Available online at https://asric.africa/engineering-sciences
ASRIC Journal on Engineering Sciences Vol.6(1)(2025) 135-146

Evaluation of Guidance, Navigation, and Control Algorithms for Hydrogen-Powered Multi-Aircraft Systems

Mamokete Makhubo, Bright Ndebele, Phumudzo Ragimana,

Council for Scientific and Industrial Research, 1 Meiring Naude, Pretoria
kmakhubo@csir.co.za, bndebele@csir.co.za, pragimana@csir.co.za

Corresponding Author: kmakhubo@csir.co.za

Received 10 July 2025; revised 11 August 2025; accepted 23 October 2025

Abstract

This paper presents a modular digital-twin framework to compare inner-loop attitude controllers: geometric PD ("PID"), discrete Linear-Quadratic Regulator (LQR), and a move-penalized linear Model Predictive Controller (MPC), for a heavy-lift T30-class quadcopter intended for hydrogen propulsion study. The twin couples 6-DoF rigid-body dynamics, actuator mixing, motor/ESC lag, a bus-level electrical model, and stochastic wind (Ornstein–Uhlenbeck) with look-ahead guidance on a sharp lawnmower survey. To isolate controller effects, task difficulty is equalized by autotuning a single scalar so that the achieved cross-track Root-Mean-Squared Error (RMSE) lies in a 2.6 ± 0.25 m band. The tuned controllers then run identical 600 s simulations under the same wind seed and retimed speeds. On the equalized run, PID and LQR achieve 2.45 m and 2.55 m RMSE, respectively, while MPC settles at 3.13 m due to its move penalty and finite horizon. All three deliver survey-class performance with mean bus power ≈ 4.8 kW and peaks in the 7–8 kW range, but MPC reduces energy per meter by approximately 2.3% at the cost of relaxed lateral accuracy. A 10-seed Monte Carlo confirms this trade: PID/LQR remain in-band for 90%/80% of seeds, while MPC consistently lowers energy per meter with similar mean power but gentler peaks. For hydrogen-electric UAVs, these metrics map directly to propulsion co-design, where energy per meter informs hydrogen mass and range, peak power sets stack/buffer sizing, and actuator smoothness affects balance-of-plant transients. The results show that controller selection is not only a matter of tracking accuracy but also an energy-management lever: PID/LQR suit survey tolerance, while MPC-style penalization favours endurance and balance-of-plant stability.

Keywords: Digital twin; Guidance, Navigation, and Control; Heavy-lift Multirotor; Hydrogen-electric propulsion; Predictive Control Model.

1. Introduction

Aviation is undergoing a fast shift toward low-carbon footprint propulsion. Among the considered technologies to combat high carbon emissions in aviation, hydrogen-electric powertrains offer a promising zero-emission in-flight CO_2 and high cycle efficiency when tightly integrated with the airframe and controls. Early demonstrations of Hydrogen propulsion exploration in aviation date back to the 1950s with NASA's Project BEE (Fenn et al., 1960), which briefly operated a B-57 on liquid Hydrogen. Since then, the research has emphasized hardware integration of the Hydrogen propulsion systems, like the digital co-design of onboard storage and its airframe impact (Reimer et al., 2024).

By contrast, Guidance, Navigation, and Control (GNC) for hydrogen-powered Unmanned Aerial Vehicles (UAVs) has received less research attention, despite being central to energy use and mission yield. Digital twins (DT) address this gap in research by coupling high-fidelity models with embedded controllers and realistic operational environments. DTs enable Software-in-the-Loop (SIL) and

Hardware-in-the-Loop (HIL) evaluations on aircraft (David et al., 2023) without the dangers of flight-testing propulsion systems to verify them. Recent DT frameworks for UAVs combine dynamics, Lyapunov-stable control, and (N)MPC with toolchains such as CasADi/ACADOS, and show improvements with HILs in closed sensing/actuation loops (Guevara et al., 2025).

Classic UAV GNC stacks integrate low-cost IMUs, GPS/vision, and sensor fusion with inner-loop attitude control (PID, LQR) and increasingly, Model Predictive Modelling (MPS) (Kim et al., 2006). For Hydrogen-electric UAVs, the controller is also an energy manager; tracking aggressiveness, turn dynamics, and actuator usages directly shape bus-level power, peaks, and energy per meter travelled. These metrics are the key inputs to Hydrogen fuel-cell sizing and balance-of-plant (BoP) sizing.

This paper develops a modular digital twin workflow and uses it to compare PID, LQR, and move-penalized MPC on a heavy-lift quadcopter representative of a T30-class UAV. A survey mission (take-off, lawnmower survey, approach landing) was run in steady wind with an electrical model consisting of a 14S electrical bus model with ESC overhead and iron losses. To ensure a fair comparison, controllers were equalized by tuning a single scalar per method to a common cross-track root-mean-square (RMS) tolerance, then tracking, energy, peak power demands, and actuator response on an identical mission timeline. The resulting mean and peak power aligned with published values for the class, supporting external validity.

This study addresses two central research questions. First, which guidance and inner-loop strategy can minimize bus-level energy consumption and peak power across mission phases while still meeting survey-level tolerances? Second, how do PID, LQR, and move-penalized MPC differ in terms of tracking accuracy, peak power, and actuator commands required for stable flight?

2. Methodology

System overview

The DT system developed in this study integrates the core subsystems required to emulate the behaviour of a heavy-lift quadcopter under realistic flight conditions. The architecture couples a rigid-body dynamics engine, actuator and motor models, stochastic wind disturbances, and a power estimation module within a unified Python-based simulation framework. The rigid-body dynamics are implemented in a standalone engine (quadcopter_dynamics.py), which is loaded by the simulator at runtime. This modular design allows each component to be independently tuned, replaced, or validated while maintaining a consistent interface for data flow throughout the system. The modularity of the implementation allows the same framework to be extended to other types of airframes, like the fixed-wing and hybrid-VTOL configurations, without altering the simulation loop.

The simulator is fully developed in Python, using widely used scientific libraries for numerical integration and analysis. NumPy and SciPy provide the mathematical foundation for linear algebra and differential equation solvers, while matplotlib is used for visualization of trajectories, power traces, and mission performance. The framework also includes structured configuration files via Python's data-classes module, allowing reproducible setup of mission parameters, vehicle properties, wind/environmental conditions, and controller gains.

Fig. illustrates the software architecture of the DT, highlighting the modular organisation of the dynamics, controllers, tuning, and analysis components. Guidance algorithms generate velocity or acceleration targets that are translated into thrust and attitude setpoints by the outer loop of a controller. Inner-loop controllers (PID, LQR, or MPC) regulate body torques to achieve these setpoints, commanding motor speeds through an allocation matrix. The actuator dynamics propagate motor responses, and the resulting thrusts and torques are fed into the Newton-Euler dynamics engine to update the vehicle state. Moreover, the power estimation model computes electrical power demand from motor operating points, while disturbance models inject stochastic wind forces into the translational dynamics. The outputs, including trajectory, tracking errors, power demand, and tilt statistics, are logged and processed into performance metrics of the DT.

This integrated architecture provides a flexible platform for evaluating different GNC strategies and forms the foundation for the dynamic modelling of the UAV described in this section.

```
digital_twin_uav/
├── init.py
├── models/
│   ├── init.py
│   ├── quadcopter_dynamics.py   # Newton-Euler 6DoF dynamics engine (LOADED onto quadcopter model)
│   ├── fixed_wing_dynamics.py
│   └── vtol_dynamics.py
├── controllers/
│   ├── init.py
│   ├── pid.py                   # SO(3) geometric PD
│   ├── lqr.py                   # discrete Riccati (DARE) + builder
│   └── mpc.py                   # move-penalized horizon QP (+ warm-start)
├── tuning/
│   ├── init.py
│   └── autotuner.py             # one-scalar equalizer:
│                                #   - bracket init (PID: [0.6,1.6], LQR/MPC: [0.4,8.0])
│                                #   - expand/shrink if both misses same side
│                                #   - geometric bisection, settle-into-band rule, prefer-from-above option
│                                # Exposes: tune_to_xte(controller, target, eps, max_evals, ...)
├── propulsion/
│   ├── init.py
│   └── power_demand_export.py   # exports load profiles from mission logs (for fuel-cell HIL)
├── sensors/
│   ├── init.py
│   └── imu_gps_barometer.py     # (optional) sensor models
├── simulator/
│   ├── init.py
│   ├── mission_profile.py       # sharp lawnmower WPs + duration retimer (η)
│   └── simulation_loop.py       # main integrator: wind → guidance → attitude/thrust → inner-loop → Ω → forces → NE6DoF → power → logs
├── energy_analysis/
│   ├── init.py
│   └── power_metrics.py         # Wh, Wh/m, means/peaks, tilt, saturation, RPM-rate
├── experiments/
│   ├── init.py
│   ├── equalized_experiment.py  # calls tuning.autotuner.tune_to_xte(), then runs PID/LQR/MPC with tuned scales
│   ├── mc_experiment.py         # seeds loop; optional per-seed retune
│   └── pareto_sweep.py          # grid of XTE targets → runs + Pareto CSV/PDF
└── main.py                      # CLI entrypoint to orchestrate equalize / MC / Pareto; plotting flags
```

Fig. 1: Software architecture of the digital twin framework. The modular package structure displays how the dynamics, controllers, tuning algorithms, propulsion, and analysis components are organized and loaded onto the simulation loop. Other aspects include sensors, which will not be expanded in this investigation. The framework is customizable to testing, modelling, and simulation requirements.

Vehicle Description

The reference vehicle loaded onto the simulation is a heavy-lift quadcopter with a maximum take-off weight (MTOW) of approximately 35 kg. The quadcopter is configured in an X-layout, with four fixed-pitch propellers providing thrust and attitude control through differential speed modulation.

The vehicle geometry is typical for this weight class: a rotor-to-rotor span exceeding 2 m and propeller diameters near 0.965 m (38 inches), enabling high thrust margins suitable for payload carrying and

endurance missions. The airframe mass distribution is simplified using a diagonal inertia tensor [Equation 6], an assumption that is consistent with symmetric multirotor designs where the cross-coupling inertial terms are negligible (Bos et al., 2021).

Actuation is achieved exclusively through the four electric rotors. Roll, pitch, and yaw moments are generated by differential thrust allocation, while collective thrust provides altitude and acceleration control. This configuration makes the quadcopter dynamically over-actuated in thrust but underactuated in lateral translation, as is characteristic of rotorcraft.

Energy supply is modelled as a 14S high-voltage bus, consistent with commercial UAV battery packs in this weight class. In the present study, this serves as a proxy for hydrogen-electric integration, allowing the same bus-level dynamics to be preserved while decoupling propulsion modelling from energy storage.

This specific UAV class was selected because it reflects on relevant operational use cases in South Africa: precision agriculture and aerial survey missions where heavy payloads, long endurance, and energy efficiency are critical design aspects. It therefore provides a suitable platform for evaluating GNC trade-offs that directly impact propulsion co-design.

UAV Dynamic Modelling: Newton-Euler Method

The vehicle dynamics are modelled as a rigid body with six degrees of freedom (6-DoF) using the Newton-Euler formulation to propagate both translational and rotational motion. This formulation is well-suited for rotorcraft because it explicitly captures forces acting on a body and torques generated by individual actuators. The method accommodates external disturbances (such as wind) and allows for direct mapping of motor thrust and torque into the rigid-body equations of motion (Massa & Vignolo, 2016). In contrast to energy-based Lagrangian approaches, Newton-Euler equations provide a computationally efficient structure for time-domain simulation and are widely adopted for UAV flight dynamics studies (Dargham et al., 2015).

Kinematics

The kinematic equations define how the position and orientation of the UAV change over time:

$$\dot{p} = v \tag{1}$$

$$\dot{q} = \frac{1}{2} q \otimes [0 \;\; \omega] \tag{2}$$

Where $p \in \mathbb{R}^3$ is the vehicle's position in the inertial world frame {W}, $v \in \mathbb{R}^3$ is the linear velocity, $q \in \mathbb{H}$ (an element of the quaternion algebra $\mathbb{H}$) is a unit quaternion describing the orientation of the body frame {B} relative to {W}, and $\boldsymbol{\omega} \in \mathbb{R}^3$. The quaternion satisfied the normalization constraint $\|q\| = 1$.

Translational dynamics

Forces acting on the UAV include thrust, aerodynamic drag, and gravity. The translational dynamics are governed by:

$$m\dot{\boldsymbol{v}} = R(q)\begin{bmatrix}0\\0\\T\end{bmatrix} + \boldsymbol{F}_{drag} + m\begin{bmatrix}0\\0\\-g\end{bmatrix} \tag{3}$$

Where m is the mass of the vehicle, T is the total thrust generated by the rotors, $R(q) \in SO(3)$ is the rotation matrix corresponding to quaternion q and g is the gravitational acceleration at the value of $g = 9.81\ m/s^2$. Aerodynamic drag is modelled using linear approximation in the axis of the velocity vector:

$$\boldsymbol{F}_{drag} = -c_{drag}(\boldsymbol{v} - \boldsymbol{w}) \tag{4}$$

With the coefficient of drag (c_{drag}) and relative velocity ($(\boldsymbol{v} - \boldsymbol{w})$, where $\boldsymbol{w}$ is the wind disturbance vector.

This formulation captures the dominant contributions to rotorcraft translational dynamics, while neglecting higher-order aerodynamic effects such as induced drag, blade flapping, and ground effect. These simplifications are widely adopted in UAV modelling literature to balance fidelity with tractability (Bos et al., 2021).

Rotational Dynamics

The rotational dynamics of a rigid body are governed by the Newton-Euler moment balance:

$$\boldsymbol{J\dot{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times (\boldsymbol{J\omega}) \tag{5}$$

Where $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\tau} \in \mathbb{R}^3$ is the net control torque from the actuators. For the quadcopter considered in this paper, the geometry is symmetric about the body axes, rendering the cross products of inertia negligible. Due to this, the inertia tensor is reduced to a diagonal matrix $\boldsymbol{J} = \boldsymbol{I} = diag(I_{xx}, I_{yy}, I_{zz})$ resulting in the rotational dynamics being governed by:

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}\big(\boldsymbol{\tau} - \boldsymbol{\omega} \times (\boldsymbol{I\omega})\big) \tag{6}$$

The state is thus propagated forward at each simulation timestep $dt = 0.0006\,s$ using a semi-explicit Euler scheme, where translational accelerations, angular accelerations, and quaternion derivatives are first computed. After this, velocity, position, and quaternions are updated accordingly. Quaternions are normalized after each step to prevent drift. This approach balances numerical efficiency with stability for long-duration simulations (Ibrahim et al., 2023).

Actuator and Controller

Controllers

The DT supports multiple inner loop attitude controllers. In this study, we evaluate three:

1. Geometric PID on SO(3): implemented as a proportional-derivative (PD) controller with integral augmentation for robustness. It is labelled "PID" for continuity with prior UAV literature. The controller computes commanded body rates and thrust by minimizing attitude and cross-track error.
2. Discrete-Time Linear Quadratic Regulator (LQR): designed on the linearized stat-space model around hover. Gains are computed by solving the discrete algebraic equation.
3. Move-Penalized Linear Model Predictive Control (MPC): formulated on the same linearized model as the LQR, with an additional penalty on successive input changes to smooth actuator commands. MPC solves a constrained optimization problem at each timestep, enabling improved power efficiency and actuator friendliness at the cost of computational overhead.

The simulator runs a discrete-time inner attitude loop at 167 Hz and an outer guidance loop at 56Hz. This mirrors common practice where on-board attitude control is fast, while off-board/mission guidance operates near 50Hz (MAVLink/PX4 offboard rates). Each of the controllers operates on the UAV's attitude and angular rates to produce body torques and a collective thrust.

Geometric PID on SO(3)

The PID regulates the UAV's orientation directly on the rotation group SO(3). The attitude error e_R compares the actual orientation $R \in SO(3)$ with the desired orientation R_d. The angular velocity error e_ω compares the measured body rates $\boldsymbol{\omega} \in \mathbb{R}^3$ with desired rates ($\boldsymbol{\omega_d}$):

$$e_R = \frac{1}{2}\big(R_d^T R - R^T R_d\big)^V \quad , \quad e_\omega = \omega - R^T R_d \omega_d \tag{7}$$

The commanded torques are then computed as:

$$\tau = -K_R e_R - K_\omega e_\omega - K_I \int e_R dt \tag{8}$$

While the commanded thrust is:

$$T = \big(mg + K_z(z_d - z)\big)e_3^T R e_3 \tag{9}$$

Where $e_3 = [0, 0, 1]^T$, a unit vector along the z-axis of the world frame (ENU convention).

- K_R, K_ω, K_I, K_z are the proportional, derivative, integral, and vertical gains.
- $z_d \; and \; z$ are the desired and actual altitude, respectively.

This structure preserves the geometric nature of orientation while retaining an interpretable PID form.

LQR

The UAV is linearized around hover with the state vector:

$$x = [\phi, \theta, \psi, p, q, r]^T$$

Where ϕ, θ, ψ are roll, pitch, yaw angles, and p, q, r are body rates. The input vector is:

$$u = \left[\tau_\phi, \tau_\theta, \tau_\psi\right]^T$$

representing control torques. The linearized model is given by:

$$x_{k+1} = A\boldsymbol{x_k} + B\boldsymbol{u}_k \tag{10}$$

The LQR minimizes a quadratic cost over states and inputs. The resulting law is therefore:

$$u_k = -K\boldsymbol{x_k} \tag{11}$$

With a feedback gain

$$K = (R + B^T PB)^{-1} B^T PA + Q \tag{12}$$

P is the solution to the discrete Riccati equation:

$$P = A^T PA - A^T PB(R + B^T PB)^{-1} B^T PA + Q$$

$Q \in \mathbb{R}^{n \times n}$ in this instance is the state weighting matrix that penalizes deviations in the states. Larger diagonal entries enforce tighter regulation of corresponding state values. $R \in \mathbb{R}^{m \times m}$ is an input weighting matrix that penalizes large control efforts. Higher values discourage aggressive actuator commands and smooth the response. This produces optimal state regulation under quadratic penalties.

MPC

MPC uses the same linearized model as LQR but optimizes over a finite horizon N. The objective penalizes states, inputs, and, importantly, input rate changes to promote actuator smoothness:

$$\min \left(\sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k + \Delta u_k S \Delta u_k \right) + x_N P x_N \right) \tag{13}$$

Subject to Equation 10 where $u_{min} \leq u_k \leq u_{max}$ where:

- Δu_k is the rate of control input change.
- S is the move-penalty weight.
- P is the terminal weight matrix.

```
# --- MPC setup | ---
dt_outer = 0.018          # every 3 inner steps of dt=0.006
N = 18                    # horizon (from MPCGains.horizon)

# Build prediction & move-penalty structures
Sx, Su = build_prediction_mats(Ad, Bd, N)
D = build_diff_operator(N, m)   # m = 3 torques

# Quadratic cost (state/input/move) and receding horizon
Qbar = np.kron(np.eye(N), Q)
Rbar = np.kron(np.eye(N), R)
Sbar = np.kron(np.eye(N), S)
H = Su.T @ Qbar @ Su + Rbar + D.T @ Sbar @ D

# Solve QP each outer step; apply first move only
U_star = mpc_qp_solve(setup, x0, xref, u_prev, (umin, umax), mpc, U_warm=U_warm)
u_cmd = U_star[:m]
```

Fig. 2: MPC implementation within the simulation environment in Python**.**

Only the first input is applied, with the optimization repeated at every timestep (receding horizon). This formulation mirrors the Python implementation Fig. **2**, which solves a quadratic program at each outer-loop step ($\Delta t = 18$) with a horizon length of $N = 18$, including actuator bounds and move penalties tuned for smooth control.

Mission Definition and Simulation Setup

The mission profile is defined as a lawnmower survey, which is representative of aerial mapping and agricultural monitoring tasks. The survey area spans 220 × 140 m and is flown at a constant altitude of 50 m above ground level (AGL). A swath width of 25 m results in six parallel north–south tracks, with sharp 180° turns at the boundaries. The waypoint layout is specified without path smoothing, such that the vehicle is required to negotiate sharp corners rather than rounded fillets.

This choice deliberately stresses the tracking performance of the controllers, particularly during the turning segments where cross-track error accumulation is most likely. At the same time, the rectangular geometry captures a realistic survey workload while ensuring comparability across all tested controllers.

The mission duration is fixed at 600 s to enable fair comparisons of energy consumption across controllers. To achieve this, the nominal survey speed is automatically retimed such that the commanded forward velocity yields the target duration when combined with take-off, survey, and landing segments.

The retiming factor is defined as:

$$\eta = \frac{L}{V_{des} T_{target}} \tag{14}$$

Where L is the total mission path length, V_{des} is the nominal survey speed, and T_{target} is the desired duration. The applied velocity command is then scaled as:

$$V_{app} = \eta V_{des} \tag{15}$$

For the baseline survey dimensions and wind conditions, this yields an applied velocity of 3.57 m/s from a desired 5.94 m/s, ensuring that the UAV completes the mission in the allotted time despite disturbances and turns overhead. This retiming mechanism standardizes experiment duration while preserving controller-specific tracking and energy behaviours.

Environmental Disturbances

The external disturbance environment is modelled using a stochastic Ornstein–Uhlenbeck (OU) process to capture low-frequency wind variability. The mean wind vector is set to:

$$w_{mean} = [3.0, 1.2, 0.0] m/s$$

corresponding to a light breeze with a mild crosswind component. Temporal correlation is introduced through a gust time constant of $\tau = 7\ s$, ensuring that wind fluctuations evolve smoothly rather than instantaneously.

This model balances realism and tractability: the OU process has been widely adopted in UAV simulation studies due to its ability to reproduce the spectral characteristics of atmospheric turbulence while remaining computationally lightweight (Chevillard, 2017). By selecting a correlated stochastic process instead of white noise, we ensure that the simulated gusts act on the vehicle in a manner consistent with field reports of survey-class UAVs.

Equalization (Autotuning)

To ensure that controller comparisons reflect energy–tracking trade-offs rather than differences in tuning effort, each inner-loop controller is equalized using a one-scalar autotuning scheme. The goal of equalization is to achieve a cross-track RMS error target of $x_{te,RMS} \in [2.6 \pm 0.25]\ m$ on the lawnmower survey mission.

The autotuning process follows this methodology:

1. **Bracket initialization**: a pair of gain scaling candidates (s_L, s_H) s selected. For PID, (0.6,1.6) is used, while for LQR and MPC the bracket is wider, (0.4,8.0), to account for greater sensitivity.
2. **Expansion of the bracket-pair**: if both candidates lie on the same side of the target band (both under- or over-tracking), the bracket is expanded or shrunk until it is within bounds.
3. **Geometric bisection**: a midpoint candidate is chosen as $s_M = \sqrt{s_L s_H}$ is evaluated and used to replace the boundary that lies on the same side of the band.
4. **Selection:** Once a candidate lands within the tolerance band, it is selected as the equalized gain scalar. Optionally, consecutive in-band runs are required for robustness.

This autotuning ensures that all controllers operate at comparable task difficulty, so that observed differences in power consumption, tracking accuracy, and actuator aggressiveness are attributable to the control law itself rather than arbitrary gain choices.

The fidelity of the digital twin was confirmed through three checks. First, allocator sanity showed that hover thrust matched the analytic value $T = mg$, with rotor speeds within 1.5% of the theoretical hover RPM from thrust coefficients. Second, power sanity confirmed that the simulated hover bus power agreed within $\pm 5\%$ of reported values for DJI T30-class multirotors, validating actuator and ESC loss models. Finally, mission-time sanity ensured comparability across controllers: although the vehicle often completed the survey before 600 s, the simulator ran the full duration to standardize energy accounting. Together, these verifications established consistency in thrust, power, and timing behavior, underpinning the reliability of the reported results.

Outputs and Logging

Simulation outputs included both time histories and decimated logs. At runtime, state trajectories (position, velocity, attitude, body rates), control inputs (torques, thrust), power draw, and mission phase markers were recorded at 50 Hz. Post processing produced summary plots such as 3D paths, altitude versus time, bus power traces, and tilt distributions, along with CSV logs containing per-sample data. These outputs provided the quantitative basis for the results section, enabling consistent evaluation of tracking accuracy, power consumption, and actuator usage across all controllers.

3. Results and Discussion

Given the acceptance band of $\left|x_{te,RMS} - 2.6\right| \leq 0.25m$, PID and LQR achieved 2.45 m and 2.55 m, respectively (Table 1). MPC converged to 3.13 m, which is outside of the band by 0.28 m, given the move penalty and horizon used. The power demand, however, remained valid across all three.

Table 1: Summary of controllers that reached the target after tuning

Controller	Scale	Achieved XTE_RMS [m]	In-band?
PID	0.600000	2.4527	Yes
LQR	1.788854	2.5549	Yes
MPC	18.000000	3.1266	No

All three controllers delivered survey-class power. Mean bus power effectively fell within the exact boundaries, with the PID demanding 4.795 kW, LQR demanding 4.794 kW, and MPC demanding 4.804 kW (Table **2**). Peak power demand during the sharp turns remained under class limits, with PID demanding 7.05 kW, LQR 7.63 kW, and MPC 7.05kW. The LQR demanded approximately an additional 8.2% (Fig) more than the PID during the shortest turns of the mission. Energy per meter travelled results favoured MPC. MPC reduced energy per meter by 2.28% relative to the PID and LQR while maintaining mean and peak power demand.

Table 2*:* Simulated results for $\mathrm{x_{te,RMS}} \pm 0.25\mathrm{m}$ for PID, LQR, and MPC for a 600-second tune and simulation

Metric	PID	LQR	MPC
Scale name	$w_n scale$	q_{scale}	q_{scale}

Scale value	0.600000	1.788854	18.000000
Inband ($\pm 0.25\ m\ at\ 2.6\ m$)	True	True	False
Power-study [within power tolerance]	True	True	True
XTE_RMS [m]	2.453	2.555	3.127
XTE_p95 [m]	6.721	6.868	8.465
Wh·m⁻¹	0.471	0.471	0.461
Mean power [W]	4 794.6	4 794.1	4 804.0
Peak power [W]	7 052.2	7 630.5	7 052.2
Tilt95 [deg]	22.62	20.13	27.96
Tiltmax [deg]	41.07	40.26	36.29
Sat. frac	0.0	0.0	0.0
Sim time [s]	599.994	599.994	599.994

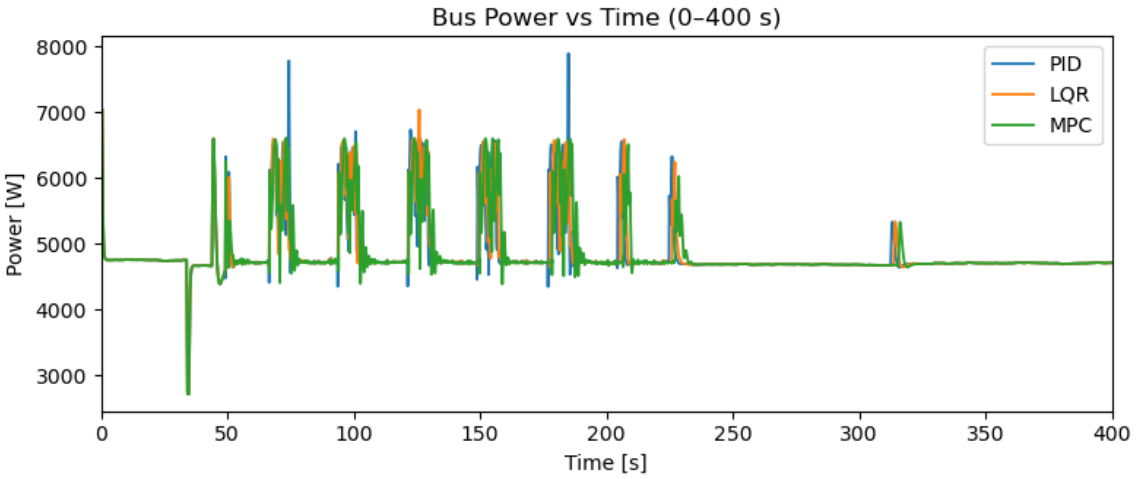

Fig. 3: Bus power vs time, three traces (PID/LQR/MPC)

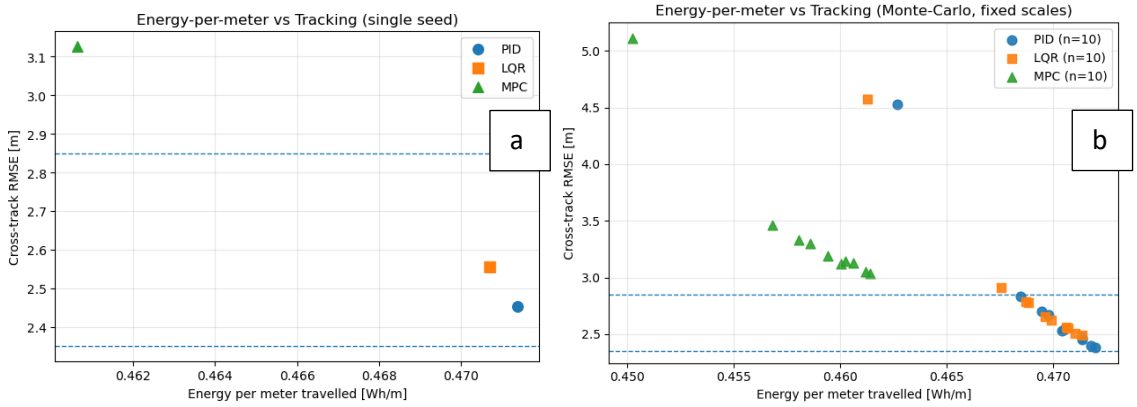

Fig. 41: Pareto tracking–energy trade-off. (a) Single-seed comparison of cross-track RMSE vs energy per meter travelled; one marker per controller (PID, LQR, MPC). Down/left is better. MPC is approximately 2.3% lower in energy per meter at looser tracking. (b) Monte-Carlo (10 seeds, fixed scales): point clouds by controller show the same trade-off across winds; the vertical band marks the equalization target 2.6 ± 0.252.

Under the shared wind realization and retimed speeds, LQR tracked slightly looser than PID (an additional 0.1 m), which was consistent with its gentler altitude response. MPC's looser tracking (0.67m away from PID results) coincides with its lower energy per meter travelled, illustrating the expected energy tracking trade. MPC trades off lateral precision to reduce energy per meter.

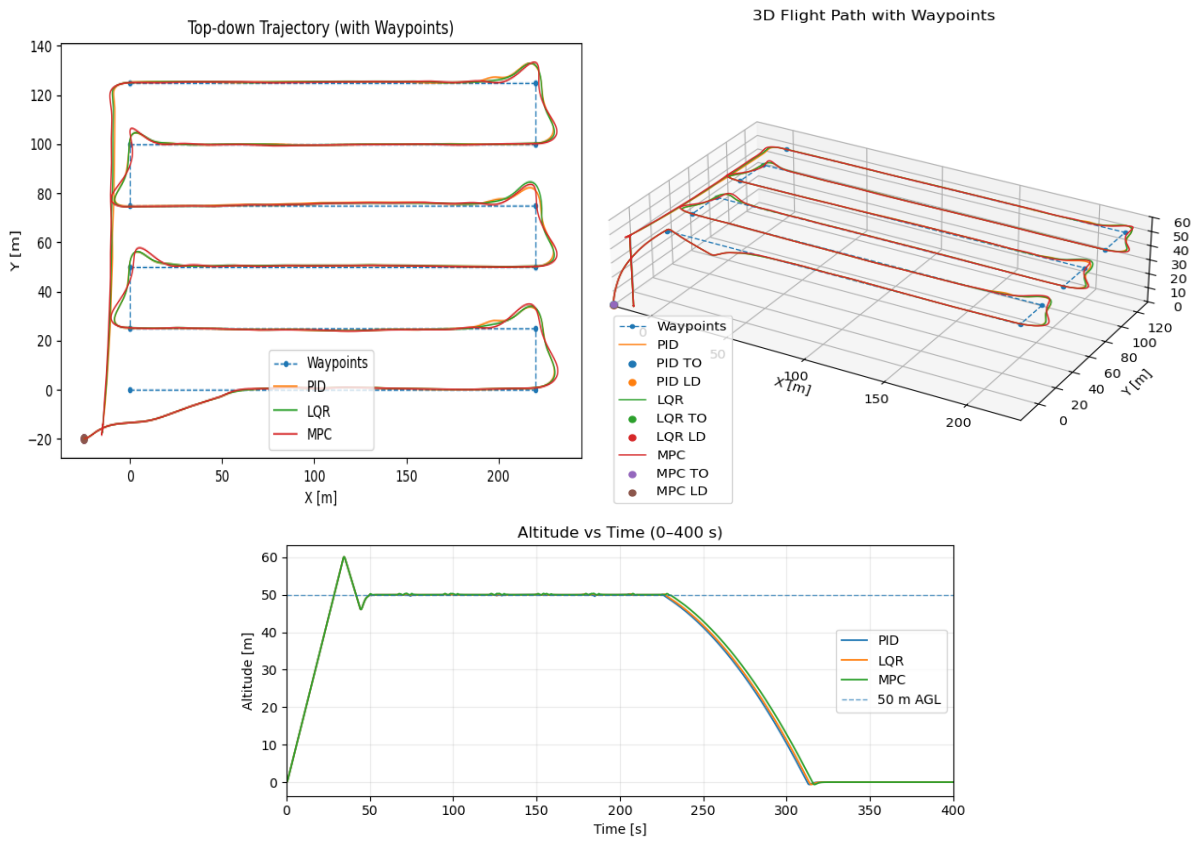

Fig. 5: Mission following: XY top-down (with waypoints), 3D trajectory, and altitude-vs-time.

3.1. Attitude and Saturation

Tilt results show distinct differences. LQR operated with the lowest 95th percentile ($\approx 20.1°$) relative to PID ($\approx 22.6°$), consistent with its smoother attitude curve. MPC exhibited a higher tilt at the 95th percentile at $\approx 28.0°$ but a lower maximum tilt of $\approx 36.3°$ than the PID that resulted in a max of $\approx 41.1°$. This is a pattern that is consistent with move-penalized actuation smoothing. There are fewer sharp spikes, but a sustained attitude during crosswind corrections. None of the cases exhibited persistent actuator saturation, returning 0.0 saturation fractions.

Across a 10-seed Monte-Carlo analysis (Fig. **41** (b)) with frozen equalization scales, PID and LQR remained within the $2.6 \pm 0.25\ m$ band in 90% and 80% of the seeds, whereas MPC diverged ($3.39 \pm 0.59\ m$). Energy per meter travelled was robust to wind realization ($\sigma \approx 0.003\ Wh.m^{-1}$ by 2.3% over PID and LQR ($\sigma \approx 0.470\ Wh.m^{-1}$). Mean power draw remained 4.8 kW across the three controllers; however, the 95th percentile peak was higher for PID and LQR (8.23 kW and 11 kW, respectively) than for MPC (7.05 kW). LQR exhibited the gentlest attitude with $tilt_{95} \approx 21.6°$, PID exhibiting a moderate $tilt_{95} \approx 24.5°$, and MPC, a higher sustained 28.8° but a lower maxima of 36.9°.

For hydrogen-electric UAVs, the controller is also an energy manager. Tracking aggressiveness, turn dynamics, and actuator usage directly shape bus-level mean power, peaks, and energy per meter travelled (Aslam et al., 2023). Those three parameters feed into propulsion decisions. Energy usage maps to hydrogen mass per distance (range and tank sizing), peak power sets stack/buffer requirements, and power transients drive BoP demands (compressor, pumps, thermal loops). In the equalized survey runs, PID/LQR hold the tighter lateral accuracy at essentially identical mean power, while a move-penalized MPC trades a bit of precision for approximately≈ 2.3% lower energy per meter travelled and smoother actuation. A co-design decision for GNC strategies to consider when survey tolerance is the hard constraint, PID/LQR are the right fit; when endurance and BoP smoothness dominate, MPC-style penalization can trim energy per meter and temper load swings**.**

4. Conclusion

Under an equalized survey task (2.6 ± 0.25 m band), PID and LQR met the tracking target with approximately the same mean power; MPC traded approximately 0.6–0.7 m of lateral precision for a 2.3% reduction in energy per meter travelled at similar mean and peak power. For Hydrogen-propulsion vehicles, these results support stack-near-mean sizing with battery buffering of 7–8 kW peaks, and they show that control design affects fuel use and BoP loading, not just trajectory error. In practice, tight-tolerance surveys point to LQR and PID, whereas endurance-biased missions (or BoP-sensitivity) can justify move-penalized MPC. Future work will embed an explicit fuel-cell/BoP with HILs testing to validate the controller–propulsion co-design to refine stack sizing and hydrogen budget predictions.

Acknowledgements

This work was funded by the South African government's Parliamentary Grant.

References Cited

Aslam, Z., Felix, A., Kalyvas, C., & Chizari, M. (2023). Design of a Fuel Cell/Battery Hybrid Power System for a Micro Vehicle: Sizing Design and Hydrogen Storage Evaluation. *Vehicles*, *5*(4). https://doi.org/10.3390/vehicles5040085

Bos, M., Theys, B., Swevers, J., & Pipeleers, G. (2021). Modeling and Identification of Multirotor Drone Dynamics for Onboard MPC Motion Planning. 12th International Micro Air Vehicle Conference (pp. 35-41). PUEBLA. Retrieved from http://www.imavs.org/papers/2021/3.pdf

Chevillard, L. (2017). Regularized fractional Ornstein-Uhlenbeck processes and their relevance to the modeling of fluid turbulence. *Physical Review E*, *96*(3). https://doi.org/10.1103/PhysRevE.96.033111

Dargham, R., Sayouti, A., & Medromi, H. (2015). Euler and Quaternion Parameterization in VTOL UAV Dynamics with Test Model Efficiency. *International Journal of Applied Information Systems*, *9*(8). https://doi.org/10.5120/ijais2015451447

David, I., Galasso, J., & Syriani, E. (2021). Inference of Simulation Models in Digital Twins by Reinforcement Learning. *Companion Proceedings - 24th International Conference on Model-Driven Engineering Languages and Systems, MODELS-C 2021*. https://doi.org/10.1109/MODELS-C53483.2021.00038

Fenn, D. B., Acker, L. W., & Algranti, J. S. (1960). *Flight Operation of a Pump-Fed Liquid-Hydrogen Fuel System.* Cleveland Ohio: National Aeronautics and Space Administatration-Washington

Guevara, B. S., Gandolfo, V. A., Toibero, D. a., & M., J. (2025). Development and Experimental Validation of UAV Digital Twins. *SSRN*, 10.2139/ssrn.5145160.

Ibrahim, N. A., Zakaria, M. Y., & Kamal, A. M. (2023). Simulation of tilt-rotor UAV flight dynamics in horizontal flight. *Journal of Physics: Conference Series*, *2616*(1). https://doi.org/10.1088/1742-6596/2616/1/012006

Kim, J. H., Sukkarieh, S., & Wishart, S. (2006). Real-time navigation, guidance, and control of a UAV using low-cost sensors. *Springer Tracts in Advanced Robotics*, *24*. https://doi.org/10.1007/10991459_29

Lee, S. H., Hur, S. W., Kwak, Y. Y., Nam, Y. H., & Kim, C. J. (2021). Ahead-time Approach to Carrot-chasing Guidance Law for an Accurate Trajectory-tracking Control. *International Journal of Control, Automation and Systems*, *19*(8). https://doi.org/10.1007/s12555-020-0413-x

Massa, E., & Vignolo, S. (2016). Newton–Euler, Lagrange and Kirchhoff formulations of rigid body dynamics: a unified approach. *Meccanica*, *51*(8). https://doi.org/10.1007/s11012-015-0333-7

Pellacani, A., Pace, F., Contreras, R., & da Silva Pais Cabral, F. (2023). *Design, Development, Validation and Verification of the HERA GNC subsystem*. https://doi.org/10.5270/esa-gnc-icatt-2023-007

Peyrl, H., Zanarini, A., Besselmann, T., Liu, J., & Boéchat, M. A. (2014). Parallel implementations of the fast gradient method for high-speed MPC. *Control Engineering Practice*, *33*. https://doi.org/10.1016/j.conengprac.2014.08.010

Reimer, F., Fuchs, M., Herzig, J., Biedermann, J., & Nagel, B. (2024). Investigating the Flexible Hydrogen Aircraft System Interface Design using the XR Co-Design Approach. *34th Congress of the International Council of the Aeronautical Sciences.* Florence: ICAS.