Estimating Time-Varying Pauli Noise with SCAPE: RMSE vs Window

I. Introduction

Time-varying noise breaks standard “more samples $\Rightarrow$ better estimates” intuition due to sampling delay. SCAPE enables simultaneous communication and parameter estimation using classical repetition codes plus receiver-side post-processing and lightweight feedback [1]. We reproduce core mechanics and analyze RMSE versus sampling window N.

II. Model and Estimation

Pauli channel. For $\rho \in \mathbb{C}^{2\times 2}$,

$$\mathcal{N}_t(\rho) = \sum_{i\in\{I,X,Y,Z\}} p_i(t)\,\sigma_i\,\rho\,\sigma_i, \tag{1}$$

with $\boldsymbol{p}(t) = [p_I, p_X, p_Y, p_Z]^\top$, $p_i \geq 0$, $\sum_i p_i = 1$. We generate $\boldsymbol{p}(t)$ via a correlation-matrix eigenvalue path parameterized by $\gamma(t) = b + a\sin(2\pi f t + \phi)$ (details in code).

EFPE inversion. Prepare and measure in Pauli bases to get crossover rates $\hat{\boldsymbol{\epsilon}} = [\hat{\epsilon}_X, \hat{\epsilon}_Y, \hat{\epsilon}_Z, 1]^\top$, then

$$\hat{\boldsymbol{p}} = A^{-1}\hat{\boldsymbol{\epsilon}}, \quad A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{2}$$

III. SCAPE Protocol Steps

Let $k \geq 3$ be the repetition factor and N the total channel uses in an estimation step.

1) **Split message:** Divide m into (m_X, m_Y, m_Z) of equal length $N/(3k)$.
2) **Encode:** For $j \in \{X, Y, Z\}$, form $E_k(m_j)$ by k-repetition.
3) **Modulate to qubits:** Map bits to $\{|0_j\rangle, |1_j\rangle\}$ and send over the channel for $N/3$ uses.
4) **Measure:** Receiver measures in basis j to obtain $\hat{E}_k(m_j)$.
5) **Decode:** Majority vote $\Rightarrow \hat{m}_j$. Re-encode $\check{E}_k(m_j) = E_k(\hat{m}_j)$.
6) **Error counting:** Estimate crossover per basis

$$\hat{\epsilon}_j = \frac{3}{N}\sum_{\ell=1}^{N/3} \mathbf{1}\{\hat{E}_k(m_j)[\ell] \neq \check{E}_k(m_j)[\ell]\}. \tag{3}$$

7) **Pauli estimate:** Compute $\hat{\boldsymbol{p}}$ via EFPE inversion above.
8) **Feedback (rate control):** Use the CRI (Sec. IV) on each received codeword to emit ternary feedback $b_j \in \{-1, 0, +1\}$ (increase, keep, decrease code rate).

IV. Codeword Reliability Index (CRI)

For a received k-repetition codeword with N_0 zeros and N_1 ones, define

$$C = \big|N_0 - N_1\big| - \frac{k-1}{2}. \tag{4}$$

Interpretation: C is the exact margin of flips that can occur without changing the majority decision. Larger C implies lower decoding risk.

A. Adaptive rate rule

Pick thresholds $s_{\min} < s_{\max}$. Over one estimation step, track the minimum CRI seen per basis. For basis j:

$$b_j = \begin{cases} +1, & \min C_j < s_{\min} \quad (\text{increase } k_j) \\ 0, & \text{otherwise if not enough evidence} \\ -1, & \frac{|\{C_j : C_j > s_{\max}\}|}{|C_j|} \geq 0.9 \quad (\text{decrease } k_j) \end{cases}$$

This adjusts k_X, k_Y, k_Z independently.

V. RMSE vs Window

We compute, for window size N,

$$\mathrm{RMSE}(N) = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \big\|\boldsymbol{p}(t_m) - \hat{\boldsymbol{p}}(t_m)\big\|_2^2}, \tag{5}$$

with timestamps aligned to the end of each window. *Observation:* RMSE is typically convex in N. Small N yields high statistical variance; large N suffers sampling delay as $\boldsymbol{p}(t)$ drifts within the window. The minimizer N^* depends on drift frequency f and SNR.

VI. Results

The SCAPE protocol was implemented both using NETSQUID package and using numpy. The version based on NETSQUID had long execution time so i decided to use pure numpy version for gathering statistics and creating figures. The obtained graphs on estimating time-varying Pauli parameters closely match the original figures. Estimation with N = 10^3 shows the existence of sampling decay, fig. 1.

According to the article, the N=350 and N=500 values show much better estimation of true parameters. It is corfirmed by the figures 2 and 3.

The norm between true parameters and estimated by SCAPE parameters is a convex function of N (window sample size). The small values of N suffer from statistical error. With too few samples per window, the estimates of error rates fluctuate strongly around true values. High values of N suffer

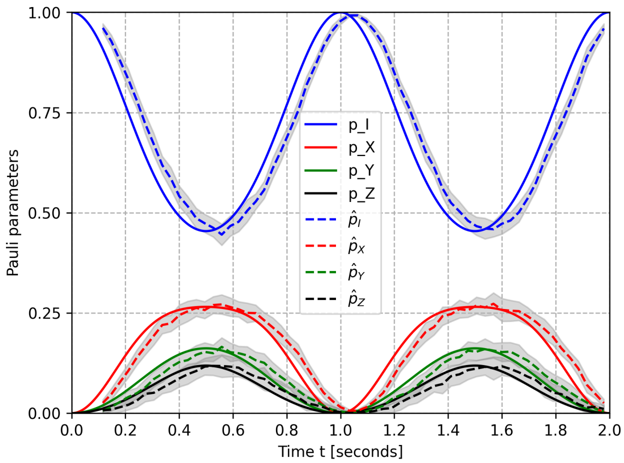

Fig. 1. Estimation of time-varying Pauli parameters, $N = 10^3$.

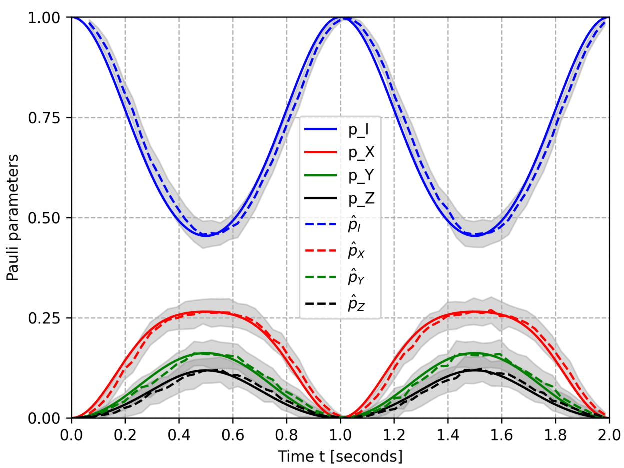

Fig. 2. Estimation of time-varying Pauli parameters, $N = 500$.

from the sampling decay, when the underlying distribution of parameters change significantly by the time the N samples are collected. This results in a possibility of chosing optimal value for N.

The value of f determine how fast the underlying parameters change. It has no effect for small values of N. The effect of f becomes noticable at large values of N. Exact ranges as in article for f and N were used.

VII. Takeaways

- The RMSE–N convexity reflects variance vs delay. There is a possibility of empirically choosing the value of N that minizes the distance between true Pauli parameters and estimated Pauli parameters.

VIII. Conclusion

We detailed SCAPE steps, derived and used CRI for rate adaptation, and measured the RMSE–window tradeoff on time-varying Pauli channels.

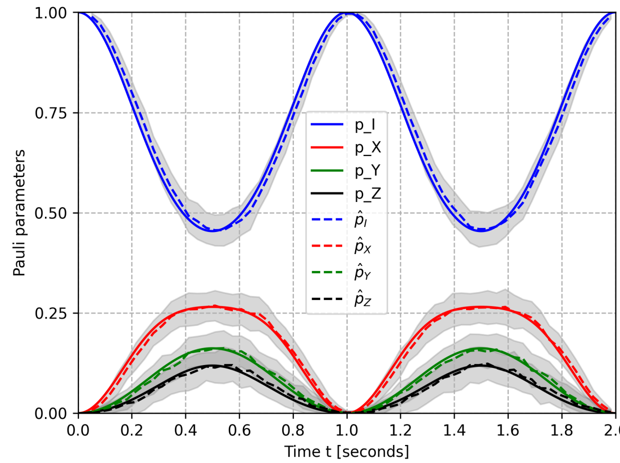

Fig. 3. Estimation of time-varying Pauli parameters, $N = 350$.

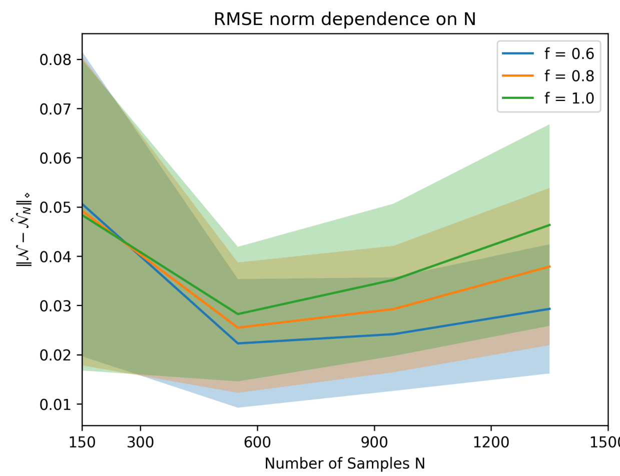

Fig. 4. RMSE versus window size N. Convex trend with optimal N^*.

References

[1] J. Ur Rehman, H. Al-Hraishawi, T. Q. Duong, S. Chatzinotas, and H. Shin, "On estimating time-varying pauli noise," *IEEE Transactions on Communications*, vol. 72, no. 4, pp. 2079–2089, 2024.