

1 Design of a Nonlinear / Non-Gaussian SSM

1.1 Stochastic Volatility State-Space Model

To benchmark nonlinear filtering methods, we adopt the stochastic volatility (SV) model used in Example 4 of Doucet and Johansen. The SV model captures time-varying latent volatility and induces a nonlinear, non-Gaussian filtering posterior. The latent state $x_t \in \mathbb{R}$ represents the log-volatility and evolves as an AR(1) Gaussian process:

$$x_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1 - \alpha^2}\right), \quad (1)$$

$$x_t = \alpha x_{t-1} + \sigma v_t, v_t \sim \mathcal{N}(0, 1), \quad (2)$$

where $|\alpha| < 1$ controls persistence and σ sets the process noise scale.

Observations $y_t \in \mathbb{R}$ are generated via a multiplicative noise model:

$$y_t = \beta \exp(x_t/2) w_t, \quad w_t \sim \mathcal{N}(0, 1), \quad (3)$$

so that the conditional likelihood is

$$p(y_t | x_t) = \mathcal{N}(0, \beta^2 e^{x_t}). \quad (4)$$

1.2 Nonlinearity and Non-Gaussianity

Although the transition (2) is linear-Gaussian, the observation model (3) is nonlinear in x_t through the exponential term.

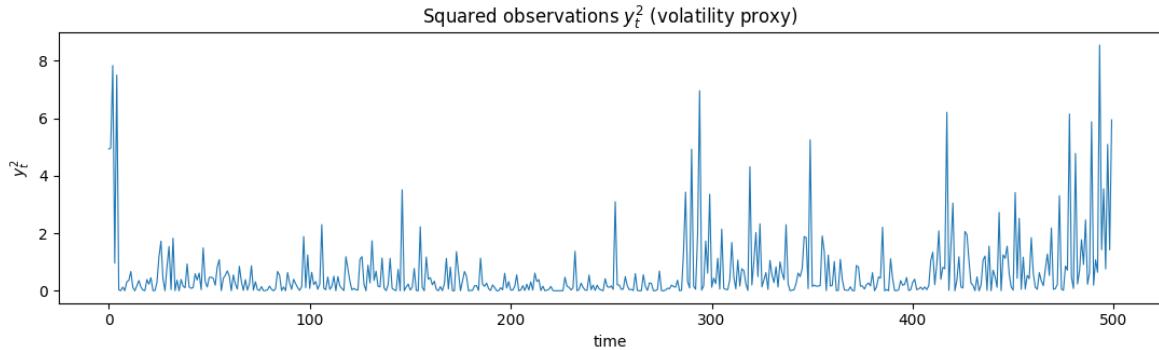


Figure 1: Squared observations y_t^2 , a proxy for realized volatility.

Consequently, the filtering posterior $p(x_t | y_{1:t})$ is not Gaussian in general: large observation shocks produce skewed and heavy-tailed posteriors that cannot be represented exactly by a single Gaussian. This makes the SV model a standard stress test for EKF/UKF approximations and particle filters.

1.3 Synthetic Data Generation

We generate a synthetic sequence $\{x_t, y_t\}_{t=1}^T$ by sampling from the prior and iterating (2)–(3). Unless otherwise stated, we use $(\alpha, \sigma, \beta) = (0.98, 0.15, 0.65)$ and $T = 500$. The script `q2_problem1_sv_ssm.py` saves the latent trajectory and observations to `data_sv.npz` for later filter comparisons.

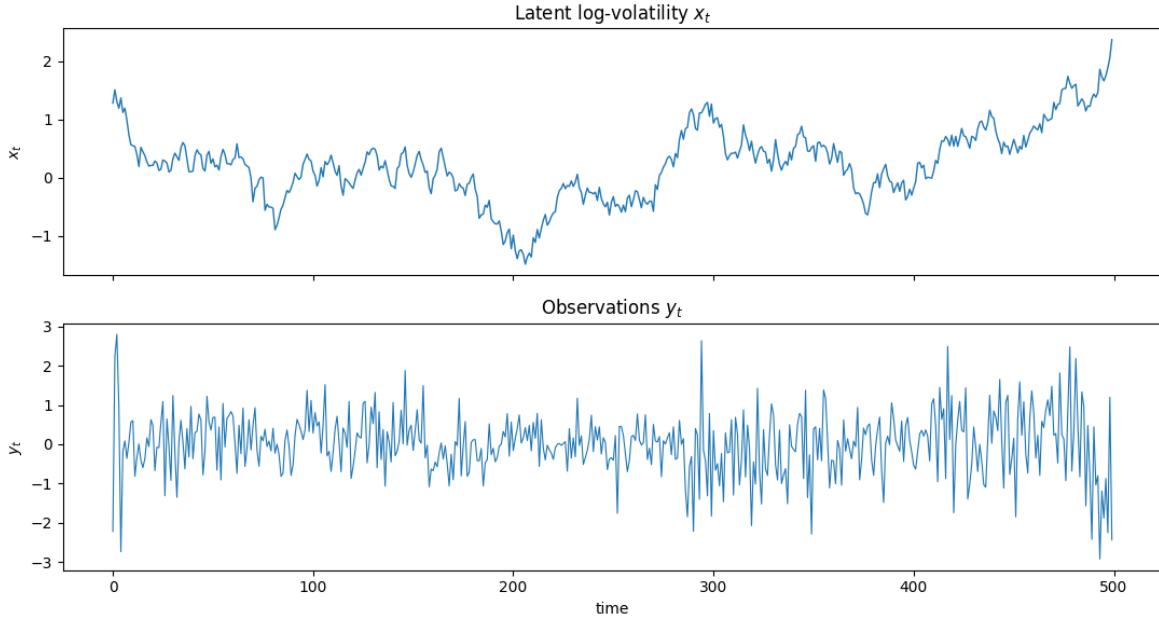


Figure 2: Synthetic stochastic volatility data: latent log-volatility x_t (top) and observations y_t (bottom).

Figure 2 shows one realization of the latent log-volatility x_t and observations y_t . Squared observations y_t^2 (Figure 1) illustrate spiky volatility episodes, where the posterior is expected to deviate most strongly from Gaussianity.

2 EKF and UKF Filtering

2.1 EKF Approximation via Log-Squared Observations

The SV observation model is multiplicative and zero-mean:

$$y_t = \beta \exp(x_t/2) w_t, \quad w_t \sim \mathcal{N}(0, 1).$$

This form is not additive-Gaussian, so a direct EKF linearization on y_t is not well-posed. Following standard SV filtering practice, we define a pseudo-measurement

$$z_t = \log(y_t^2 + \epsilon),$$

yielding

$$z_t = \log(\beta^2) + x_t + \underbrace{\log(w_t^2)}_{\eta_t}.$$

The term η_t follows a $\log\chi^2(1)$ distribution, which is non-Gaussian. To obtain a tractable EKF recursion, we approximate it by a Gaussian

$$\eta_t \approx \mathcal{N}(\mu_\eta, \sigma_\eta^2),$$

with $\mu_\eta \approx -1.27036$ and $\sigma_\eta^2 \approx 4.93480$. Thus the approximate measurement model becomes linear-Gaussian:

$$z_t \approx h_z(x_t) + e_t, \quad h_z(x) = x + c, \quad e_t \sim \mathcal{N}(0, R_z),$$

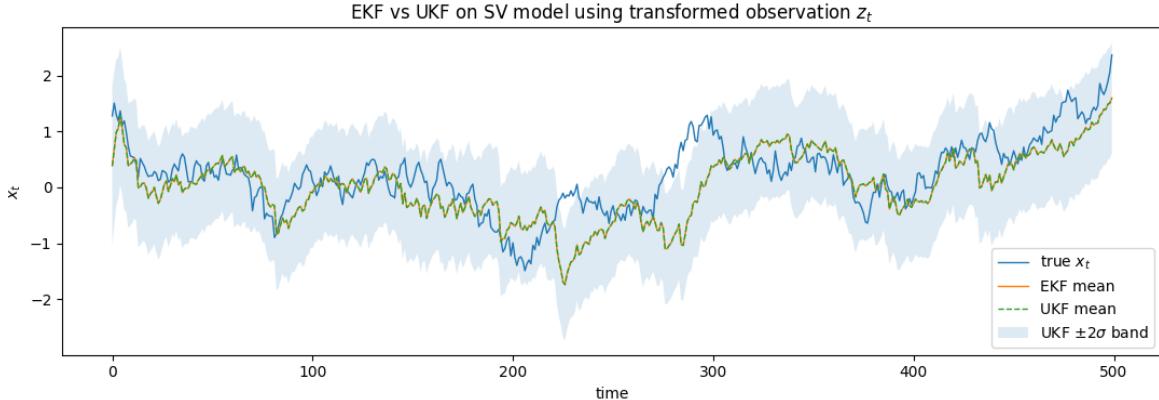


Figure 3: EKF and UKF filtered means on the SV model using transformed observations z_t . The two estimates nearly coincide because h_z is linear under the Gaussian approximation.

where $c = \log(\beta^2) + \mu_\eta$ and $R_z = \sigma_\eta^2$.

Under this approximation the EKF reduces to a Kalman filter on z_t . Because the approximation ignores the heavy-tailed structure of η_t , EKF errors increase during volatility spikes, when the true posterior is strongly skewed.

2.2 UKF on the Transformed Measurement

We also implement a scaled unscented Kalman filter (UKF), which propagates sigma points through the nonlinear measurement mapping to approximate posterior moments without explicit Jacobians. For the Gaussian prior

$$x_t | y_{1:t-1} \approx \mathcal{N}(m_{t|t-1}, P_{t|t-1}),$$

the UKF uses $2d_x + 1$ sigma points (here $d_x = 1$) and corresponding weights, following the standard scaled-UT construction. The update is computed from the predicted measurement mean, covariance, and state–measurement cross-covariance.

In the SV case with z_t , the mapping $h_z(x) = x + c$ is linear, so the UT is exact and UKF closely matches EKF numerically. Figure 3 shows both filtered means overlaid on the ground-truth latent state.

2.3 Sigma-Point Failure Under Strong Nonlinearity / Non-Additive Noise

To highlight UKF limitations, we apply a *naive* UKF directly to the raw observations y_t by incorrectly assuming an additive-Gaussian measurement model. However, in the true SV likelihood,

$$p(y_t | x_t) = \mathcal{N}(0, \beta^2 e^{x_t}),$$

the conditional mean is zero for all x_t . Consequently, sigma points propagated through the conditional mean mapping produce nearly identical predicted measurements, leading to a near-zero state–measurement cross-covariance and hence an almost zero Kalman gain. The filter performs essentially no measurement update and tracks only the prior.

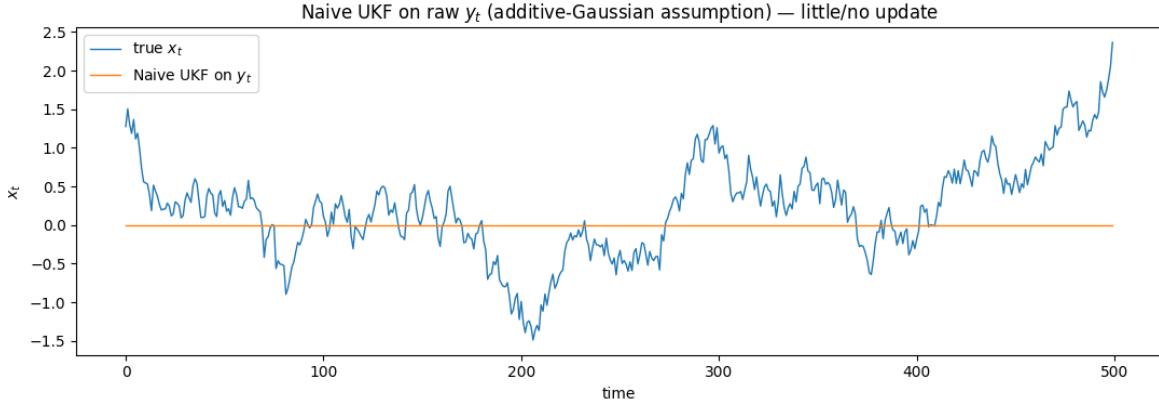


Figure 4: Naive UKF on raw y_t assuming additive Gaussian noise. Because y_t has zero conditional mean, the UKF gain collapses and the filter performs little or no update.

Table 1: Filtering accuracy for EKF/UKF on the SV benchmark.

Method	RMSE	MAE	Avg. posterior var.
EKF on z_t	0.5090	0.3792	0.2484
UKF on z_t	0.5090	0.3795	0.2486
Naive UKF on y_t	0.6569	0.5112	0.5682

2.4 Quantitative Summary

Table 1 shows that EKF and UKF applied to the transformed measurements $z_t = \log(y_t^2 + \epsilon)$ yield essentially identical accuracy (RMSE ≈ 0.509 , MAE ≈ 0.379) and posterior variance ($\bar{P} \approx 0.248$). This is expected because under the Gaussian approximation the transformed measurement model is linear, so the unscented transform provides no advantage over first-order linearization. In contrast, the naive UKF applied directly to raw observations y_t performs substantially worse (RMSE 0.657, MAE 0.511) and maintains a much larger average posterior variance (0.568), reflecting the collapse of the Kalman gain when the informative structure of $p(y_t | x_t)$ resides in the variance rather than the mean.

3 Particle Filter on the SV Model

3.1 Bootstrap Particle Filter Setup

We now apply a bootstrap particle filter (PF) to the stochastic volatility model using the *correct* likelihood $p(y_t | x_t)$ from Eq. (4), without the Gaussian log-squared approximation used in the EKF/UKF section. The proposal is the transition prior,

$$x_t^{(i)} \sim p(x_t | x_{t-1}^{(a_i)}), \quad w_t^{(i)} \propto w_{t-1}^{(a_i)} p(y_t | x_t^{(i)}),$$

where (a_i) are ancestor indices drawn by multinomial resampling whenever the effective sample size (ESS) falls below $N_{\text{th}} = N/2$. We use $N = 5000$ particles and $T = 500$ time steps.

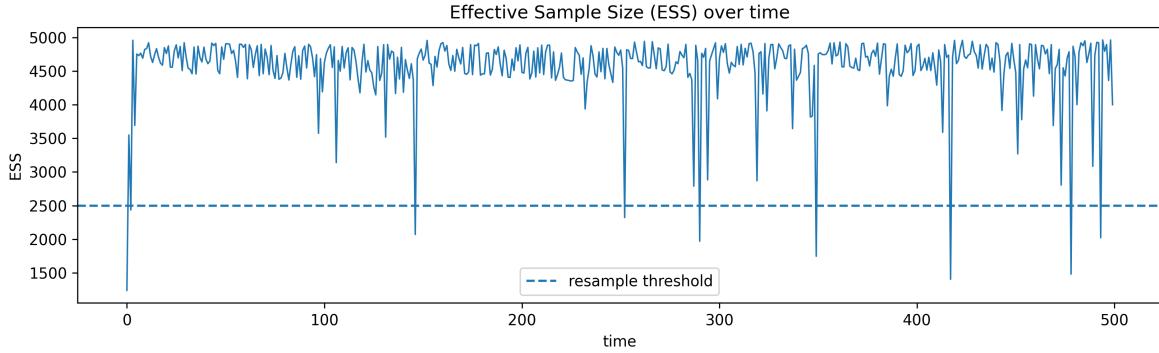


Figure 5: Effective sample size (ESS) over time for the bootstrap PF with $N = 5000$ particles. The dashed line shows the resampling threshold $N_{\text{th}} = N/2 = 2500$. ESS remains high except during volatility spikes, where weight degeneracy increases and resampling is triggered.

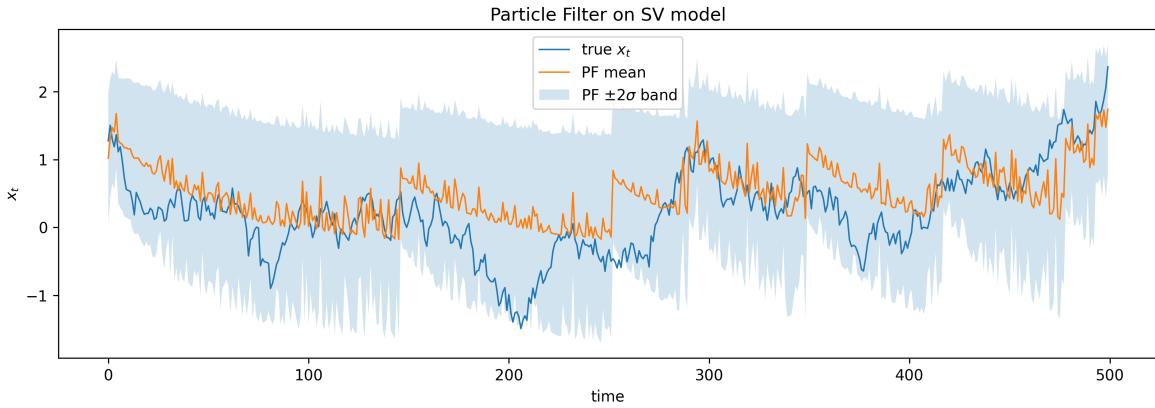


Figure 6: Bootstrap PF on the SV model. The PF posterior mean (orange) tracks the true x_t (blue). The shaded region indicates the empirical $\pm 2\sigma$ interval, which grows substantially during volatility spikes.

3.2 Effective Sample Size and Resampling Behaviour

Figure 5 shows the ESS trajectory over time. For most time steps the ESS remains in the range 4000–5000, indicating that importance weights are well balanced. Sharp ESS drops occur during volatility spikes, when y_t is highly unlikely under the prior predictive distribution, concentrating weight on a small subset of particles and triggering resampling.

3.3 Filtering Accuracy and Uncertainty

Figure 6 compares the PF posterior mean and its $\pm 2\sigma$ band with the true latent log-volatility. The PF tracks low-frequency structure well and expands its uncertainty band during spikes, reflecting strongly non-Gaussian posteriors.

Quantitatively, the PF achieves RMSE 0.5595 and MAE 0.4452 on x_t , with average posterior variance 0.4080. The runtime is 0.672 s total (1.34 ms/step), comparable to EKF and far cheaper than UKF, while operating on the exact likelihood.

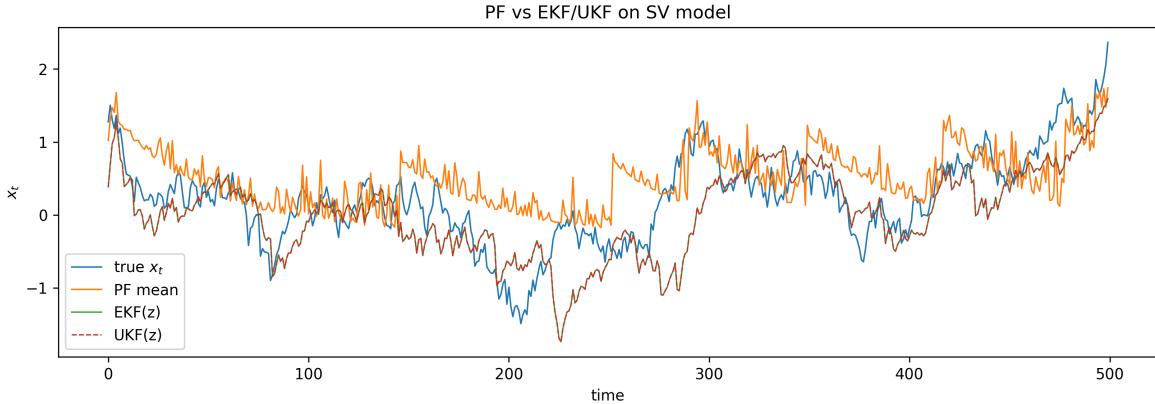


Figure 7: PF versus EKF/UKF on the SV benchmark. PF (orange) reacts more sharply to volatility shocks than EKF(z) / UKF(z) (green/red), reflecting the exact likelihood structure.

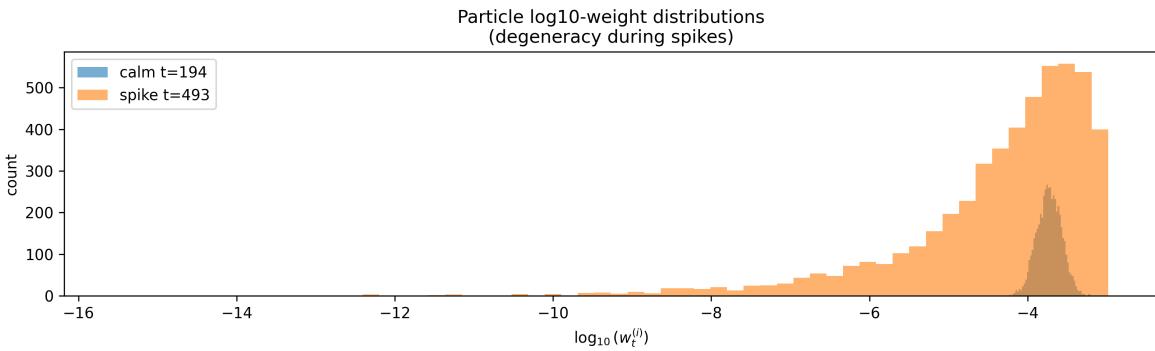


Figure 8: Particle log-weight distributions at calm ($t = 194$) and spike ($t = 493$) times. Spikes induce extreme weight imbalance, revealing degeneracy that is only partially controlled by ESS-based resampling.

3.4 Comparison to EKF/UKF

Figure 7 contrasts PF with EKF(z) and UKF(z). EKF and UKF nearly coincide due to the linearized z_t approximation (Section 2), whereas PF responds more strongly to extreme observations. Although PF has slightly larger RMSE/MAE than EKF/UKF (compare with Table 1), it provides posterior statistics consistent with the true non-Gaussian SV likelihood.

3.5 Weight Degeneracy During Volatility Spikes

Figure 8 shows histograms of particle log-weights at a calm time ($t = 194$) and a spike ($t = 493$). At $t = 194$ the weights concentrate tightly, implying near-uniform importance. At $t = 493$ the distribution spans many orders of magnitude, showing that only a small fraction of particles remain relevant. This illustrates classical PF degeneracy, motivating more advanced particle transport ideas such as particle flow methods.

Table 2: Comparison of filtering methods on the SV model. Cov@ 2σ is the empirical coverage probability of the $\pm 2\sigma$ posterior band.

Method	RMSE	MAE	AvgVar	Cov@ 2σ	Runtime (s)	PeakCPU (MB)
EKF(z)	0.5090	0.3792	0.2484	0.930	0.0251	0.01
UKF(z)	0.5090	0.3792	0.2484	0.930	0.1735	0.01
Naive UKF(y)	2.0363	1.9108	0.3840	0.118	0.1649	0.01
PF ($N = 5000$)	0.5595	0.4452	0.4080	0.974	1.2245	0.32

4 Comparison and Discussion

4.1 Overall Method Comparison

To summarize the behavior of Gaussian filters and particle filtering on the SV benchmark, Table 2 reports accuracy, uncertainty calibration, runtime, and peak CPU memory. EKF(z) and UKF(z) are applied to the transformed pseudo-observations $z_t = \log(y_t^2 + \epsilon)$ under the Gaussian approximation; Naive UKF(y) is applied directly to raw observations y_t assuming incorrect additive noise; PF uses the exact likelihood on y_t with $N = 5000$ particles.

4.2 Interpretation of Accuracy and Calibration

EKF(z) and UKF(z). EKF and UKF achieve the lowest point-estimate errors (RMSE ≈ 0.509 , MAE ≈ 0.379), because the log-square transformation plus Gaussian approximation converts the measurement model into an affine linear-Gaussian form. In this regime the UKF offers no advantage over EKF, so both filters collapse to the same Kalman recursion (Section 2). The coverage Cov@ $2\sigma = 0.930$ is close to the nominal Gaussian value (0.954 for a perfectly calibrated normal posterior), implying that the approximate posterior variance $\bar{P} \approx 0.248$ is slightly *under-dispersed* during spike episodes, where the true posterior is skewed and heavy-tailed.

Naive UKF(y). Applying UKF directly to raw y_t with an additive-Gaussian assumption fails catastrophically. The RMSE (2.0363) and MAE (1.9108) are nearly four times larger than EKF/UKF(z), and Cov@ $2\sigma = 0.118$ indicates that the uncertainty band almost never contains the true state. This matches the theoretical failure mechanism: under the true SV likelihood $p(y_t|x_t) = \mathcal{N}(0, \beta^2 e^{x_t})$ the conditional mean is identically zero, so sigma points produce nearly identical predicted measurements and the Kalman gain collapses.

Particle filter. The PF yields slightly higher point-estimate errors (RMSE 0.5595, MAE 0.4452) than EKF/UKF(z), but importantly provides much better uncertainty calibration: Cov@ $2\sigma = 0.974$ exceeds the Gaussian filters and is closer to the nominal 95% coverage target. The larger posterior variance $\bar{P} \approx 0.408$ reflects the fact that PF represents the true non-Gaussian posterior rather than forcing a single Gaussian approximation. Thus PF trades a small loss in RMSE for a more faithful posterior and better spike-aware uncertainty.

4.3 Computational Trade-offs

EKF(z) is fastest (0.025 s total), while UKF(z) is $\sim 7\times$ slower (0.173 s) due to sigma-point propagation, without improving accuracy. PF is the slowest (1.225 s total) because it maintains and reweights 5000 particles at each step. Peak CPU memory follows the same pattern: EKF/UKF

store only a few scalars per step (0.01 MB), whereas PF stores the entire particle cloud and weights (0.32 MB). Although this memory cost is modest in the 1D SV model, it scales linearly with particle count and state dimension.