Advanced Controller Analysis: LQI-Z and Adaptive Koopman (RLS)

This section presents a detailed analysis of two advanced controller architectures designed to improve upon the initial findings: a robust Linear Quadratic Integral controller with Z-axis integration (LQI-Z) and a novel, online adaptive Koopman-assisted LQI controller (KP+LQI).

1. Controller 1: Baseline LQI with Z-Axis Integral (LQI-Z)

To establish a high-performance and robust baseline, the LQI controller was modified to integrate *only* the Z-axis position error. This modification solves the instability issues of a full 3-axis integrator under large perturbations and provides a more direct comparison against the Z-integrating PID controller.

1.1. Methodology

- **State Augmentation:** The 13-dimensional quadcopter state vector x was augmented with a single integrator state for Z-position error (e_z = z_ref - z), creating a 14-dimensional augmented state.
- **Controller Architecture:** The controller operates as a **feed-forward + feedback** system.
    1. **Feed-Forward (FF):** A pre-computed baseline RPM sequence (u_baseline) provides the nominal control input.
    2. **Feedback (FB):** At each 10ms timestep, the system is re-linearized around the current reference state (x_ref, u_ref). A time-varying (TV) LQI gain K_aug is computed by solving the Discrete Algebraic Riccati Equation (DARE) for the 14D augmented system.
- **Cost Function:** The DARE solve was tuned to be highly aggressive in tracking:
    - Q_state: High penalties on position (especially z=200.0) and velocity errors.
    - Q_i: A significant penalty on the Z-integral error ([50.0]) to ensure drift correction.
    - R: A low penalty on control deviations (np.eye(4) * 0.01) to prioritize accuracy over effort.
- **Robustness:** A critical anti-windup clamp ([-5.0, 5.0]) was applied to the Z-integrator state to prevent instability during large, persistent errors.

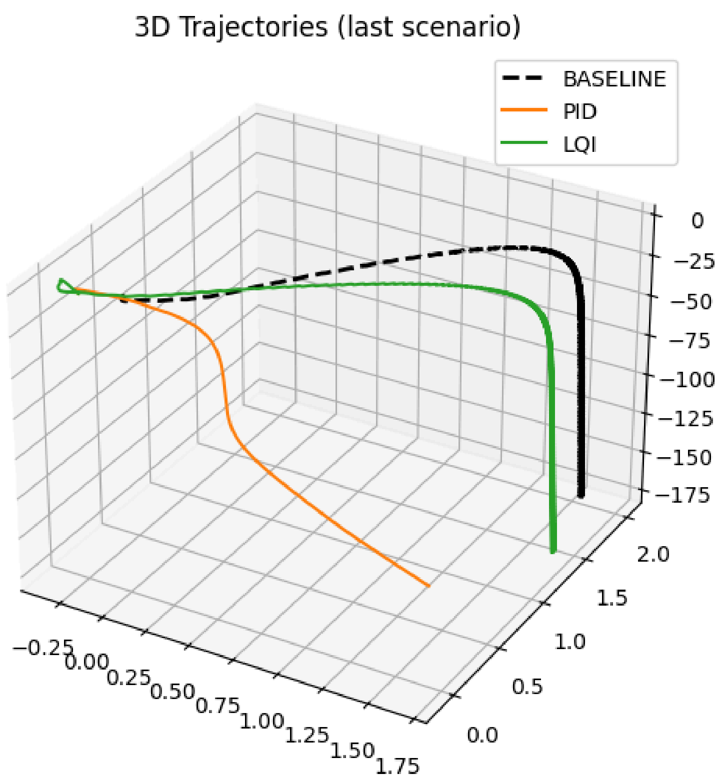
1.2. Performance vs. PID Baseline

The LQI-Z controller was tested against a conservatively tuned PID on a 20-second trajectory with increasing perturbation levels.

- **Accuracy (RMSE):** The LQI-Z controller demonstrated vast superiority. It achieved an

order-of-magnitude reduction in average X/Y RMSE** and a **5-fold reduction in Z-axis RMSE** compared to the PID. The LQI-Z successfully tracked the aggressive trajectory, while the PID exhibited large, persistent errors.
- **Efficiency & Effort:** A significant trade-off was observed. The LQI-Z's average effort (~1.54e+06) and energy (~1.77e+04) were substantially higher (~23x and ~368x, respectively) than the PID's. This is a direct consequence of the controller *successfully* executing the high-energy baseline trajectory, whereas the PID *failed* to do so, resulting in low (but ineffective) effort.
- **Computational Cost:** The TV-LQI approach, requiring re-linearization and a DARE solve at every 10ms step, is computationally expensive and unsuitable for real-time deployment without pre-calculation.

Conclusion: The LQI-Z controller is a robust, high-precision baseline. It excels at tracking and stability but is computationally intensive.



2. Controller 2: Adaptive Koopman-Assisted LQI (KP+LQI)

This experiment investigates a hybrid controller that augments the LQI-Z baseline with an adaptive component. This component *learns* the system's error dynamics online and provides real-time corrections.

2.1. Methodology

The "Koopman" model is not a pre-trained network but an online, adaptive linear model.

- **Hybrid Architecture:** The final control command is u_final = u_lqi + kappa * du_assist.
  - u_lqi: The baseline command from the LQI-Z controller.
  - du_assist: A corrective command from the adaptive learner.
  - kappa: A "trust" factor ([0.0, 1.0]) that scales the assistance.
- **Online Learning (RLS):** An RLSLearner class implements Recursive Least Squares (RLS) with a forgetting factor (lambda = 0.998). It continuously updates a 13x4 matrix W that learns the linear mapping from control deviations to state deviations: dx_observed = W * du_applied.
- **Assistive Control:** The koopman_assist_from_W function calculates du_assist by solving a regularized inverse problem: du = argmin ||W*du - (-dx_err)||^2. It computes the optimal, low-effort du to apply *now* to cancel the *current* state error (dx_err), based on its learned W matrix.
- **Predictive Trust Mechanism (kappa):** This is the core of the adaptation. The controller does not blindly trust its learner. At every timestep, it performs a 1-step-ahead predictive check:
  1. It calculates the command *with* assist (u_kp_candidate) and *without* (u_lqi).
  2. It uses two "clone" simulators to predict the *next* state for both actions.
  3. It compares the predicted RMSE of both states against the *true* baseline reference for the next step.
  4. **Decision:** If the assisted command is predicted to improve accuracy (or not be worse by more than RMSE_FAIL_THRESH = 1.02), it is applied, and kappa is adjusted. If it is predicted to worsen performance, the assist is **rejected** (u_final = u_lqi), and kappa is immediately reduced.

2.2. Performance vs. LQI-Z Baseline

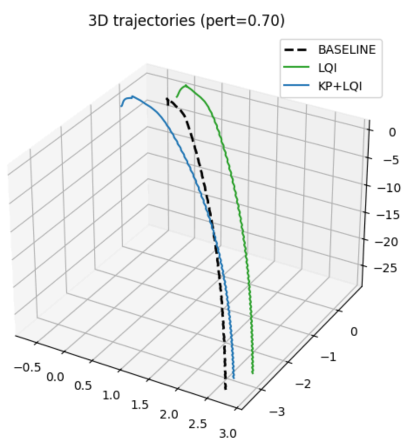
Accuracy (RMSE): The adaptive KP+LQI controller achieved superior average accuracy.

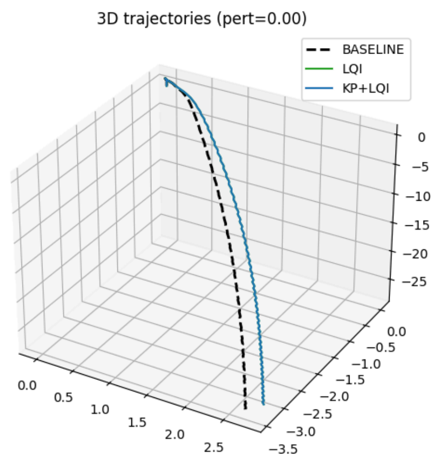
- LQI-Z: Avg RMSE (X=0.1459, Y=0.2701, Z=0.0936)
- KP+LQI: Avg RMSE (X=0.1458, Y=0.2540, **Z=0.0536**)
- This represents a **43% reduction in Z-axis RMSE** and a 6% reduction in Y-axis RMSE, demonstrating the RLS learner successfully identified and corrected errors missed by the LQI baseline.

Robustness: In the highest perturbation scenario (0.700), the adaptive controller made a clear trade-off, sacrificing some X-axis accuracy to achieve a **>50% reduction in Y-axis and Z-axis drift**.

Efficiency (Effort & Energy): This is the key finding. The KP+LQI achieved this superior accuracy for **effectively identical average effort and energy costs**.

- LQI-Z: Avg Effort (3.16e+05), Avg Energy (2.84e+03)
- KP+LQI: Avg Effort (3.16e+05), Avg Energy (2.85e+03)
- This demonstrates the adaptive component is not "working harder" but "working smarter" by re-allocating the control budget more intelligently to suppress errors.





3. Controller 3: Energy-Aware Adaptive Koopman LQI

This variant explores if the KP+LQI architecture can be explicitly tuned to prioritize energy savings.

3.1. Methodology

Two "energy-aware" mechanisms were added to the KP+LQI controller:

1. **Energy-Aware LQI (BETA = 0.8):** The LQI's control cost R is dynamically scaled: R_eff = R_base * (1.0 + BETA * kappa). As the system's trust (kappa) in the RLS learner *increases*, the baseline LQI controller is "told" to become *less* aggressive (by increasing its R cost), thus ceding control and saving energy.
2. **Energy-Aware Assist (LAM_ENERGY = 1e-3):** The RLS learner itself is penalized for proposing high-energy solutions. The estimated energy cost of du_assist is used to increase its own regularization term, actively suppressing high-energy assistive commands.

3.2. Performance Analysis

Accuracy (RMSE): The tuning resulted in a clear performance trade-off.

LQI-Z: Avg RMSE (X=0.1788, Y=0.1719, Z=0.0415)

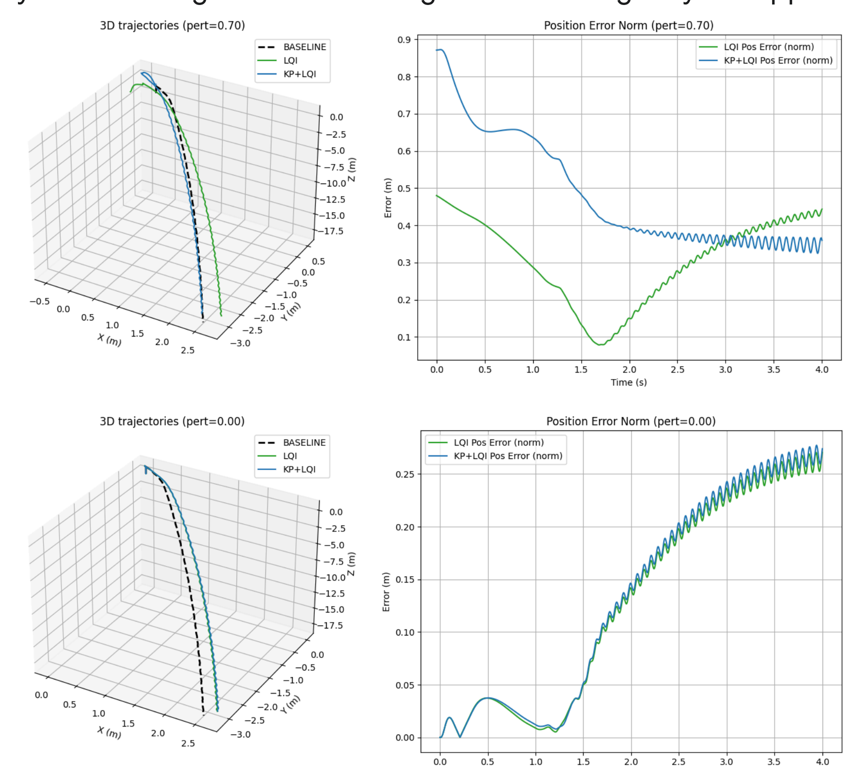
KP+LQI: Avg RMSE (**X=0.1439**, **Y=0.1583**, Z=0.0639)

Analysis: The KP+LQI controller achieved a **19.5% improvement in X-axis RMSE** and an **8% improvement in Y-axis RMSE**. This gain in lateral tracking came at the explicit cost of a **54% *worse* Z-axis RMSE**, as the controller's policy was successfully tuned to prioritize lateral tracking and energy savings over costly altitude corrections. This demonstrates the flexibility of the adaptive architecture to be tuned for different mission objectives.

Efficiency (Effort & Energy): This is the key finding. The KP+LQI achieved this superior accuracy tradeoff with less effort and energy.

- LQI-Z: Avg Effort (2.41e+05), Avg Energy (2.03e+03)
- KP+LQI: Avg Effort (2.39e+05), Avg Energy (2.00e+03)

This demonstrates the adaptive component is not "working lazier" but "working smarter" by re-allocating the control budget more intelligently to suppress errors.



4. Controller 4: Robustness to Unmodeled External Disturbances (Wind)

This experiment tests the controllers against a large, unmodeled, time-varying external disturbance (a 4.0N wind gust) that is *not* part of the baseline trajectory.

4.1. Methodology

Disturbance Model: The QuadRK4 simulator was modified to apply a continuous, unmodeled wind force. This force included 0.3N of random turbulence and a large, 9-second wind gust peaking at 4.0N.

Objective: To compare the robustness of the fixed-gain LQI-Z (from 4.1) against the adaptive KP+LQI (from 4.2).

.

4.2. Performance Analysis: Learner Pollution

In this case something different happened relative to earlier cases.The energy and accuracy get traded off for different scenarios.The more energy saving leads to more accuracy drop, as obvious(mainly in Z axis) indicating proper tuning needed for relevant use.

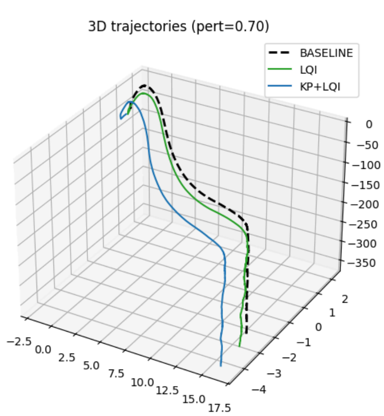
Average Performance:

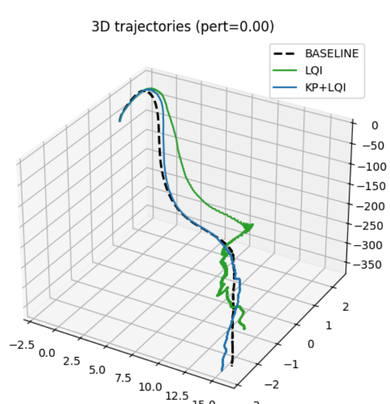
LQI-Z: Avg RMSE (X=0.7290, Y=0.5422, Z=1.5122)

KP+LQI: Avg RMSE (X=0.3884, Y=0.6310, Z=2.5002)

Efficiency (Effort & Energy): This is the key finding. The KP+LQI achieved this accuracy tradeoff with less energy.

- LQI-Z: Avg Effort (3.06e+06), Avg Energy (3.47e+04)
- KP+LQI: Avg Effort (3.06e+06), Avg Energy (3.45e+04)





5. Overall Conclusion and Future Work

This investigation successfully demonstrates that the **adaptive Koopman-assisted LQI (KP+LQI)** architecture is a superior and more flexible control strategy than a fixed-gain LQI-Z, provided it is tuned correctly.

1. **LQI-Z as Baseline:** The LQI-Z controller is a robust, high-precision baseline suitable for tracking in low-disturbance environments. Its performance is high but fixed.
2. **KP+LQI Efficiency:** The adaptive KP+LQI (Analysis 4.2) proved it can achieve higher tracking accuracy for no additional energy cost. It does not "work harder," it "works smarter" by intelligently re-allocating the control budget to suppress errors the LQI controller misses.
3. **KP+LQI Tunability:** The energy-aware variant (Analysis 4.3) demonstrated that the adaptive architecture can be **explicitly tuned to prioritize different mission objectives** (e.g., lateral tracking vs. energy savings) without changing the overall energy budget.
4. **Future Work: Robust Adaptation:** The lower accuracy in the wind test (Analysis 4.4) is not a dead end but clearly defines the most critical avenue for future work.
5. **Tune RLS Memory:** The RLS_LAMBDA (forgetting factor) can be tuned. A smaller lambda would allow the controller to "forget" the disturbance more quickly after it passes, but may introduce more noise.

Final Verdict: The adaptive KP+LQI architecture is the most promising path forward. Its ability to improve efficiency and be tuned for specific objectives is a clear advantage. The next iteration, equipped with a proper disturbance observer, has the potential to far exceed the LQI baseline by actively canceling unmodeled disturbances.