,\boldsymbol{I}_\eta + K^d_\eta\,\dot{\boldsymbol{e}}_\eta.$$

In the implementation we consider quasi-static attitude references at the inner-loop level and set

$$\dot{\eta}_{\text{des}} = 0,$$

so that the derivative action reduces to a pure damping term on the reconstructed Euler rates. Each rate channel uses a first order low pass:

$$\alpha = e^{-2\pi f_c \Delta t}, \qquad \omega_{\text{lpf}} \leftarrow \alpha\,\omega_{\text{lpf}} + (1-\alpha)\,\omega,$$

The measured rate is obtained from the filtered body rates via

$$\dot{\eta}_{\text{meas}} = \omega_{\text{lpf}},$$

and the PID law becomes

$$\tau = K_{p,\eta} e_\eta + K_{i,\eta} I_\eta - K_{d,\eta}\dot{\eta}_{\text{meas}}.$$

The controller gains are kept fixed across all experiments; the values are reported in Table I.

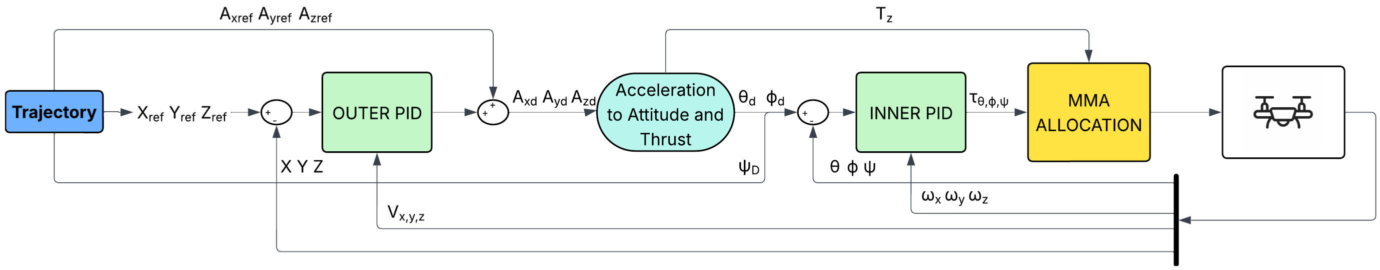


Fig. 2: Overview of the control structure

TABLE I: PID gains used for the outer position loop and inner attitude loop.

outer position loop	K_p	K_i	K_d
x	0.50	0.01	0.80
y	0.50	0.01	0.80
z	0.65	0.02	0.70
inner attitude loop	K_p	K_i	K_d
roll	0.25	0.12	0.03
pitch	0.25	0.12	0.03
yaw	0.20	0.08	0.04

D. Anti–Windup: Conditional Integration and Back–Calculation

The controller uses complementary anti–windup mechanisms:

(i) Conditional integration. An integrator is advanced only when its paired actuator is not saturated in the direction of its incremental action. For example, the integral is updated only when the actiator command is strictly inside its bounds or when the error is small (near–setpoint logic), preventing runaway integral during saturation.

(ii) Back–calculation anti–windup behavior. When actuator saturation is detected before integration, the proportional–derivative attitude torque estimate $\boldsymbol{\tau}_p$ is examined. If this estimate would exceed the actuator limits, the attitude integrator $\boldsymbol{I}_\eta$ is driven toward zero using a short tracking time, rapidly removing any accumulated integral bias that cannot be produced by the motors.

Let $u^\star$ be the motor command provided by the allocator and let $w_{\text{alloc}} = B\,u^\star$ be the resulting realized wrench. The allocation residual is defined as

$$r \triangleq w_{\text{des}} - w_{\text{alloc}} = \begin{bmatrix} r_{M_x} \\ r_{M_y} \\ r_{M_z} \\ r_{F_z} \end{bmatrix}. \tag{7}$$

Only the moment components $r_{1:3}$ are used for attitude anti–windup. With $D_{K_i} = \text{diag}(K^i_{\eta,\phi},\, K^i_{\eta,\theta},\, K^i_{\eta,\psi})$ and $K_{aw} = \text{diag}(k_{aw,\phi},\, k_{aw,\theta},\, k_{aw,\psi})$, the integrator update becomes

$$I^+_\eta = I_\eta + e_\eta\,\Delta t - K_{aw}\,\frac{r_{1:3}}{D_{K_i}}\,\Delta t.$$

The anti–windup gains follow the standard *tracking–time* heuristic widely used in practical PID implementations. For each axis i, the gain is chosen as

$$k_{aw,i} = \frac{1}{T_{t,i}}, \qquad T_{t,i} = \sqrt{T_{i,i}\,T_{d,i}},$$

where $T_{i,i} = K_{p,i}/K_{i,i}$ and $T_{d,i} = K_{d,i}/K_{p,i}$ are the integral and derivative time constants. This rule is consistent with practical guidelines for PID anti–windup design reported in Viola *et al.* [5]. These mechanisms prevent integral runaway during torque/thrust saturation and ensure smooth recovery when the allocator returns to the feasible region.

E. Limit Handling and Safety Mechanisms

The controller incorporates several layers of limits that reflect actuator feasibility and preserve robustness across aggressive maneuvers. The outer loop clamps horizontal acceleration commands componentwise,

$$a_{r,x} \leftarrow \text{sat}(a_{r,x}, -a_{xy,\max}, a_{xy,\max}),$$

$$a_{r,y} \leftarrow \text{sat}(a_{r,y}, -a_{xy,\max}, a_{xy,\max}),$$

as implemented in software. Componentwise bounds are computationally inexpensive, avoid geometric coupling, and prevent the allocator from being driven into infeasible lateral-tilt requests. Vertical acceleration is handled separately through the thrust law, providing a clean separation between lateral tilt and vertical lift authority. To avoid excessive attitude demands, the desired roll and pitch are individually limited,

$$|\phi_d| \le \phi_{\max}, \qquad |\theta_d| \le \theta_{\max},$$

which matches the per–axis clamping used in the code. Although these bounds define a rectangular region in (ϕ, θ) rather than a spherical tilt cone, the approach is simple, robust, and effective for the envelope explored in our tests.

III. Motor–Mixing Allocation (MMA): Wrench Tracking, Constraints, Priorities, and Anti–Windup Coupling

The multirotor has n normalized actuators $\boldsymbol{u} \in [0, 1]^n$, each producing thrust proportional to $k_f u_i$, with torque contributions determined by motor locations and spin directions. The goal of the allocation layer is to convert the desired body–frame wrench

$$\boldsymbol{w}_{\text{des}} = \begin{bmatrix} M_x^{\text{des}} \\ M_y^{\text{des}} \\ M_z^{\text{des}} \\ F_z^{\text{des}} \end{bmatrix}, \qquad F_z < 0 \text{ for up–thrust},$$

into feasible actuator commands while respecting motor bounds, slew constraints, and for this project the idea is to prioritize roll–pitch authority under saturation.

This section formalizes the allocation logic implemented in software and details the geometric motivation for each term.

Let the i–th motor be located at (x_i, y_i) in FRD coordinates. With thrust coefficient k_f and yaw torque ratio k_m, the contribution of motor i to the body wrench is

$$M_x^{(i)} = (-y_i)k_f u_i, \qquad M_y^{(i)} = (+x_i)k_f u_i,$$

$$M_z^{(i)} = s_i(k_m k_f)u_i, \qquad F_z^{(i)} = -k_f u_i,$$

where $s_i \in \{+1, -1\}$ denotes the spin direction (CW/CCW). Stacking all motors yields

$$\boldsymbol{w} = B\,\boldsymbol{u}, \qquad B \in \mathbb{R}^{4\times 4},$$

with columns of B built from $(x_i, y_i, k_f, k_m, s_i)$. For a quadrotor in “X” geometry with arm coordinates (xi , yi) the matrix B is:

$$\begin{bmatrix} -y_1 k_f & -y_2 k_f & -y_3 k_f & -y_3 k_f \\ x_1 k_f & x_2 k_f & x_3 k_f & x_4 k_f \\ s_1 k_m k_f & s_2 k_m k_f & s_3 k_m k_f & s_4 k_m k_f \\ -k_f & -k_f & -k_f & -k_f \end{bmatrix}.$$

This model exactly matches the implementation: roll is generated by differential thrust along y, pitch along x, yaw by reaction torques, and $F_z < 0$ indicates upward thrust in the FRD convention.

The mixer maintains a nominal hover command

$$\boldsymbol{u}_0 = u_{\text{hov}}\mathbf{1}, \quad B\boldsymbol{u}_0 \approx [0\;0\;0\;-mg]^\top,$$

where u_{hov} is tuned experimentally. The effective commanded wrench relative to hover is

$$\delta\boldsymbol{w} = \boldsymbol{w}_{\text{des}} - B\boldsymbol{u}_0,$$

and the incremental actuator required to realize it is $\delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_0$, satisfying

$$\delta\boldsymbol{w} = B\,\delta\boldsymbol{u}.$$

Although the QP acts on absolute $\boldsymbol{u}$, this incremental view explains the structure of the regularization terms and the bias toward symmetric solutions around hover.

A. Closed–form allocation

When no actuator limits or priorities are active, the desired wrench increment $\delta\boldsymbol{w} \in \mathbb{R}^4$ is mapped to a motor increment $\delta\boldsymbol{u} \in \mathbb{R}^4$ through the inverse of the allocation matrix $B \in \mathbb{R}^{4\times 4}$. For the classic quadrotor mixer used here, B is nonsingular, so the mapping is uniquely defined as

$$\delta\boldsymbol{u} = B^{-1}\,\delta\boldsymbol{w}.$$

The corresponding absolute command is obtained by adding the hover anchor $\boldsymbol{u}_0$, namely

$$\boldsymbol{u} = \boldsymbol{u}_0 + \delta\boldsymbol{u} = \boldsymbol{u}_0 + B^{-1}\,\delta\boldsymbol{w}.$$

This closed–form relation exactly realizes any commanded wrench increment $\delta\boldsymbol{w}$ as long as the resulting $\boldsymbol{u}$ remains inside the admissible actuator domain. In practice, however, the inverse mapping by itself does not enforce input bounds, rate limits, or the desired yaw–versus–roll priority. To handle these constraints and priorities explicitly, the implementation replaces the pure inverse B^{-1} with a constrained quadratic program, in line with the prioritized allocation strategy proposed in [6].

B. Quadratic Programming Formulation

In order to directly deal with limits saturation and directionality preservation it was decided to use QP with a custom cost function. The four wrench axes differ in both physical authority and operational priority. To enforce uniform conditioning and clear relative importance, define normalization

$$S = \text{diag}\Big(\tfrac{1}{M_{x,\max}}, \tfrac{1}{M_{y,\max}}, \tfrac{1}{M_{z,\max}}, \tfrac{1}{F_{z,\max}}\Big)$$

and a diagonal priority weight

$$W = \text{diag}(w_x, w_y, w_z, w_f), \qquad w_x = w_y \gg w_z,$$

where yaw is the first axis to relax under saturation. The composite scaling

$$C = WS$$

appears throughout the QP cost and matches exactly the normalization used in code.

Directionality in the roll–pitch plane

Define the roll–pitch block $B_{rp} \in \mathbb{R}^{2\times n}$ such that

$$\boldsymbol{t}_{rp}(\boldsymbol{u}) = \begin{bmatrix} M_x \\ M_y \end{bmatrix} = B_{rp}\boldsymbol{u}. \tag{8}$$

Given a desired in–plane torque $\boldsymbol{t}_{rp}^{\text{des}} = \begin{bmatrix} M_x^{\text{des}} & M_y^{\text{des}} \end{bmatrix}^\top$, its unit direction is

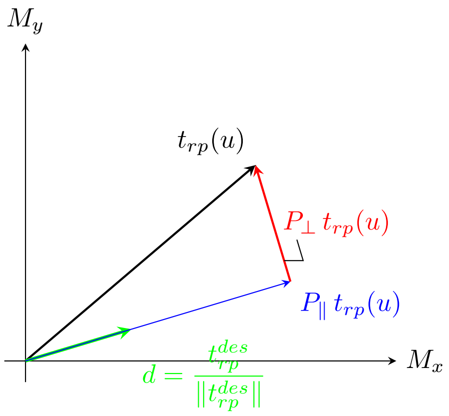
$$\boldsymbol{d} = \frac{\boldsymbol{t}_{rp}^{\text{des}}}{\|\boldsymbol{t}_{rp}^{\text{des}}\|}, \quad \text{used when } \|\boldsymbol{t}_{rp}^{\text{des}}\| > \varepsilon. \tag{9}$$

Introduce the parallel and perpendicular projectors

$$P_{\parallel} = \boldsymbol{d}\,\boldsymbol{d}^\top, \qquad P_{\perp} = I_2 - \boldsymbol{d}\,\boldsymbol{d}^\top. \tag{10}$$

Directionality preservation is the geometric requirement $P_{\perp}\,\boldsymbol{t}_{rp}(\boldsymbol{u}) = \mathbf{0}$, which enforces collinearity with $\boldsymbol{d}$ while

allowing the magnitude to scale. The misalignment measure is
$\|P_{\perp}\boldsymbol{t}_{rp}(\boldsymbol{u})\|_2^2 = \|\boldsymbol{t}_{rp}(\boldsymbol{u})\|_2^2 - (\boldsymbol{d}^{\top}\boldsymbol{t}_{rp}(\boldsymbol{u}))^2$.



The in–plane torque decomposes into a parallel component along the desired direction and an orthogonal one. The projector penalty reduces the orthogonal component, thereby driving the angle θ to zero while allowing the magnitude along $\boldsymbol{d}$ to scale according to feasibility and the axis priorities.

C. Constrained weighted QP with hover and smoothness regularization

At each control cycle, the allocator solves the strictly convex QP

$$\begin{aligned}\min_{\boldsymbol{u}}\ \ & J(\boldsymbol{u}) = \tfrac{1}{2}\|C(B\boldsymbol{u} - \boldsymbol{w}_{\text{des}})\|^2 + \tfrac{\lambda_{\text{dir}}}{2}\|P_{\perp}B_{rp}\boldsymbol{u}\|^2 \\ & \qquad + \tfrac{\rho_0}{2}\|\boldsymbol{u} - \boldsymbol{u}_0\|^2 + \tfrac{\rho_v}{2}\|\boldsymbol{u} - \boldsymbol{u}_{\text{prev}}\|^2 \\ \text{s.t.}\ \ & \boldsymbol{u}_{\min} \le \boldsymbol{u} \le \boldsymbol{u}_{\max}, \\ & -\Delta \le \boldsymbol{u} - \boldsymbol{u}_{\text{prev}} \le \Delta.\end{aligned} \tag{11}$$

- The first term tracks the desired wrench with axis normalization and priority weighting.
- The directionality term penalizes deviation from the desired roll–pitch direction under saturation.
- The ρ_0 term anchors the solution near hover.
- The ρ_v term discourages rapid changes, yielding smooth motor trajectories and improving QP stability.
- Box constraints enforce motor authority; the slew box implements communication and actuator dynamics limits ($\Delta = \dot{u}_{\max}\Delta t$).

QP Structure and Cost Construction

A standard convex quadratic program in actuator commands has the canonical form

$$\begin{aligned}\min_{\boldsymbol{u}}\ & \tfrac{1}{2}\boldsymbol{u}^{\top}H\boldsymbol{u} + g^{\top}\boldsymbol{u} \\ \text{s.t.}\ \ & Ax \le b, \\ & Ex = d\end{aligned} \tag{12}$$

where $H \succeq 0$ is the Hessian, g is the linear term, and the constraints encode actuator bounds and slew-rate limits. This structure is directly compatible with operator-splitting solvers OSQP [7], which require only H, g and the stacked box constraints.

The implemented allocator instantiates (12) with a Hessian composed of the three cost contributions, the wrench tracking terms, the direction preservation term and the regularization and smoothness term. Collecting all components, the final Hessian and gradient are

$$H = (CB)^{\top}(CB) + \lambda_{\text{dir}}\, B_{rp}^{\top}P_{\perp}B_{rp} + (\rho_0 + \rho_v)I_n,$$

$$g = -(CB)^{\top}C\boldsymbol{w}_{\text{des}} - \rho_0\boldsymbol{u}_0 - \rho_v\boldsymbol{u}_{\text{prev}}.$$

Because $P_{\perp} \succeq 0$ and $(\rho_0 + \rho_v)I_n \succ 0$, the matrix H is symmetric positive definite for all feasible weights. Hence the QP remains strictly convex irrespective of the operating point or saturation level. The solver (OSQP) exploits the fixed sparsity pattern of H and A, and is warm-started with $(\boldsymbol{u}_{\text{prev}}, \boldsymbol{y}_{\text{prev}})$ every cycle, making the solve time negligible at the controller's update rate.

D. Summary and relation to classical mixing

Classical mixers implement $\boldsymbol{u} = B^{\dagger}\boldsymbol{w}_{\text{des}}$ followed by clipping. The approach presented here replaces post–hoc clipping with a single convex optimization that respects actuator limits and rate constraints by construction, encodes axis priorities through C, preserves the physical axis of tilt through the directionality projector. This unifies mixing and prioritization in one mathematically consistent stage that remains computationally light.

IV. Trajectory Generation

A Lissajous/figure–8 reference is produced as

$$x_r(t) = x_c + A_x \sin(\omega t), \tag{13}$$
$$y_r(t) = y_c + A_y \sin(2\omega t + \phi), \tag{14}$$
$$z_r(t) = z_c, \tag{15}$$

with tangential yaw $\psi_d(t) = \operatorname{atan2}(\dot{y}_r, \dot{x}_r)$. In addition to the Lissajous trajectory, a second reference is introduced in order to deliberately excite saturation and expose allocator behaviour under sustained high load. A constant–radius circle is defined as

$$x_r(t) = x_c + R\cos(\omega t), \qquad y_r(t) = y_c + R\sin(\omega t), \tag{16}$$

where R is the commanded radius and ω sets the traversal speed. This trajectory generates a nearly constant lateral specific forcehen approaches the available lateral thrust margin, the actuators enter prolonged saturation intervals. This makes the circular path an effective stress test for the allocation layer, revealing how the mixer redistributes thrust, prioritizes roll–pitch authority, and relaxes yaw torque when the commanded wrench becomes infeasible.

V. MAIN RESULTS

The controller is evaluated on two reference trajectories generated as described in Section IV: a nominal figure–eight and an aggressive constant–radius circle. The first trajectory exercises the cascaded PID in a regime where the required lateral acceleration and collective thrust stay close to the envelope but do not lead to sustained saturation. The second trajectory is explicitly designed to push the vehicle into prolonged saturation and to expose the behaviour of the allocation layer under high load.

All results are computed from recorded flight data after resampling reference and measured signals on a common time grid and discarding an initial transient of duration T_{skip}. The horizontal position error at sample t_k is

$$e_{xy}(t_k) = \sqrt{\big(x(t_k) - x_r(t_k)\big)^2 + \big(y(t_k) - y_r(t_k)\big)^2}, \quad (17)$$

and the corresponding RMS value over N samples is

$$\text{RMS}_{XY} = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1} e_{xy}^2(t_k)}. \quad (18)$$

Vertical and full 3D position RMS values are obtained in the same way from $z(t_k) - z_r(t_k)$ and from the full position error vector.

Attitude tracking is evaluated for roll, pitch and yaw separately. The wrapped errors are

$$e_\phi(t_k) = \phi(t_k) - \phi_r(t_k), \quad e_\theta(t_k) = \theta(t_k) - \theta_r(t_k),$$
$$e_\psi(t_k) = \text{wrap}_{(-\pi,\pi]}\big(\psi(t_k) - \psi_r(t_k)\big),$$

and each channel is summarized by its RMS value.

To assess allocation quality, the directional cosine between desired and realized wrench in x and y is used,

$$c(t) = \cos\theta(t) = \frac{w_{\text{des}}(t)^\top w(t)}{\|w_{\text{des}}(t)\|\|w(t)\|}, \quad (19)$$

which equals one when the applied wrench is perfectly aligned with the commanded wrench. Actuator saturation intervals are detected whenever at least one motor command reaches its lower or upper bound; these intervals are later used to condition the statistics of $c(t)$.

A. *Figure–eight tracking*

The first set of experiments evaluates the controller on a figure–eight trajectory in a nominal regime, where the required lateral acceleration and collective thrust remain close to the available envelope but only rarely drive the actuators into hard saturation. The horizontal ground track in Fig. 3 shows that both the closed form inversion mixer and the QP allocator follow the reference path closely, with only a small amplitude loss near the lobes where the combined lateral acceleration and thrust demand is highest. The residual bias is consistent with the imposed bounds and with the directionality term in the QP cost, which prefers to scale the torque magnitude along the desired in–plane direction instead of rotating it. The corresponding altitude evolution in Fig. 4 confirms that the vertical motion remains essentially flat.

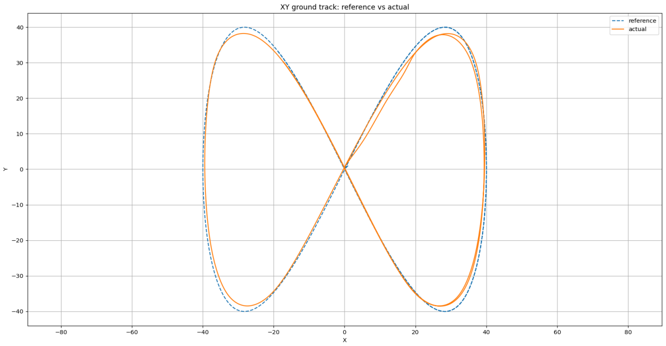


Fig. 3: XY ground track for the figure–eight trajectory. The vehicle follows the reference with only a small amplitude loss near the lobes, where lateral acceleration and thrust demand are highest. The residual bias is consistent with the imposed bounds and with the directionality term, which scales the torque magnitude along the desired in–plane axis instead of rotating it.

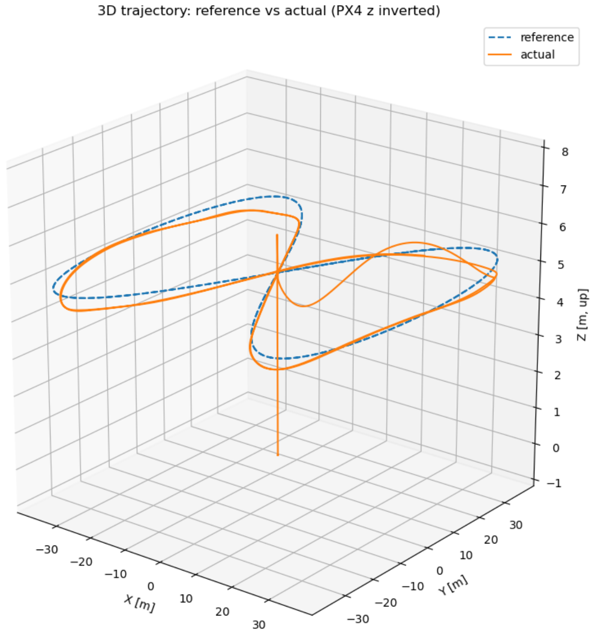
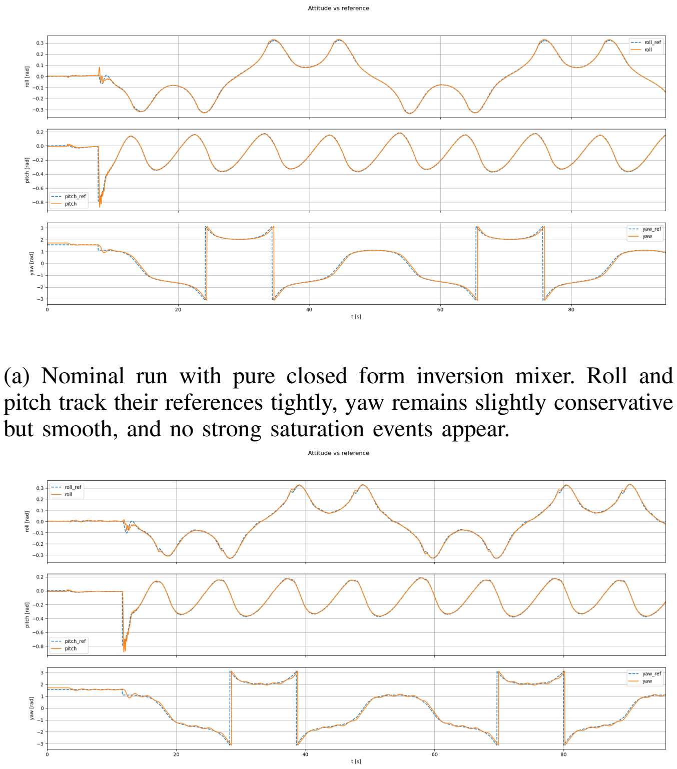


Fig. 4: Altitude profile during figure–eight tracking. The vehicle maintains the commanded height with only mild deviations around the lobes, consistent with thrust limits and the directionality cost in the allocator.

The attitude trajectories in Fig. 5 further illustrate that roll and pitch track their references tightly in both cases. With the closed form inversion mixer, the yaw response is slightly conservative but smooth, whereas the QP allocator produces a yaw trace with a bit more high–frequency content, reflecting the lower weight assigned to yaw in the QP cost.

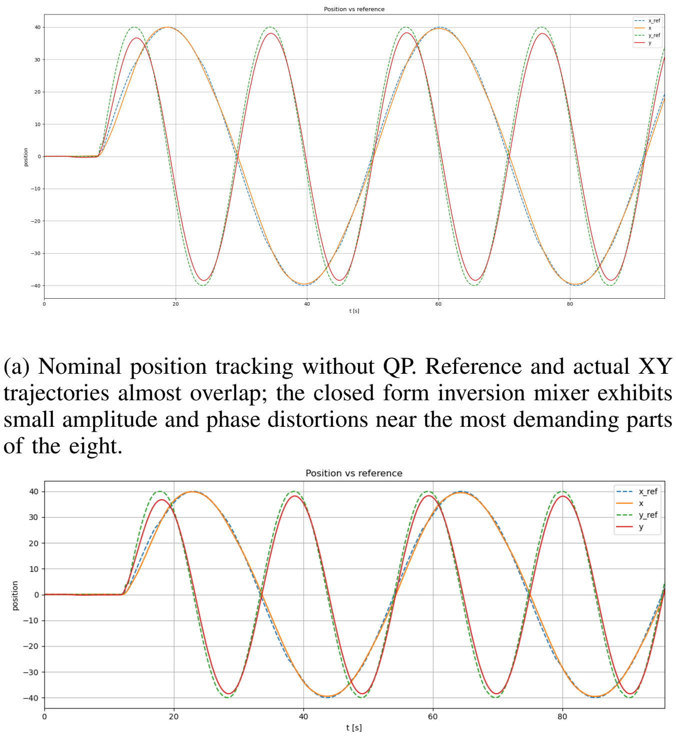
Because the ground tracks in Fig. 3 and the position traces



(a) Nominal run with pure closed form inversion mixer. Roll and pitch track their references tightly, yaw remains slightly conservative but smooth, and no strong saturation events appear.

(b) Nominal run with QP allocator. Attitude trajectories stay within limits and show slightly more high–frequency motion in yaw, reflecting the lower yaw weight in the QP cost.

Fig. 5: Comparison of nominal attitude tracking for the closed form inversion mixer and the QP allocator during figure–eight tracking.



(a) Nominal position tracking without QP. Reference and actual XY trajectories almost overlap; the closed form inversion mixer exhibits small amplitude and phase distortions near the most demanding parts of the eight.

(b) Nominal position tracking with QP. The XY path remains visually indistinguishable from the closed form inversion mixer case, while the RMS analysis reveals a modest reduction in horizontal error.

Fig. 6: Comparison of nominal position tracking for the closed form inversion mixer and the QP allocator during figure–eight tracking.

in Fig. 6 are visually very similar, the performance difference between LS and QP is better captured by the RMS metrics.

The first two rows of Table II report the figure–eight RMS values. The horizontal position RMS decreases from $2.77\,\mathrm{m}$ with the closed form inversion mixer to $2.58\,\mathrm{m}$ with the QP allocator. Attitude RMS errors remain well below $0.03\,\mathrm{rad}$ for roll and pitch and in the 7° range for yaw in both cases. Overall, on the nominal trajectory the constrained QP formulation delivers a modest but systematic improvement in horizontal tracking without degrading attitude performance.

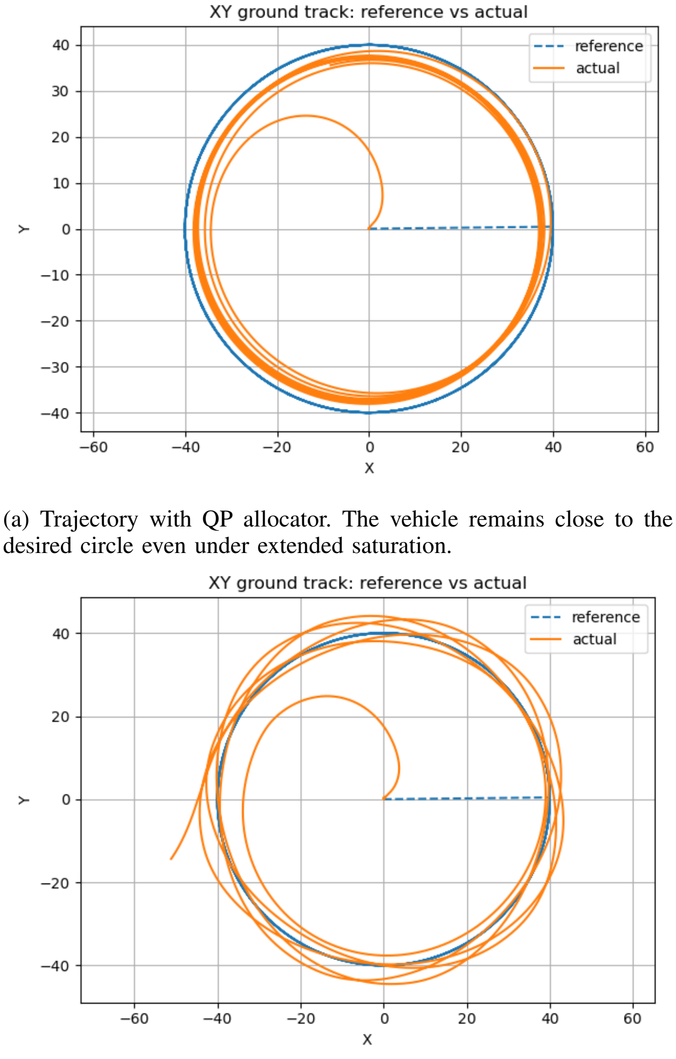
Directional–cosine statistics for the figure–eight are summarized by the first two rows of Tables III and IV. Globally, both allocators maintain excellent alignment between desired and realized wrench, with mean cosine above 0.998 and more than 99.5% of the time spent with $c(t) \geq 0.99$. The QP allocator roughly halves the RMS misalignment angle with respect to closed form inversion (from 3.27° to 1.72°), indicating a slightly cleaner use of the available actuator authority even in the lightly saturated regime. Saturation events are rare on this trajectory: the closed form inversion mixer spends only 0.4% of the time in saturation and the QP allocator about 4.5%. In both cases the mean cosine during saturation remains close to one and the corresponding RMS misalignment angles stay below 10°, confirming that direction preservation is not critical for this nominal case.

B. Aggressive circle maneuvers and direction preservation

To highlight the advantages of the QP allocator under strong saturation, a second experiment considers an aggressive constant–radius circle, flown at a ground speed of about $80\,\mathrm{km/h}$. In this regime the lateral specific force required to close the circle pushes the actuators into sustained saturation, so that the allocator must explicitly decide how to trade roll–pitch torque, yaw torque and collective thrust while remaining inside the feasibility region.

The trajectories in Fig. 7 show that, with the QP allocator, the vehicle remains close to the desired circular path and the radius and centre of the trajectory are preserved even during long saturation intervals. With the closed form inversion mixer, instead, the lack of explicit prioritization causes roll and pitch authority to be compromised when the motors hit their limits: the circle deforms, its centre drifts, and the overall tracking quality degrades visibly.

The directional–cosine plots in Figs. 8 and 9 make this difference more explicit. In the closed form inversion case the light blue bands highlight intervals where at least one



(a) Trajectory with QP allocator. The vehicle remains close to the desired circle even under extended saturation.

(b) Trajectory wit closed form inversion mixer. The circle deforms and drifts when the actuators saturate, reflecting the lack of explicit direction preservation.

Fig. 7: Comparison of aggressive circular trajectories with and without the QP allocator.

motor is saturated. Whenever the system enters these bands the cosine drops sharply below one, indicating that the realized roll–pitch torque rotates away from the commanded direction even though its magnitude is clamped by saturation. In contrast, with the QP allocator the cosine remains close to one throughout most saturated intervals, demonstrating that the QP formulation preserves the torque direction in the (M_x, M_y) plane and only scales its magnitude according to feasibility, as intended by the projector term in the cost.

Attitude time histories in Figs. 10 and 11 provide a complementary view. With the closed form inversion mixer, roll and pitch struggle to maintain the nearly constant tilt required by the circular motion whenever the actuators saturate, and yaw exhibits large excursions that couple back into the lateral dynamics. When the same test is repeated with the

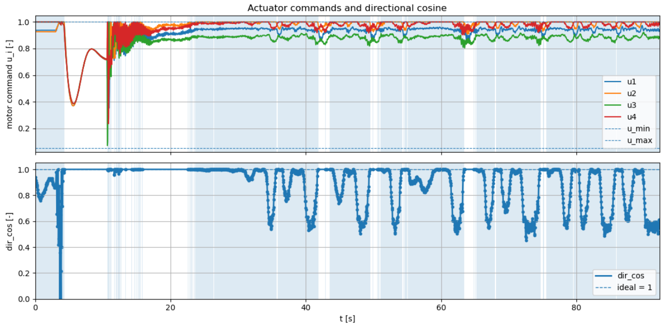


Fig. 8: Aggressive circle with closed form inversion mixer. Actuator commands and directional cosine metric. In the light blue saturation intervals the directional cosine departs significantly from one, indicating loss of alignment between the desired and allocated roll–pitch torque.

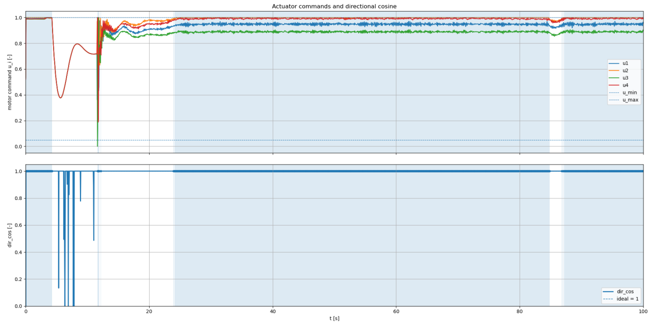


Fig. 9: Aggressive circle with QP allocator. Actuator commands and directional cosine metric. Even under sustained saturation, the directional cosine remains close to one, showing that the allocator preserves the torque direction in the roll–pitch plane while reducing its magnitude.

QP allocator, roll and pitch remain much closer to their references during the high–load portions of the maneuver, while yaw is explicitly relaxed according to its lower weight in the cost. This behaviour is consistent with the goal of the direction–preserving QP formulation, which sacrifices yaw torque in favour of preserving the roll–pitch torque direction needed to sustain the circle.

The last two rows of Table II summarize the RMS errors for the aggressive circle. The QP allocator slightly reduces horizontal and vertical position RMS with respect to closed form inversion and, more importantly, significantly improves attitude RMS. In particular, yaw RMS decreases from about 9.7° with the closed form inversion mixer to roughly 7.5° with the QP allocator, indicating that the combination of prioritised allocation and anti–windup logic prevents large attitude excursions even under prolonged saturation.

Directional–cosine metrics for the circle are reported in the last two rows of Tables III and IV. Globally, the closed form inversion mixer achieves a mean cosine of 0.89 with a large standard deviation and an RMS misalignment angle

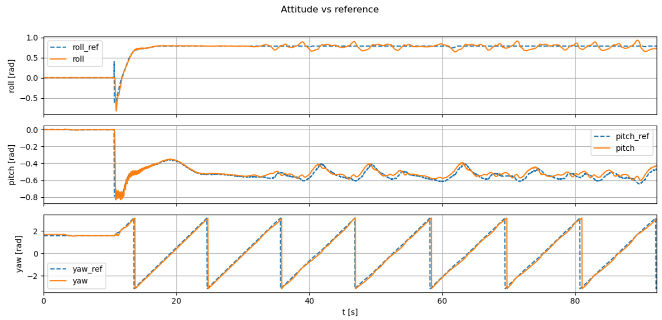


Fig. 10: Aggressive circle with closed form inversion mixer. The attitude loop struggles to maintain the desired roll and pitch during saturation, and yaw excursions become large, degrading trajectory tracking.

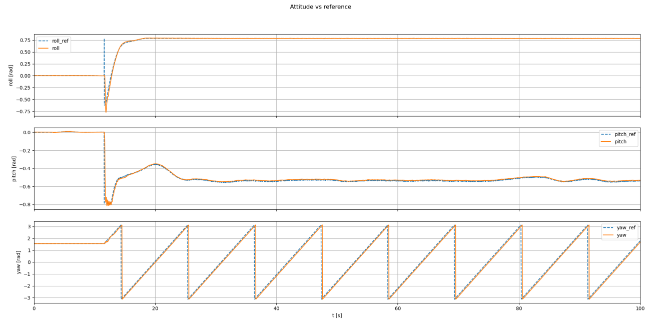


Fig. 11: Aggressive circle with QP allocator. Roll and pitch remain closer to the desired nearly constant tilt despite saturation, while yaw is relaxed according to its lower priority in the cost.

of about 27°, whereas the QP allocator keeps the mean cosine near 0.996 and the RMS angle below 6°. The contrast becomes even sharper when conditioning on saturation. The QP allocator operates in saturation for about 80% of the experiment but maintains a mean cosine essentially equal to one and an RMS misalignment angle of only 0.12° in those intervals; the few larger deviations are concentrated outside saturation when the vehicle enters and leaves the high–load regime. The closed form inversion mixer, on the other hand, saturates for about 78% of the time and exhibits an RMS misalignment angle of roughly 31° during those intervals, confirming that the QP allocator makes a much more effective use of the limited actuator authority.

C. Quantitative summary

Tables II–IV collect the main quantitative results, with rows grouped by trajectory (figure–eight first, then aggressive circle). The nominal figure–eight confirms that, away from heavy saturation, the closed form inversion (CFI) mixer and QP allocator deliver very similar behaviour and that the additional structure introduced by the QP primarily refines horizontal positioning and wrench alignment. The aggressive circle highlights the main benefit of the proposed allocation strategy: under strong and sustained saturation the direction–preserving QP with yaw de–prioritization and smoothness regularization yields substantially better tracking and a much more controlled misalignment between desired and realized wrench.

TABLE II: Position and attitude RMS tracking errors for figure–eight and aggressive circle trajectories. Attitude RMS is given in degrees. Rows are grouped by trajectory: first the figure–eight, then the aggressive circle.

scenario	RMS XY [m]	RMS Z [m]	RMS roll [deg]	RMS pitch [deg]	RMS yaw [deg]
QP figure–eight	2.580	1.023	0.48	1.56	6.15
CFI figure–eight	2.768	1.030	0.38	1.60	6.90
QP aggressive circle	12.839	2.676	1.61	1.57	7.46
CFI aggressive circle	14.254	3.918	5.52	5.68	9.70

TABLE III: Directional cosine statistics over the whole actuator interval. Time fractions are expressed as percentages of the total experiment duration. Rows are grouped by trajectory.

scenario	mean	std	time $c \geq 0.99$ [%]	RMS angle [deg]
QP figure–eight	0.9996	0.0187	100.0	1.723
CFI figure–eight	0.9986	0.0329	99.6	3.268
QP aggressive circle	0.9961	0.0619	99.4	5.778
CFI aggressive circle	0.8942	0.1590	51.4	27.205

TABLE IV: Directional cosine statistics in and out of actuator saturation. “In saturation” refers to time instants where at least one motor command is at its bound. Rows are grouped by trajectory.

scenario	time in saturation [%]	mean c in sat	mean c out of sat	RMS angle in sat [deg]
QP figure–eight	4.5	0.9917	1.0000	0.142
CFI figure–eight	0.4	0.9859	0.9987	9.641
QP aggressive circle	79.8	1.0000	0.9805	0.118
CFI aggressive circle	77.3	0.8388	0.9965	30.946

VI. Conclusion

This work presented a fully external dual–layer cascaded PID architecture for multirotor Offboard control on ROS 2 and PX4, combining an acceleration–centric outer loop, an attitude–torque inner loop, and a geometry–aware multi–motor allocator. The allocator unifies classical mixing, actuator limits, desaturation priorities, and slew constraints in a single quadratic program that approximately preserves roll–pitch torque direction while relaxing yaw and thrust when saturation is unavoidable. Experiments on a nominal figure–eight and an aggressive high–speed circle show that, away from heavy saturation, the QP allocator matches the tracking performance of a least–squares mixer while slightly improving horizontal accuracy, and that under strong saturation it yields substantially better trajectory tracking and wrench alignment according to position RMS and directional–cosine metrics. Future work will extend the framework to more complex multirotor geometries and investigate automatic tuning and learning–based adaptation of the allocation weights.

References

[1] T. Mien, T. Tu, and V.-A. Vo, “Cascade pid control for altitude and angular position stabilization of 6-dof uav quadcopter,” *International Journal of Robotics and Control Systems*, vol. 4, pp. 814–831, 05 2024.

[2] "Px4 autopilot user guide," "PX4 Guide (main)", Online, 2025, available: https://docs.px4.io/main/en/ [Accessed: 25 Nov. 2025].

[3] Z. Bingul and K. Gul, "Intelligent-pid with pd feedforward trajectory tracking control of an autonomous underwater vehicle," *Machines*, vol. 11, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2075-1702/11/2/300

[4] D. Kooijman, A. P. Schoellig, and D. J. Antunes, "Trajectory tracking for quadrotors with attitude control on $\mathcal{S}^2 \times \mathcal{S}^1$," 2019. [Online]. Available: https://arxiv.org/abs/1906.06926

[5] J. Viola, J. Rueda, J. Ortega, and C. Flórez, "Practical pid controller implementation for the speed control of a motor generator system," in *IOP Conference Series: Materials Science and Engineering*, vol. 437, no. 1. IOP Publishing, 2018, p. 012013, creative Commons Attribution 3.0 License.

[6] E. Smeur, D. Höppener, and C. D. Wagter, "Prioritized control allocation for quadrotors subject to saturation," in *International Micro Air Vehicle Conference and Flight Competition 2017*, H. de Plinval, J.-M. Moschetta, and G. Hattenberger, Eds. Toulouse, France: 9th International Micro Air Vehicles Conference and Flight Competition, September 2017, pp. 37–43. [Online]. Available: http://www.imav2017.org/

[7] B. Stellato, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: The operator splitting quadratic program solver — documentation," https://osqp.org/docs/solver/index.html, accessed: 2025-02-20.