

TAG-K: Tail-Averaged Greedy Kaczmarz for Computationally Efficient and Performant Online Inertial Parameter Estimation

Shuo Sha¹, Anupam Bhakta^{2*}, Zhenyuan Jiang^{1*}, Kevin Qiu^{1*},
Ishaan Mahajan¹, Gabriel Bravo-Palacios^{3,4}, Brian Plancher^{3,4}

Abstract—Accurate online inertial parameter estimation is essential for adaptive robotic control, enabling real-time adjustment to payload changes, environmental interactions, and system wear. Traditional methods often struggle to track abrupt parameter shifts or incur high computational costs, limiting their effectiveness in dynamic environments and for computationally constrained robotic systems. We introduce TAG-K, a lightweight extension of the Kaczmarz method that combines greedy randomized row selection for rapid convergence with tail averaging for robustness under noise and inconsistency. This design enables fast, stable parameter adaptation while retaining the low per-iteration complexity inherent to the Kaczmarz framework. We evaluate TAG-K in synthetic benchmarks and quadrotor tracking tasks against RLS, KF, and other Kaczmarz variants. TAG-K achieves 1.5×–1.9× faster solve times on laptop-class CPUs and 4.8×–20.7× faster solve times on embedded microcontrollers. More importantly, these speedups are paired with improved robustness to measurement noise and a 25% reduction in estimation error, leading to nearly 2× better end-to-end tracking performance. Website, documentation, and code available at: <https://a2r-lab.org/TAG-K/>.

I. INTRODUCTION

Model-based control has enabled significant advances across humanoids, quadrupeds, and aerial robots [1], [2], [3], [4], [5], [6], [7]. A prerequisite for such precise prediction and stable control is accurate *system identification* (SysID), i.e., determining both the structure and parameters of the dynamics model. While the model structure is often known, inertial parameters such as masses, centers of mass, and inertia tensors vary during operation due to payloads, contacts, or reconfiguration. *Online parameter estimation* addresses this challenge by updating parameters in real time, allowing robots to adapt their dynamics models to changing conditions, enabling advances in manipulation [8], [9], adaptive trajectory tracking [10], [11], and aerial robotics [12], [10].

However, traditional online estimators face fundamental limitations. Recursive Least Squares (RLS) efficiently handles measurement noise and can accommodate slowly varying parameters through a forgetting factor [13], [14], [15], [16], but adapts poorly to abrupt parameter shifts. The Kalman Filter (KF), as a more general state-space estimator, explicitly models both process and measurement noise but struggles with time-varying process noise (e.g., unmodeled disturbances) and its updates are more computationally expensive [17], [18], [19]. Learning-based approaches provide

a data-driven alternative by either fitting the dynamics model end-to-end or embedding physics constraints/parameters during training [20], [21], [22]. While these methods can support high-order complex nonlinear systems, like deformable or soft-body objects [23], [24], they may not always offer consistent and reliable physics-based guarantees for systems with well-established models [25], [26]. Across all of these methods, the computational complexity of these approaches limits their applicability, particularly for embedded, compute-limited systems.

This motivates the development of lightweight, adaptive, and robust alternatives. One promising direction is to enhance classical methods with improved underlying solvers, such as the Kaczmarz family of algorithms [27], [28], [29], [30]. These methods solve linear systems by sequentially assimilating relevant measurements through projections onto valid manifolds, enabling computationally efficient solutions without requiring access to the entire data matrix [31], and have found recent success in robotics applications [32].

As such, we introduce the *Tail-Averaged Greedy Randomized Kaczmarz* (TAG-K) algorithm for online parameter estimation in robotic systems. TAG-K leverages the Kaczmarz family’s efficiency in projecting onto subsets of system manifolds and augments it with greedy row selection for rapid closed-loop adaptation and tail averaging for robustness to noise. This yields a high-performance framework well-suited for online estimation on embedded robotic hardware.

We evaluate TAG-K on challenging quadrotor tracking tasks, where it achieves 1.5×–1.9× faster solve times on laptop-class CPUs, and more importantly, improved resilience to measurement noise, and a 25% reduction in estimation error, leading to nearly 2× better end-to-end tracking performance. To further validate our computational efficiency, and demonstrate suitability for embedded deployment, we benchmark TAG-K on synthetic parameter estimation tasks on an embedded microcontroller. TAG-K achieves 4.8×–20.7× faster solves, highlighting the importance of light-weight online estimation algorithms for systems with limited processing power.

II. BACKGROUND AND RELATED WORK

A. Online Parameter Estimation

Consider a robot with n_d degrees of freedom and n_b rigid bodies. Its dynamics follow the Newton–Euler equations:

$$\mathbf{M}(\boldsymbol{\theta}, \mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\boldsymbol{\theta}, \mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\boldsymbol{\theta}, \mathbf{q}) = \mathbf{F}, \quad (1)$$

¹ School of Applied Science, Columbia University.

² Columbia College, Columbia University.

^{3,4} Barnard College, Columbia University and Dartmouth College.

* These authors contributed equally.

Corresponding Author Contact: plancher@dartmouth.edu

where $\theta \in \mathbb{R}^{10n_b}$ is the inertial parameters for each rigid body (the mass and 3×3 inertia-matrix components), $\mathbf{q} \in \mathbb{R}^{n_d}$ is the joint-space configuration, $\mathbf{M}, \mathbf{C} \in \mathbb{R}^{n_d \times n_d}$ are the mass and Coriolis force matrices, and $\mathbf{g}, \mathbf{F} \in \mathbb{R}^{n_d}$ are the gravity and net force vectors.

A key property is that Eq. (1) is *linear* in the inertial parameters, giving the regression model:

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \theta = \mathbf{F}, \quad (2)$$

where \mathbf{A} is the regressor matrix derived from kinematic measurements [33], [34].

For the remainder of the paper, we embed the online parameter estimation problem into the following stochastic state-space framework:

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k; \theta_t) + \omega_k, & \omega_k &\sim \mathcal{N}(0, \mathbf{R}), \\ \mathbf{F}_t &= \mathbf{A}_t \theta_t + \nu_t, & \nu_t &\sim \mathcal{N}(0, \mathbf{Q}). \end{aligned} \quad (3)$$

where \mathbf{x}_k and \mathbf{u}_k denote the robot state and control input, and ω_k, ν_k denote the process and measurement noise. We distinguish two time indices: a fast control loop indexed by k , and a slower estimation loop indexed by t , where parameters θ_t are updated every κ control steps (Fig. 1).

Applying a forward Euler approximation, the discrete-time dynamics model is defined as follows:

$$f(\mathbf{x}_k, \mathbf{u}_k; \theta) = \begin{bmatrix} \mathbf{q}_k + \Delta t \dot{\mathbf{q}}_k \\ \dot{\mathbf{q}}_k + \Delta t \ddot{\mathbf{q}}_k \end{bmatrix} \quad (4)$$

with the online parameter estimation cast as a weighted least-squares problem:

$$\hat{\theta}_t = \arg \min_{\theta_t} \|\mathbf{F}_t - \mathbf{A}_t \theta_t\|_2^2. \quad (5)$$

The central challenge, therefore, is to update $\hat{\theta}_t$ in a manner that is *efficient*, running in real time on resource-constrained platforms; *rapid*, adapting to abrupt parameter changes during operation; and *stable*, remaining robust under diverse real-world noise conditions.

B. Traditional Online Parameter Estimation Methods

1) *Recursive Least Squares (RLS)*: RLS computes parameter updates by minimizing a cumulative squared prediction error and then updating the inverse system covariance via the matrix inversion lemma. Modern variants attempt to improve robustness and flexibility: Recursive Total Least Squares (RTLTS) mitigates bias from noisy regressors [35], [36], variable-forgetting RLS adapts weighting to track varying parameters [16], [37], and kernelized RLS maps the regression into nonlinear feature spaces [38]. Despite these extensions, a fundamental disadvantage of RLS remains: the forgetting factor must be constrained to a narrow, hand-tuned memory window, which in practice lies extremely close to one [37], [16]. This makes the method poorly suited to handling abrupt shifts, since smaller forgetting factors cause instability while larger ones prevent adaptation. As a result, adaptive RLS variants remain not adaptive enough for versatile online estimation.

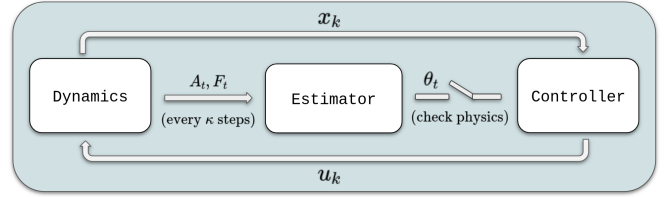


Fig. 1: **Online Parameter Estimation Overview.** Closed-loop estimation with fast control updates and slower parameter adaptation. The dynamics model maps control inputs \mathbf{u}_k to states \mathbf{x}_k , which generate subsequent controls. Every κ steps, the estimator updates controller parameters θ_t via (3). A physics-based safety check (15) safeguards adaptation.

2) *The Kalman Filter (KF)*: KF interprets parameters as stochastic states and computes the gain by balancing prior uncertainty against measurement noise. Extensions include: adaptive covariance adjustment [39], Ensemble Kalman Filters (EnKF) for high-dimensional nonlinear systems [40], and square-root formulations for numerical stability [41], [42]. While such methods improve robustness, they introduce significant computational burden and remain sensitive to initialization and noise-model assumptions, making them less practical for embedded online estimation.

C. Randomized Kaczmarz Methods:

The Randomized Kaczmarz (RK) algorithm [28] builds upon the classical iterative Kaczmarz method [43]. It solves a linear system $\mathbf{A}\theta = \mathbf{b}$ by randomly selecting a row \mathbf{a}_i^T with probability proportional to $\|\mathbf{a}_i\|_2^2$ and then orthogonally projecting the current estimate onto the row's hyperplane $\{\theta : \langle \mathbf{a}_i, \theta \rangle = b_i\}$, yielding an expected exponential rate of convergence for consistent linear systems [28]. The iterative update is given by:

$$\theta_{t+1} = \theta_t + \frac{r_i}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i, \quad r_i = b_i - \langle \mathbf{a}_i, \theta_t \rangle. \quad (6)$$

Several recent variants of these methods have emerged to improve convergence and robustness. Greedy Randomized Kaczmarz (GRK) [31] biases selection toward rows with larger residuals, prioritizing directions where the estimate is most inconsistent with the data and converging at a faster exponential rate than RK for consistent linear systems. Tail-Averaged Randomized Kaczmarz (TARK) [44] further stabilizes performance by averaging recent iterates, smoothing fluctuations and converging at a hybrid rate that balances exponential and polynomial convergence for inconsistent linear systems arising from noise or unmodeled dynamics. Together, these mechanisms extend RK into regimes relevant for online parameter estimation.

III. RECURSIVE ESTIMATION FROM RLS AND KF TO KACZMARZ ALGORITHMS

To provide further motivation before explaining our approach, we look at online parameter estimation from a unified recursive-gain perspective. Recursive algorithms can be expressed in the generic form:

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \mathbf{K}_t \varepsilon_t, \quad (7)$$

TABLE I: **Comparison of Parameter Estimation Methods.** Columns summarize desirable properties of online estimators, and rows list representative approaches. TAG-K satisfies all listed criteria while maintaining linear-time complexity per iteration.

Methods	Features				
	Low Cost	Robust to Process Noise	Handles Fast Changes	Physics -Reliable	Computational Complexity
RLS	✗	✗	✗	✓	$O(m^3)$
KF	✗	✓	✗	✓	$O(m^3)$
Learning	✗	✓	✓	✗	-
TAG-K	✓	✓	✓	✓	$T \cdot O(mn)$

where $\hat{\theta}_t$ is the current estimate, $\varepsilon_t = \mathbf{b}_t - \mathbf{A}_t \hat{\theta}_t$ is the prediction error, and \mathbf{K}_t is the gain matrix that determines how new data influences each update. The distinction among methods lies not in this form, but in how \mathbf{K}_t is constructed, directly impacting computational cost and adaptivity.

RLS constructs \mathbf{K}_t from recursive covariance minimization, with the forgetting factor $\lambda \in (0, 1]$ and where $\mathbf{A}_t \in \mathbb{R}^{m \times n}$ denotes the block of m regressors at time t :

$$\mathbf{K}_t^{\text{RLS}} = \mathbf{P}_{t-1} \mathbf{A}_t^\top (\lambda \mathbf{I} + \mathbf{A}_t \mathbf{P}_{t-1} \mathbf{A}_t^\top)^{-1}, \quad (8)$$

$$\mathbf{P}_t = \lambda^{-1} (\mathbf{I} - \mathbf{K}_t^{\text{RLS}} \mathbf{A}_t) \mathbf{P}_{t-1}.$$

Because we operate in a block-update setting that naturally comes with online parameter estimation, each step requires a matrix inversion of size m , leading to an $O(m^3)$ time complexity (as noted in Table I) [45].

KF, on the other hand, interprets parameters as stochastic states, propagating and updating error covariances with block measurements $\mathbf{A}_t \in \mathbb{R}^{m \times n}$. In this case, the gain matrix is:

$$\mathbf{K}_t^{\text{KF}} = \mathbf{P}_{t|t-1} \mathbf{A}_t^\top (\mathbf{A}_t \mathbf{P}_{t|t-1} \mathbf{A}_t^\top + \mathbf{R}_t)^{-1}, \quad (9)$$

where, due to the block formulation, each update requires inversion of an $m \times m$ matrix, incurring $O(m^3)$ complexity per step (again noted in Table I) [17]. While optimal under linear-Gaussian assumptions, KF variants remain expensive and fragile under non-stationary conditions.

Finally, Kaczmarz updates also take the form of Eq. (7), but now each update is based on a single row \mathbf{a}_i^\top . The prediction error ε_t reduces to a scalar residual as in Eq. (6), and the corresponding gain is the row-dependent vector

$$\mathbf{K}_t^{\text{RK}} = \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|_2^2}, \quad (10)$$

Unlike RLS and KF, which operate on the full regressor matrix at each update, require only $T \leq m$ iterations in practice and need not traverse the entire row space. As each TAG-K iteration is dominated by the greedy selection, at $O(mn)$ time per update, a sweep of T rows requires $T \cdot O(mn)$ operations (see Table I) [31], [28], making it particularly attractive for embedded, resource-constrained robots. Moreover, online parameter estimation typically yields overdetermined systems for which Kaczmarz methods remain fast and accurate [29], [43], [28], as new equations accumulate faster than parameters are updated.

Algorithm 1 TAG-K ($\mathbf{A}, \mathbf{b}, \hat{\theta}_0, T, t_b, \hat{\theta}_s = \mathbf{0} \rightarrow \hat{\theta}_{t+1}$)

- 1: **for** $t = 0$ in $\dots T$ **do**
 - 2: // Greedy Row Selection
 - 3: $\epsilon_t, \tau_t \leftarrow$ Compute greedy candidates via (11), (12)
 - 4: $i_t \leftarrow$ Sample via (13)
 - 5: // Online Update
 - 6: $\hat{\theta}_{t+1}(i_t) \leftarrow$ KZ Gain update via (6), (10)
 - 7: // Tail Averaging
 - 8: **if** $t \geq t_b$ **then**
 - 9: $\hat{\theta}_s \leftarrow \hat{\theta}_s + \hat{\theta}_{t+1}$
 - 10: $\hat{\theta}_{avg} \leftarrow$ Tail-averaging via (14)
 - 11: **return** $\hat{\theta}_{avg}$
-

IV. TAIL-AVERAGED GREEDY KACZMARZ

In this section, we introduce the Tail-Averaged Greedy Kaczmarz (TAG-K) algorithm, a Kaczmarz variant tailored for online parameter estimation for computationally constrained robotic systems. TAG-K combines greedy randomized row selection with tail averaging, enabling rapid closed-loop adaptation, robustness to measurement and process noise, and lightweight computational complexity. This combination makes it well-suited for real-time robotic applications where both speed and reliability are essential.

The foundation of TAG-K lies in the broader Kaczmarz family of algorithms, which update solutions through projection-based steps that sequentially assimilate measurements. Compared to Recursive Least Squares (RLS) and Kalman Filters (KF), which often suffer from higher computational cost or instability, Randomized Kaczmarz (RK) and its variants provide adaptive and efficient updates. RK achieves exponential convergence in low-noise regimes under standard row-norm sampling assumptions [28], [29]. In noisier or streaming settings, its randomized sampling can naturally emphasize *more informative* constraints: in the standard scheme, row i is selected with probability $p_i = \|\mathbf{a}_i\|_2^2 / \|\mathbf{A}\|_F^2$, so higher-norm (higher-energy) measurements are enforced more frequently, which can empirically accelerate adaptation [28], [29]. Moreover, in the presence of measurement noise or inconsistency, RK is known to converge to an error neighborhood whose radius depends on the noise level [46]. TAG-K further extends these benefits by adapting the greedy row selection in GRK [31] and the tail averaging in TARK [44] to achieve both faster convergence and improved robustness. As summarized in Table I, TAG-K outperforms traditional RLS, KF, and learning-based methods across several axes. It offers low computational cost, robustness to noise, responsiveness to sudden system changes, and physical reliability, all while maintaining the lowest-order computational complexity.

Before performing the Kaczmarz gain update rule defined in (6), we select the target row(s) through a *greedy selection* step which prioritizes rows that contribute most to the current

residual error. First, a threshold parameter ϵ_k is computed to adaptively scale residual importance:

$$\epsilon_t = \frac{1}{2} \left(\frac{\max_i (r_i^2 / s_i)}{R^2} + \frac{1}{\|A\|_F^2} \right), \quad (11)$$

where $s_i = \|A^{(i)}\|_2^2$ and $R = \|r\|_2$. Then, a candidate set of indices with the largest scaled errors is formed,

$$\tau_t = \{i \mid r_i \geq \epsilon_t R s_i\}, \quad (12)$$

where finally, row i_t is sampled with probability

$$\mathbb{P}(i) = \frac{r_i^2}{\sum_{j \in \tau_t} r_j^2}, \quad i \in \tau_t. \quad (13)$$

We modify the gain update (10), by incorporating these new sampling probabilities (13) instead of using probabilities weighted by row norms as seen in the classical RK formulation. This step biases the gain such that it updates towards directions with higher instantaneous error, enabling faster convergence. In online estimation, such targeted updates allow rapid adaptation from only a short data window, which is essential for closed-loop robotic control.

While greedy selection improves adaptation speed, it can also increase estimator variance in the presence of noise or inconsistency. To mitigate this, TAG-K employs *tail averaging* [44]: after a burn-in period t_b , the final estimate is formed as the average of the most recent iterates,

$$\hat{\theta}_{avg} = \frac{1}{t - t_b} \sum_{s=t_b}^t \hat{\theta}_s, \quad (14)$$

which smooths fluctuations and guarantees convergence to a least-squares solution even in inconsistent systems [44]. For robotics applications, where process and sensor noise are unavoidable, this averaging provides a critical safeguard against estimator instability.

V. EXPERIMENTS AND RESULTS

We evaluate TAG-K against a suite of baselines to assess parameter-estimation accuracy, computational runtime, and downstream robotic performance. Specifically, we aim to answer the following questions:

- Q1:** Does TAG-K produce faster and more accurate parameter estimates than existing baselines?
- Q2:** Which features of estimation performance or algorithmic design most strongly influence performance?
- Q3:** Does TAG-K reduce on-device compute compared to baselines on resource-constrained hardware?

To address these questions, we benchmark TAG-K and baselines on online inertial parameter estimation in simulation (V-A.2), study the contribution of key design choices and estimation-time behavior via targeted analyses and ablations (V-A.3, V-A.4), and measure runtime and resource usage on microcontrollers (V-B).

A. Quadrotor Under Unknown Payload

1) *Experiment Setup:* We first perform a three-part evaluation of our approach leveraging a simulated model of

a Crazyflie 2.1 [47] quadrotor, a widely used testbed in edge robotics research [48], [49], [50], [51], and evaluate its performance under unknown payload events to address **Q1** and **Q2**. These experiments were conducted on a laptop with a 3.3 GHz AMD Ryzen 9 5900HS CPU.

In each experiment, the quadrotor starts at a randomized location, and tracks one of five reference trajectories: a Figure-8, Circle, Spiral, Helix, or Ellipse. Across each, the peak linear and angular speeds are limited to 0.5 m/s and 0.5 rad/s, respectively. For stabilization and trajectory tracking, we employ a standard LQR controller [52] running at 50 Hz, while the parameter estimator runs at 2.5 Hz using a 5-frame history of the 10 inertial parameters (the body mass plus 3×3 inertia-matrix components). The estimator minimizes the weighted least-squares loss in (5), while the LQR controller minimizes the mean squared tracking error $\|e\|_2^2$. Each experiment lasts 20 s. To emulate a payload-transfer task, a payload equal to 30–50% of the vehicle mass is attached at an offset of 20–30% of the arm length at a random time in [4, 6] s and removed at a random time in [12, 14] s. At each parameter-estimation step, we re-linearize the dynamics about the current operating point so that updated inertial parameters are consistently reflected in the control-affine model.

To model real-world sensor imperfections, we inject Gaussian measurement noise at four levels, ranging from noise-free to a maximum standard deviation of 2.5 cm/s in velocity and 0.25 cm/s² in acceleration, applied to the state inputs of both controller and estimator. For baselines, we conduct a grid search over key hyperparameters (e.g., forgetting factor, initial covariance magnitude, Kalman-filter noise priors) and report the best configuration for each method under the end-to-end tracking objective [53], [54]. To prevent catastrophic outcomes for *baseline* methods, we use a lightweight inertial-parameter safety filter that rejects physically invalid estimates, where $L_{arm} = 0.0325$ and $\alpha = 10^{-4}$:

$$\{m > 0, \|\mathbf{r}_{CoM}\| \leq L_{arm}, I \succ 0, \lambda_{\max}(I) \leq \alpha\} \quad (15)$$

2) *Estimation and End-to-End Performance:* We visualize a representative trial from this experiment set in Figure 2 (upper left). In this trial, each estimator is initialized from the same start state and tracks the dotted figure-8 reference toward a pre-set goal state (both marked with purple triangles) under injected measurement noise using the aforementioned LQR controller. Payload add/drop events are marked with gray rings. After each event, TAG-K, by its greedy mechanism, rapidly adapts its inertial parameters and re-converges to the reference, whereas baselines mis-estimate the parameters, drift persistently, and in several cases abort due to compounding errors (marked with black Xs). In this specific trial, only TAG-K completes the task. However, baseline methods can complete the task in many other trials.

Figure 2 (bottom) summarizes the aggregated outcomes across all 2,000 trials per estimator method. The bottom left panel presents the time evolution of the average parameter estimation error, while the bottom right panel shows the cumulative distribution function (CDF) of the end-to-end

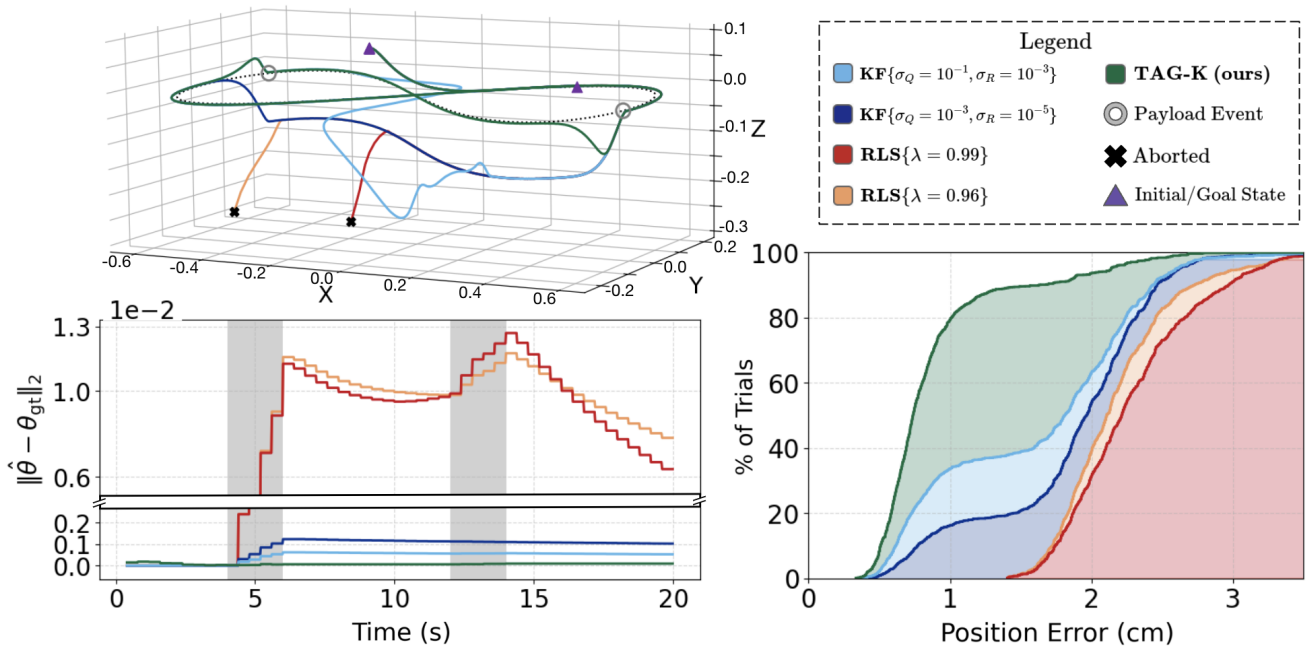
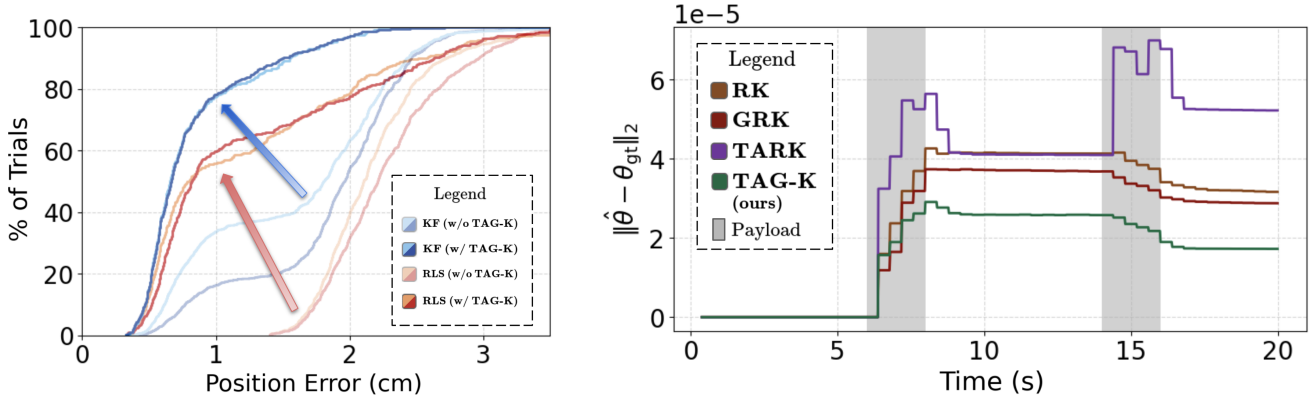


Fig. 2: **Tracking and Estimation Performance.** *Upper Left:* Example single trial of a quadrotor tracking a figure-8 reference trajectory (dotted line, units in meters) with unknown payload add/drop events (gray rings). *Lower Left:* Prediction error timeseries comparing TAG-K with baseline estimators averaged over 2,000 trials per estimator. Grey shaded regions denote payload add/drop events, highlighting TAG-K’s rapid and consistent re-convergence after disturbances. *Lower Right:* Cumulative distribution function (CDF) of position tracking error over 2,000 trials per estimator, averaged across four different noise levels. TAG-K achieves the lowest overall error, demonstrating superior adaptation and end-to-end tracking performance.



(a) Position error CDF before (light) and after (dark) substituting the first post-event estimate with TAG-K’s estimate. (b) Prediction error comparison across Kaczmarz variants. Grey regions indicate payload addition/removal events.

Fig. 3: **Ablation of Early Adaptation and Algorithmic Design.** Left: Substituting the first post-event estimate from baseline estimators with TAG-K’s estimate improves tracking error, highlighting the importance of rapid early adaptation. Right: Comparison with Kaczmarz variants shows that combining greedy row selection and tail averaging yields superior robustness and adaptation under abrupt payload changes (grey regions).

tracking position error. Together, these plots highlight the differences in both convergence behavior and final tracking accuracy. We observe that RLS and KF consistently maintain higher parameter estimation errors, which directly translate into larger tracking deviations. By contrast, TAG-K rapidly adapts after payload events, quickly reducing estimation error and maintaining accurate parameters. This fast and stable adaptation yields markedly lower tracking errors across the full distribution of trials, demonstrating our method’s ability to robustly handle abrupt system changes.

Table II provides a more detailed breakdown by noise

regime, further reinforcing these observations. Note that we consider a trial aborted if the positional tracking error exceeds 30 cm and is successful if the error remains below 5 cm at 10 steps after the payload event, without being previously aborted. We use this success metric to avoid crediting estimators that fail to adapt during the payload addition event but coincidentally exhibit low estimation error after the payload is dropped. Across all noise levels, TAG-K consistently achieves the lowest mean estimation and end-to-end position errors, alongside the highest success rates. While KF shows slightly lower abort rates under extreme noise,

TABLE II: **Quadrotor Tracking under Varying Measurement Noise.** We run 500 trials per noise level per algorithm across five reference trajectories with randomized initial states, trajectory parameters, event times, and payload mass/location. RLS-low/high is $\lambda=0.99/0.96$; KF-low/high is $\sigma_Q=10^{-3}/10^{-1}$, $\sigma_R=10^{-5}/10^{-3}$. $(t+1)$ Est. Error is the mean estimation error one step after the payload event.

Noise	Algorithm	Estimation Accuracy			Tracking Performance		Iteration Time (μ s)	
		Pos. Error (cm)	Mean Est. Error	$(t+1)$ Est. Error	Success	Aborted	Median	P95
none	RLS-low	2.13	8.47e-03	1.10e-02	0.0%	11.8%	63.1	92.1
	RLS-high	2.18	8.10e-03	1.00e-02	0.0%	22.2%	64.4	90.0
	KF-low	1.79	1.10e-03	1.42e-03	20.2%	4.4%	74.1	94.0
	KF-high	1.40	4.62e-04	5.99e-04	48.2%	3.6%	73.9	97.8
	TAG-K (ours)	0.84	3.92e-04	7.11e-05	87.8%	3.6%	40.4	59.6
low	RLS-low	2.13	8.45e-03	1.10e-02	0.0%	12.0%	62.5	94.5
	RLS-high	2.18	8.08e-03	1.00e-02	0.0%	22.0%	64.2	91.4
	KF-low	1.78	1.09e-03	1.39e-03	21.6%	3.6%	74.3	96.2
	KF-high	1.40	4.61e-04	5.97e-04	49.2%	3.2%	74.9	101
	TAG-K (ours)	0.83	3.14e-04	7.06e-05	87.8%	3.6%	43.0	60.3
medium	RLS-low	2.18	7.47e-03	8.99e-03	0.2%	17.8%	63.7	94.0
	RLS-high	2.30	6.63e-03	8.14e-03	0.2%	39.4%	66.2	93.7
	KF-low	1.76	7.70e-04	9.26e-04	28.4%	5.4%	75.1	113
	KF-high	1.56	4.42e-04	5.60e-04	43.6%	3.8%	75.6	118
	TAG-K (ours)	0.84	3.12e-04	9.89e-05	87.4%	4.0%	43.7	60.9
high	RLS-low	2.25	5.75e-03	6.24e-03	0.6%	23.0%	63.5	94.4
	RLS-high	2.47	5.00e-03	5.46e-03	0.4%	49.4%	63.2	90.8
	KF-low	2.05	4.47e-04	4.84e-04	17.4%	15.0%	73.9	96.5
	KF-high	2.05	3.72e-04	4.06e-04	22.0%	17.0%	73.5	97.8
	TAG-K (ours)	1.03	2.86e-04	8.46e-05	58.4%	31.4%	42.5	58.2

its diminished success rates reflect closed-loop instability. In contrast, TAG-K maintains stable tracking and accurate parameter estimation even in high-noise conditions. Overall, TAG-K offers the most effective balance of speed, accuracy, and robustness, outperforming all evaluated methods.

3) *Importance of Early Adaptation:* Across all trials, inaccurate parameter estimates quickly push the quadrotor into noisier, less stable states, with errors compounding over time. This creates a strong dependence on rapid correction after a disturbance or payload change. We therefore hypothesize that early adaptation, specifically the accuracy of the very first updated estimate, plays a decisive role in overall closed-loop performance. To evaluate the role of the first estimates, we ran controlled trials in which two estimators were executed in parallel. Immediately following a payload event, the baseline estimator’s first update was replaced by TAG-K’s estimate, after which the baseline method continued independently.

As shown in Figure 3a, initializing baselines with TAG-K’s first post-event estimate dramatically improves performance, confirming that early correction quality dictates long-term tracking accuracy. These findings are consistent with Table II, where TAG-K achieves an order-of-magnitude improvement in immediate $(t+1)$ adaptation. Collectively, these results demonstrate TAG-K’s superior “step-one corrections,” underscoring its unique effectiveness in scenarios where rapid and robust adaptation is essential for stable closed-loop operation.

4) *Kaczmarz Family Ablation:* To better understand the impact of greedy selection and tail averaging, we ablate these components by comparing TAG-K to three Kaczmarz

variants in Figure 3b: 1) Randomized Kaczmarz (RK), which employs random row sampling weighted by row norm without greedy selection or tail averaging; 2) Tail-Averaged Randomized Kaczmarz (TARK), which introduces tail averaging, stabilizing estimates while still retaining row-norm-based random sampling; and 3) Greedy Randomized Kaczmarz (GRK) which uses greedy row selection for faster convergence but lacks the stability provided by averaging.

Across 2,000 trials, the results clearly show the value of combining these two changes: reduced estimation errors and iteration times, and improved tracking performance. TAG-K adapts faster than RK and TARK after sudden changes, driven by the selective power of greedy sampling. While GRK alone can sometimes converge marginally faster in the very first iterations, it exhibits higher long-term variability. By incorporating tail averaging, TAG-K reduces this variance and maintains stable estimates, preventing drift and compounding errors in closed-loop operation. This explains TAG-K’s closed-loop outperformance, achieving both rapid initial adaptation and strong long-term robustness.

B. Microcontroller Timing Benchmark

To further assess Q3, the computational performance of our approach, we include an additional benchmark on a Teensy 4.1 [55] microcontroller (MCU) development board.¹

¹The Teensy features an ARM Cortex-M7 microcontroller operating at 600 MHz, with 7.75 MB of flash memory, 512 kB of tightly coupled static RAM, and an additional 512 kB of tightly coupled dynamic RAM. All are orders of magnitude less compute than the CPU used in earlier benchmarks, and a common baseline for tiny robotic tasks [51].

TABLE III: **Benchmark Runtimes across Parameter Dimensions.** Average per-update runtime (μs) over 10 trials on a Teensy 4.1 microcontroller. Baseline abbreviations follow Table II. TAG-K achieves $4.7\times$ – $20.7\times$ speedups, with gains increasing at larger dimensions.

Parameters	RLS-low	RLS-high	KF-low	KF-high	TAG-K	Speedup vs TAG-K
40	2,292	2,289	2,279	2,279	480	4.7–4.8 \times
60	4,574	4,574	4,536	4,535	597	7.6–7.7 \times
80	8,023	8,025	7,928	7,929	716	11.1–11.2 \times
100	13,011	13,011	12,851	12,850	837	15.4–15.5 \times
120	19,685	19,683	19,451	19,452	950	20.5–20.7 \times
Average	9,500	9,500	9,429	9,429	716	13.2–13.3 \times

This experiment highlights how TAG-K’s computational efficiency enables deployment on resource-constrained edge robotic systems. In this experiment, we benchmark RLS, KF, and TAG-K across a variety of parameter window sizes ($n = 40, 60, 80, 100, 120$, corresponding to inertial parameter estimation problems for 4, 6, 8, 10, 12 DoF robots. We set the measurement window size to $m = 30$ and use the same forgetting factor, noise covariance and other optimized hyperparameters as in the prior setups.

Table III summarizes the average per-update runtimes (in μs) across 10 independent trials, with the final column reporting the relative speedup of TAG-K over the baselines. The results show that TAG-K consistently and substantially outperforms both RLS and KF in computational efficiency. Depending on the parameter dimension, TAG-K achieves speedups ranging from $4.8\times$ to $20.7\times$, with an average improvement of approximately $12\times$ across all tested scenarios. This demonstrates that TAG-K not only offers strong estimation accuracy, but also substantially reduces the computational burden. This is especially important for embedded systems, like our target MCU, where limited processing power, small caches, and constrained register files mean that even modest increases in problem size can lead to large increases in computation time. Most importantly, this level of efficiency is particularly valuable in real-time, resource-constrained robotic platforms, where saving even a few microseconds can have a meaningful impact on overall system performance.

VI. CONCLUSION AND FUTURE WORK

We presented TAG-K, a tail-averaged greedy Kaczmarz algorithm tailored for online inertial parameter estimation in robotics. By integrating greedy residual-driven row selection with tail averaging, TAG-K achieves both rapid adaptation to abrupt parameter changes and stable convergence under noise. Our experimental evaluations demonstrate that TAG-K consistently outperforms RLS, KF, and existing Kaczmarz variants, yielding up to $2\times$ improvement in end-to-end tracking accuracy, an order-of-magnitude reduction in early post-event estimation error, robust closed-loop performance across diverse trajectories and noise conditions, and up to a $20\times$ reduction in latency on MCU-class hardware.

While our evaluation considered diverse trajectories, noise models, and payload variations in high-fidelity simulation, the next step is validating TAG-K on physical quadrotors

to assess robustness under sensing imperfections, delays, and unmodeled disturbances. Beyond quadrotors, TAG-K naturally extends to domains like manipulation, where online inertial and contact estimation is key, and legged locomotion, where adaptive identification of payloads and terrain can improve stability. Finally, future work includes deeper theoretical analysis of convergence and closed-loop stability under biased noise or delays. Together, these directions position TAG-K as a practical framework to enable online parameter estimation for high-performance control across real-world robotic platforms.

REFERENCES

- [1] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, “Optimization-based control for dynamic legged robots,” *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.
- [2] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, “Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot,” *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.
- [3] S. Kuindersma, “Taskable agility: Making useful dynamic behavior easier to create,” 2023, Princeton University Seminar.
- [4] D. Kim, J. Di Carlo, B. Katz, G. Bledd, and S. Kim, “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” *arXiv preprint arXiv:1909.06586*, 2019.
- [5] Y.-H. Shin, S. Hong, S. Woo, J. Choe, H. Son, G. Kim, J.-H. Kim, K. Lee, J. Hwangbo, and H.-W. Park, “Design of kaist hound, a quadruped robot platform for fast and efficient locomotion with mixed-integer nonlinear optimization of a gear train,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6614–6620.
- [6] H. Li and P. M. Wensing, “Cafe-mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control,” *IEEE Transactions on Robotics*, vol. 41, pp. 837–856, 2025.
- [7] M. Krinner, A. Romero, L. Bauersfeld, M. Zeilinger, A. Carron, and D. Scaramuzza, “Mpcc++: Model predictive contouring control for time-optimal flight with safety constraints,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.17551>
- [8] B. Zhang, Z. Zhou, and R. Vasudevan, “Provably-safe, online system identification,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.21486>
- [9] D. Baek, B. Peng, S. Gupta, and J. Ramos, “Online learning-based inertial parameter identification of unknown object for model-based control of wheeled humanoids,” 2024. [Online]. Available: <https://arxiv.org/abs/2309.09810>
- [10] I. Lopez-Sanchez, F. Rossomando, R. Pérez-Alcocer, C. Soria, R. Carelli, and J. Moreno-Valenzuela, “Adaptive trajectory tracking control for quadrotors with disturbances by using generalized regression neural networks,” *Neurocomputing*, vol. 460, pp. 243–255, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231221010092>
- [11] I. H. Imran, K. Wood, and A. Montazeri, “Adaptive control of unmanned aerial vehicles with varying payload and full parametric uncertainties,” *Electronics*, vol. 13, no. 2, 2024. [Online]. Available: <https://www.mdpi.com/2079-9292/13/2/347>

- [12] G. Baraban, M. Shekells, S. Kim, and M. Kobilarov, "Adaptive parameter estimation for aerial manipulation," in *2020 American Control Conference (ACC)*, 07 2020, pp. 614–619.
- [13] R. L. Plackett, "Some theorems in least squares," *Biometrika*, vol. 37, no. 1/2, pp. 149–157, 1950. [Online]. Available: <http://www.jstor.org/stable/2332158>
- [14] A. H. Sayed, *Fundamentals Of Adaptive Filtering*. Wiley, 2003. [Online]. Available: <https://api.semanticscholar.org/CorpusID:60598442>
- [15] J. Jiang and Y. Zhang, "A revisit to block and recursive least squares for parameter estimation," *Computers & Electrical Engineering*, vol. 30, no. 5, pp. 403–416, 2004. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790604000217>
- [16] C. Paleologu, J. Benesty, and S. Ciochina, "A robust variable forgetting factor recursive least-squares algorithm for system identification," *IEEE Signal Processing Letters*, vol. 15, pp. 597–600, 2008.
- [17] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.
- [18] F. Gustafsson, *On-line Approaches*. John Wiley & Sons, Ltd, 2000, ch. 3, pp. 55–87. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/0470841613.ch3>
- [19] B. Plancher, C. D. Brumar, I. Brumar, L. Pentecost, S. Rama, and D. Brooks, "Application of approximate matrix multiplication to neural networks and distributed slam," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, USA, September 2019, pp. 1–7.
- [20] V. Sivtsov, A. Papanikolaou, I. Markovic, I. Petrovic, and F. Bonsignorio, "Physics-informed neural networks in robotics: A review," 01 2025. [Online]. Available: <https://dx.doi.org/10.2139/ssrn.5125543>
- [21] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, "Deep learning and system identification," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020, 21st IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896320317353>
- [22] H. Li, H. Chen, Y. Chai, H. Zhao, Y. Zhao, Y. Han, and J. Luo, "Pinn-based predictive control combined with unknown payload identification for robots with prismatic quasi-direct-drives," *IEEE Robotics and Automation Letters*, pp. 1–8, 2025.
- [23] S. Mamedov, A. R. Geist, R. Viljoen, S. Trimpe, and J. Swevers, "Learning deformable linear object dynamics from a single trajectory," 2024. [Online]. Available: <https://arxiv.org/abs/2407.03476>
- [24] K. Zhang, B. Li, K. Hauser, and Y. Li, "Particle-grid neural dynamics for learning deformable object models from rgb-d videos," 2025. [Online]. Available: <https://arxiv.org/abs/2506.15680>
- [25] M. Lutter, J. Silberbauer, J. Watson, and J. Peters, "Differentiable physics models for real-world offline model-based reinforcement learning," 2020. [Online]. Available: <https://arxiv.org/abs/2011.01734>
- [26] G. Qu, C. Yu, S. Low, and A. Wierman, "Combining model-based and model-free methods for nonlinear control: A provably convergent policy gradient approach," 2020. [Online]. Available: <https://arxiv.org/abs/2006.07476>
- [27] S. Karczmarz, "Angenaherte auflösung von systemen linearer gleichungen," *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.*, pp. 355–357, 1937.
- [28] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, pp. 262–278, 2009.
- [29] A. Zouzias and N. M. Freris, "Randomized extended Kaczmarz for solving least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 2, p. 773–793, Jan. 2013. [Online]. Available: <http://dx.doi.org/10.1137/120889897>
- [30] X. Chen, "The Kaczmarz algorithm, row action methods, and statistical learning algorithms," *Frames and harmonic analysis*, vol. 706, pp. 115–127, 2018.
- [31] Z.-Z. Bai and W.-T. Wu, "On greedy randomized Kaczmarz method for solving large sparse linear systems," *SIAM Journal on Scientific Computing*, vol. 40, no. 1, pp. A592–A606, 2018. [Online]. Available: <https://doi.org/10.1137/17M1137747>
- [32] A. Agrawal, P. Mayer, Z. Kingston, and G. Hollinger, "Constrained nonlinear Kaczmarz projection on intersections of manifolds for coordinated multi-robot mobile manipulation," 10 2024.
- [33] C. P. Neuman and P. K. Khosla, *Identification of Robot Dynamics: An Application of Recursive Estimation*. Boston, MA: Springer US, 1986, pp. 175–194. [Online]. Available: https://doi.org/10.1007/978-1-4757-1895-9_12
- [34] W. Khalil and F. Boyer, "An efficient calculation of computed torque control of flexible manipulators," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1, 1995, pp. 609–614 vol.1.
- [35] D. Kubus, T. Kroger, and F. M. Wahl, "On-line estimation of inertial parameters using a recursive total least-squares approach," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3845–3852.
- [36] S. Rhode and F. Gauterin, "Online estimation of vehicle driving resistance parameters with recursive least squares and recursive total least squares," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 269–276.
- [37] S.-H. Leung and C. So, "Gradient-based variable forgetting factor rls algorithm in time-varying environments," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3141–3150, 2005.
- [38] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.
- [39] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation," in *2017 IEEE Power & Energy Society General Meeting*, 2017, pp. 1–5.
- [40] G. Evensen, "The Ensemble Kalman Filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, Nov. 2003.
- [41] K. Tracy, "A square-root Kalman filter using only QR decompositions," 2022. [Online]. Available: <https://arxiv.org/abs/2208.06452>
- [42] R. Merwe and E. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 6, 02 2001, pp. 3461 – 3464 vol.6.
- [43] I. A. Ferreira, J. A. Acebrón, and J. Monteiro, "Survey of a class of iterative row-action methods: The Kaczmarz method," 2024. [Online]. Available: <https://arxiv.org/abs/2401.02842>
- [44] E. N. Epperly, G. Goldshlager, and R. J. Webber, "Randomized Kaczmarz with tail averaging," 2025. [Online]. Available: <https://arxiv.org/abs/2411.19877>
- [45] A. Sayed, *Kalman Filtering and RLS*. John Wiley & Sons, Ltd, 2008, ch. 31, pp. 501–514. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470374122.ch46>
- [46] D. Needell, "Randomized Kaczmarz solver for noisy linear systems," *BIT Numerical Mathematics*, vol. 50, no. 2, pp. 395–403, 2010.
- [47] Bitcraze, "Crazyflie 2.1," 2023. [Online]. Available: <https://www.bitcraze.io/products/crazyflie-2-1/>
- [48] J. Preiß, W. Hoenig, G. Sukhatme, and N. Ayanian, "CrazySwarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 05 2017, pp. 3299–3304.
- [49] L. Pichierrì, A. Testa, and G. Notarstefano, "Crazychoir: Flying swarms of crazyflie quadrotors in ros 2," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4713–4720, 2023.
- [50] A. Javeed, V. Jiménez, and J. Grönqvist, "Reinforcement learning-based control of crazyflie 2.x quadrotor," 06 2023.
- [51] K. Nguyen, S. Schoedel, A. Alavill, B. Plancher, and Z. Manchester, "Tinympc: Model-predictive control on resource-constrained microcontrollers," in *IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, May. 2024.
- [52] F. L. Lewis, D. Vrabie, and V. Syrmos, "Optimal Control," 1 2012.
- [53] S. M. M. N. Naik, R. M. O. Gemson, and M. R. Ananthasayanam, "Introduction to the kalman filter and tuning its statistics for near optimal estimates and cramer rao bound," 2015. [Online]. Available: <https://arxiv.org/abs/1503.04313>
- [54] A. Onose and B. Dumitrescu, "Cyclic adaptive matching pursuit," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3745–3748.
- [55] "Teensy@ 4.1." [Online]. Available: <https://www.pjrc.com/store/teensy41.html>