

Monthly Report

Jiajia HUANG

December 31, 2025

Contents

1	Linear-Gaussian SSM with Kalman Filter	1
1.1	Linear-Gaussian State Space Model	1
1.2	Kalman Filter	2
1.3	Results and Discussion	3
1.3.1	Compare filtered means/covariances to the Kalman recursion	4
1.3.2	Joseph stabilized covariance and Conditioning number	6
2	Nonlinear/Non-Gaussian SSM with EKF/UKF and Particle Filter	8
2.1	Model Description: Stochastic Volatility	8
2.2	Extend Kalman filter (EKF) and linearization accuracy limits	8
2.2.1	Results and discussion	9
2.3	Unscented Kalman Filter(UKF) and sigma point failures	11
A	Appendix	11
A.1	Derivation of Covariance Update Equation for Kalman Filter	11
A.2	Optimal Kalman Gain	11

1 Linear-Gaussian SSM with Kalman Filter

1.1 Linear-Gaussian State Space Model

We first set up a context for the problem. Assume a system is an object moving with constant velocity. The state of a 2D system is $X_n = [x_n, y_n, v_{xn}, v_{yn}]^T$, and the observation is a 2D vector $Y_n = [x_n^{obs}, y_n^{obs}]^T$. Assume we have a noise V_n that acts on velocity v_{xn}, v_{yn} and W_n that acts on measurement \hat{x}_n, \hat{y}_n respectively. This means that:

$$x_n = x_{n-1} + v_{xn} * \Delta t \quad (1)$$

$$y_n = y_{n-1} + v_{yn} * \Delta t \quad (2)$$

$$v_{xn} = v_{xn-1} + V_n \quad (3)$$

$$v_{yn} = v_{yn-1} + W_n \quad (4)$$

$$x_n^{obs} = x_n + W_n \quad (5)$$

$$y_n^{obs} = y_n + W_n \quad (6)$$

This setting lives up to a linear Gaussian state space model:

$$X_n = A * X_{n-1} + B * V_n \quad (7)$$

$$Y_n = C * X_n + D * W_n \quad (8)$$

Where $V_n \sim N(0, I)$ and $W_n \sim N(0, I)$. The state transition matrix is $A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$,

Δt is the time step. The process noise matrix is $B = b * \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, b is a scaling factor. The observation matrix is $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. The observation noise matrix is $D = d * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, d is a scaling factor. The initial state is $X_0 \sim N(0, \Sigma)$:

1.2 Kalman Filter

Assume that we now only have the observation Y_n data, and we know A, B, C, D, Σ_0 , we can use Kalman Filter to estimate the hidden state X_n . In the following, we will use the notation $\hat{X}_{n|n}$ to denote the filtered mean and $\hat{P}_{n|n}$ to denote the filtered covariance.

Kalman Filter can be seen as the optimal weighting of two Gaussian processes, predition process and observation process. It first predicts the prior from the previous state variable (=prediction) and calculates the likelihood when the observation value is given, then finds the posterior pdf according to the Bayesian rule (=correction). In one iteration of the Kalman recursion, the filtered means/covariances are updated by the following equation:

$$\begin{array}{c} \hat{X}_{n-1|n-1} \xrightarrow{\text{Prediction}} \hat{X}_{n|n-1} \xrightarrow{\text{Correction}} \hat{X}_{n|n} \\ \hat{P}_{n-1|n-1} \xrightarrow{\text{Prediction}} \hat{P}_{n|n-1} \xrightarrow{\text{Correction}} \hat{P}_{n|n} \end{array} \quad (9)$$

In prediction step, the predicted mean and covariance are updated by the following equation:

$$\hat{X}_{n|n-1} = A * \hat{X}_{n-1|n-1} \quad (10)$$

$$\hat{P}_{n|n-1} = A * \hat{P}_{n-1|n-1} * A^T + B * B^T \quad (11)$$

In correction step, we use the information from the observation Y_t to update the filtered mean and covariance by the following equation:

$$\hat{X}_{n|n} = \hat{X}_{n|n-1} + K_n * (Y_n - C * \hat{X}_{n|n-1}) \quad (12)$$

Here the Kalman gain K_n is introduced. Then we can have (see Appendix for derivation):

$$\hat{P}_{n|n} = (I - K_n * C) * \hat{P}_{n|n-1} * (I - K_n * C)^T + K_n * D * D^T * K_n^T \quad (13)$$

In principle, (12) and (13) are valid with any form of K_n . However, in practice, we want to find the optimal K_n that minimizes the error between the filtered mean and the true mean. This optimal K_n is given by the following equation []

$$K_n = \hat{P}_{n|n-1} * C^T * (C * \hat{P}_{n|n-1} * C^T + D * D^T)^{-1} \quad (14)$$

Using (14), (13) can be rewritten as:

$$\hat{P}_{n|n} = (I - K_n * C) * \hat{P}_{n|n-1} \quad (15)$$

Eq. (15) is the standard covariance update equation, while Eq. (13) is the Joseph stabilized covariance update equation.

Known that the initial state is $X_0 \sim N(0, \Sigma)$. In calculation, we use $X_0 = [0, 0, 0, 0]^T$ and $P_0 = \Sigma$ to initialize the filter.

1.3 Results and Discussion

Figure 1 shows the Kalman Filter for Position X and Y. The blue shaded area is the 95% confidence interval, which is calculated as the $x_{pred} \pm 1.96 * \sqrt{P_{pred}}$. Within this confidence interval, the filtered trajectory(blue line) is close to the ground truth trajectory(black line).

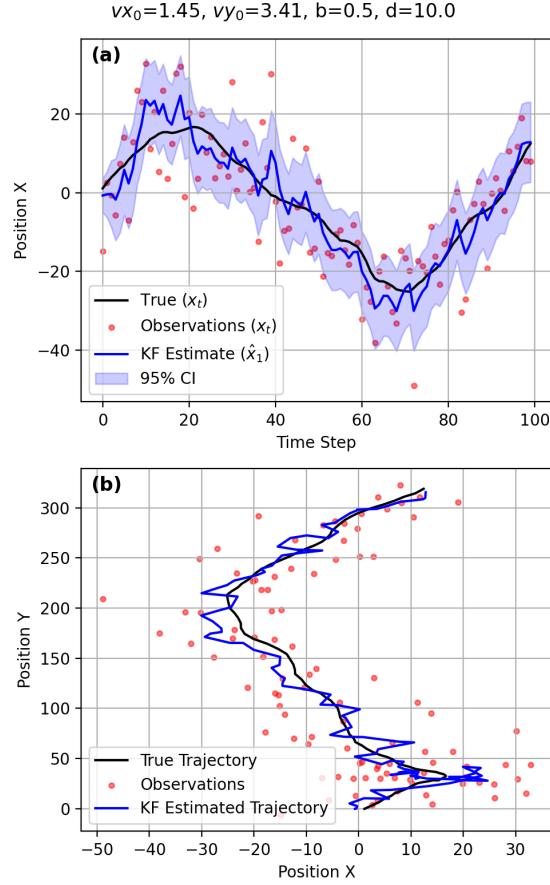


Figure 1: Kalman Filter for Position X and Y.

1.3.1 Compare filtered means/covariances to the Kalman recursion

This question asks us to compare the filtered means/covariances to the Kalman recursion. The filtered means/covariances are the estimated state/covariance of the system at the current time step, while the Kalman recursion is the process of updating the state/covariance of the system at the current time step based on the observation.

Fig. 2(a) shows the Kalman filtered means/covariances to the Kalman recursion. The True state (Black) follows a complex, wave-like path. The Observations (Red dots) are noisy. The KF Estimate (Blue line) successfully smooths the red dots. It doesn't chase every single noisy measurement, resulting in a path that is much closer to the Black line than the observation red dots. At the end of the simulation ($t > 175$), the trajectory drops rapidly. The filter tracks this sudden change in velocity quite well, showing that it is responsive and not "stuck" in its previous motion model.

In Fig. 2(b), the error of $X_{n|n}$ is always closer to the x-axis than $X_{n|n-1}$, and the error covariance (blue line) in 2(c) is always lower than the prior covariance (green line). This visually demonstrates

the core principle of the Kalman filter: after correction based on the observed data, the uncertainty of the estimation is reduced.

The red line in 2(c) is the K_n gain. One can always rewrite eq.(12) as:

$$\hat{X}_{n|n} = K_n * Y_n + (I - K_n * C) * \hat{X}_{n|n-1} \quad (16)$$

This means that K_n actually determines how much the filter believes in the observation Y_n (the red dots in (a)) and how much it believes in the predicted state $\hat{X}_{n|n-1}$ (the green line in (a)) when updating the state.

In the first few steps (from time 0 to 20), the K_n rises sharply and then drops, which is called the convergence process. The filter is rapidly adjusting to adapt to the environment. After that, K_n becomes a horizontal line, means the system has entered the steady state. As long as the noise characteristics of the system (process noise B and measurement noise D) remain unchanged, the gain will remain constant.

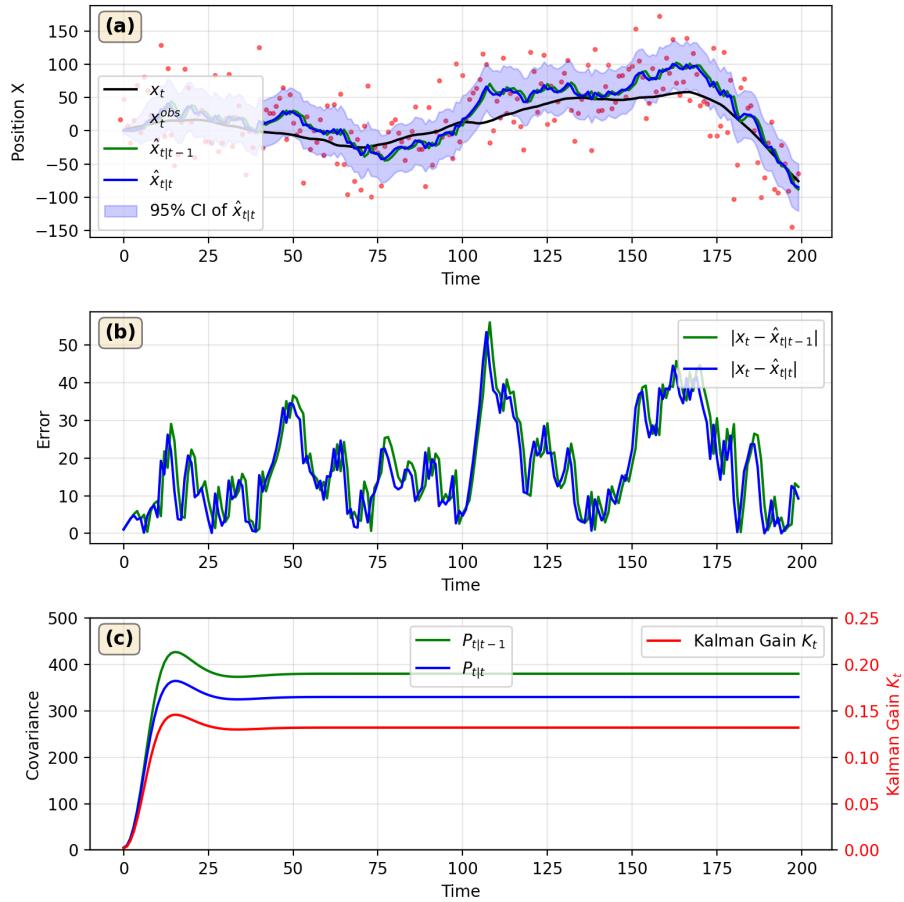


Figure 2: Kalman Filter for Position X and Y.

1.3.2 Joseph stabilized covariance and Conditioning number

Although mathematically equivalent, the standard form (15) and the Joseph stabilized form (13) for covariance update exhibit different numerical properties in finite-precision arithmetic. The standard update $P_{n|n} = (I - K_n C)P_{n|n-1}$ suffers from two main numerical issues:

1. **Loss of symmetry:** Due to floating-point roundoff errors, the computed covariance matrix may not be exactly symmetric, i.e., $P_{n|n} \neq P_{n|n}^T$. This asymmetry can accumulate over time and lead to numerical instabilities.
2. **Loss of positive definiteness:** The standard form does not guarantee that the updated covariance matrix remains positive definite. In extreme cases (e.g., very small measurement noise, large initial uncertainty, or long time horizons), numerical errors can cause the matrix to lose positive definiteness, resulting in negative eigenvalues.

The Joseph stabilized form explicitly maintains both symmetry and positive definiteness:

$$P_{n|n} = (I - K_n C)P_{n|n-1}(I - K_n C)^T + K_n R K_n^T \quad (17)$$

where $R = DD^T$ is the measurement noise covariance. The term $K_n R K_n^T$ is always positive semi-definite, ensuring that $P_{n|n}$ remains positive definite even in the presence of numerical errors. Additionally, the explicit symmetric form $(I - K_n C)P_{n|n-1}(I - K_n C)^T$ preserves symmetry by construction.

The **condition number** of a matrix measures how sensitive the solution is to perturbations in the input data and to roundoff errors made during the solution process. For a covariance matrix P , the condition number is defined as:

$$\kappa(P) = \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)} \quad (18)$$

where λ_{\max} and λ_{\min} are the maximum and minimum eigenvalues of P , respectively. A large condition number (e.g., $\kappa > 10^{12}$) indicates that the matrix is ill-conditioned, making numerical computations unreliable.

In the context of Kalman filtering, ill-conditioned covariance matrices can arise when:

- The measurement noise is very small relative to the process noise ($d \ll b$), leading to large Kalman gains
- The initial uncertainty is very large ($\sigma_0 \gg 1$)
- The filter runs for a long time, allowing numerical errors to accumulate

To demonstrate the numerical differences, we compare both methods using single-precision floating-point arithmetic (float32) with extreme parameters: $\sigma_0 = 1000$, $b = 0.5$, $d = 0.01$, and $T = 1000$ time steps. Figure 3 shows that:

- The standard method exhibits larger symmetry errors ($\|P - P^T\|_F$) compared to the Joseph method
- The standard method may lose positive definiteness (negative minimum eigenvalues), while the Joseph method maintains positive definiteness throughout

- Both methods show similar condition numbers, but the Joseph method provides better numerical stability

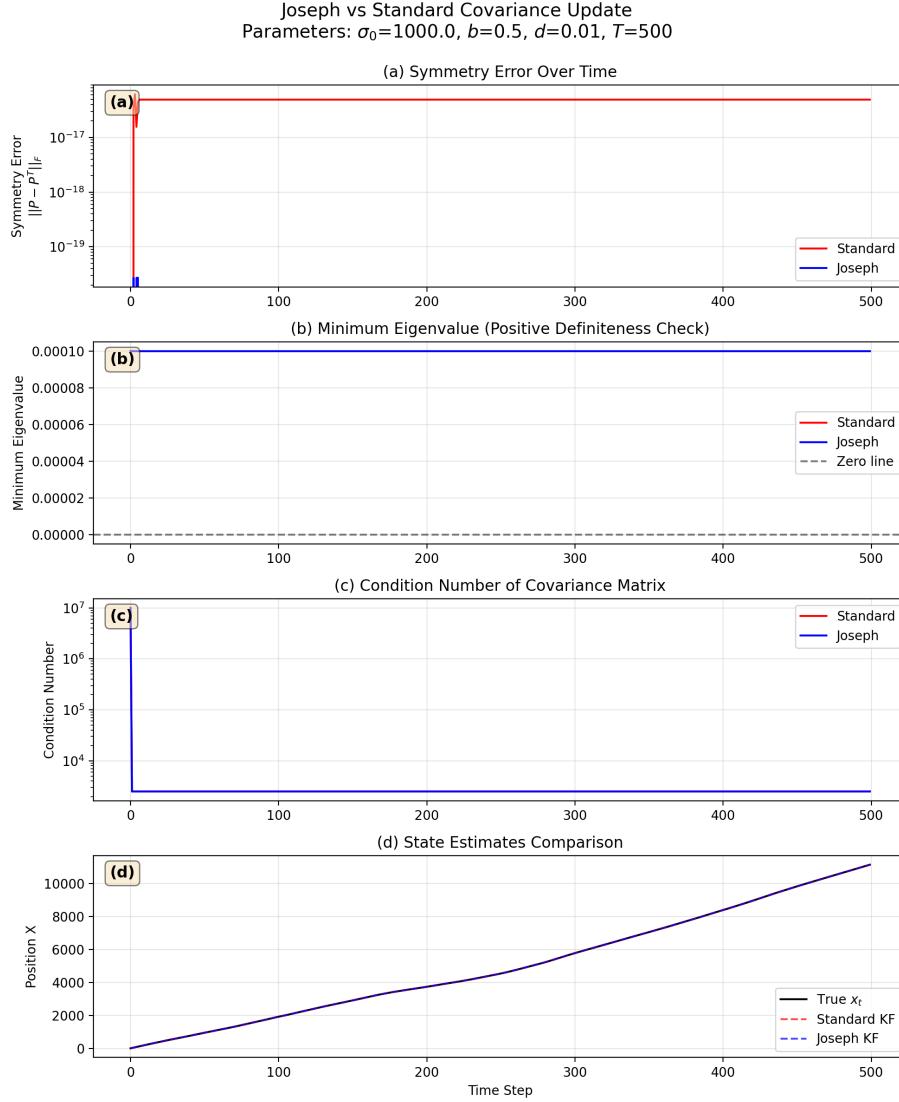


Figure 3: Comparison of Standard and Joseph stabilized covariance update methods. (a) Symmetry error over time, (b) Minimum eigenvalue (positive definiteness check), (c) Condition number, (d) State estimates comparison.

In practice, the Joseph stabilized form is recommended for applications requiring long-term stability, especially when using single-precision arithmetic or when the system parameters lead to ill-conditioned covariance matrices.

2 Nonlinear/Non-Gaussian SSM with EKF/UKF and Particle Filter

2.1 Model Description: Stochastic Volatility

We use the Stochastic Volatility(SV) Model in example 4 in [Doucet(09)]. This is similar to the classic Taylor (1986) Stochastic Volatility model [1] [2] [3], a discrete-time state space framework designed to capture the changing variance of financial returns. It assumes that observed asset returns (Y_n) are conditionally normal but driven by a latent, mean-reverting autoregressive (AR) process (X_n) representing log-volatility.

Here is the whole logic of the SV model in example 4 in [Doucet(09)]: Assume that in the n th moment, the volatility (the standard deviation of Y_n) is s_n and $Y_n \sim N(0, s_n^2)$. Therefore

$$Y_n = s_n * W_n \in R \quad (19)$$

Where $W_n \sim N(0, 1)$. Now define a new variable X_n as the corresponding log-variance of the rate of return:

$$X_n = \log\left(\frac{s_n}{\beta}\right)^2 \in R \quad (20)$$

With β being the Baseline Volatility Level. Assume that X_n follows the following AR(1) process:

$$X_n = \alpha * X_{n-1} + \sigma * V_n \quad (21)$$

where $V_n \sim N(0, 1)$ is the noise term. α is a scaling factor, determining how much current log-variance is affected by the previous log-variance. α is typically close to 1 but less than 1, indicating that the log-variance is highly persistent and the volatility process is stationary.

On the other hand, Eq.(19) can be formulated as:

$$Y_n = \beta * \exp(X_n/2) * W_n \quad (22)$$

Eq.(22) and eq.(21) are the complete form of the SV model in example 4 in [Doucet(09)]. $\alpha = 0.91$, $\sigma = 1.0$ and $\beta = 0.5$ in this example. The initial state is $X_0 \sim N(0, \frac{\sigma^2}{1-\alpha^2})$, making sure that we begin with a stationary state($Var(X_n) = Var(X_{n-1})$), the marginal distribution of all X_n is the same.

In application, we can use the SV model to estimate the volatility of the asset price. By accurately estimating the current hidden volatility state through filtering, we can leverage the persistence of volatility to forecast future volatility levels and, consequently, the distribution of future returns for risk assessment. Because the nonlinear/non-Gaussian nature of the SV model, we need to use nonlinear filters like EKF or UKF to estimate the state and covariance.

2.2 Extend Kalman filter (EKF) and linearization accuracy limits

Basically, what EKF [4] do is using the first-order Taylor approximation to approximate the nonlinear model to a linear model before applying the Kalman filter. Since the observation noise in eq.(22) is multiplicative, directly linearizing Y_n will directly lead to the disappearance of the observed information. We can use the log-transformation to linearize the observation equation:

$$\log(Y_n^2) = X_n + \log(\beta^2) + \log(W_n^2) \quad (23)$$

Taking $\log(Y_n^2)$ as the observation, linearly related to X_n . $\log(W_n^2)$ is the noise. We can now use the Kalman filter to estimate the state X_n under observation equation(23) and state equation(21). In the prediction step:

$$\hat{X}_{n|n-1} = \alpha * \hat{X}_{n-1|n-1} \quad (24)$$

$$\hat{P}_{n|n-1} = \alpha^2 * \hat{P}_{n-1|n-1} + \sigma^2 \quad (25)$$

In the correction step:

$$\hat{X}_{n|n} = \hat{X}_{n|n-1} + K_n * (\log(Y_n^2) - E[\log(Y_n^2)|\hat{X}_{n|n-1}]) \quad (26)$$

$$\hat{P}_{n|n} = (I - K_n) * \hat{P}_{n|n-1} \quad (27)$$

Here

$$\begin{aligned} E[\log(Y_n^2)|\hat{X}_{n|n-1}] &= \hat{X}_{n|n-1} + \log(\beta^2) + E[\log(W_n^2)] \\ &= \hat{X}_{n|n-1} + \log(\beta^2) - 2 \int_0^\infty \log(w^2) * \frac{1}{\sqrt{2\pi}} * e^{-\frac{w^2}{2}} dw \\ &= \hat{X}_{n|n-1} + \log(\beta^2) - 1.27 \end{aligned} \quad (28)$$

For the kalman gain K_n , we still use the same equation(eq.(14)) as in the Kalman filter. C is now the Jacobian matrix of $\log(Y_n^2)$ with respect to X_n , which is I. And $D * D^T$ now is the variance of $\log(W_n^2) = \pi^2/2$ [2] [5]:

$$K_n = \hat{P}_{n|n-1}/(\hat{P}_{n|n-1} + \pi^2/2) \quad (29)$$

But it should be noted that the kalman gain K_n now is not the minimum mean square error gain, because the observation noise $\log(W_n^2)$ is not Gaussian.

2.2.1 Results and discussion

Figure 4 shows the results of the EKF for the SV model. The upper plot shows that in SV model, the volatility of the asset price is bursts out in a concentrated manner, but it will eventually return to normal level. The lower plot shows the estimated volatility(red line) and the true volatility(blue line). We can see that when the true volatility(blue line) suddenly increase or decrease large, the estimated volatility(red line) will not be able to catch up.

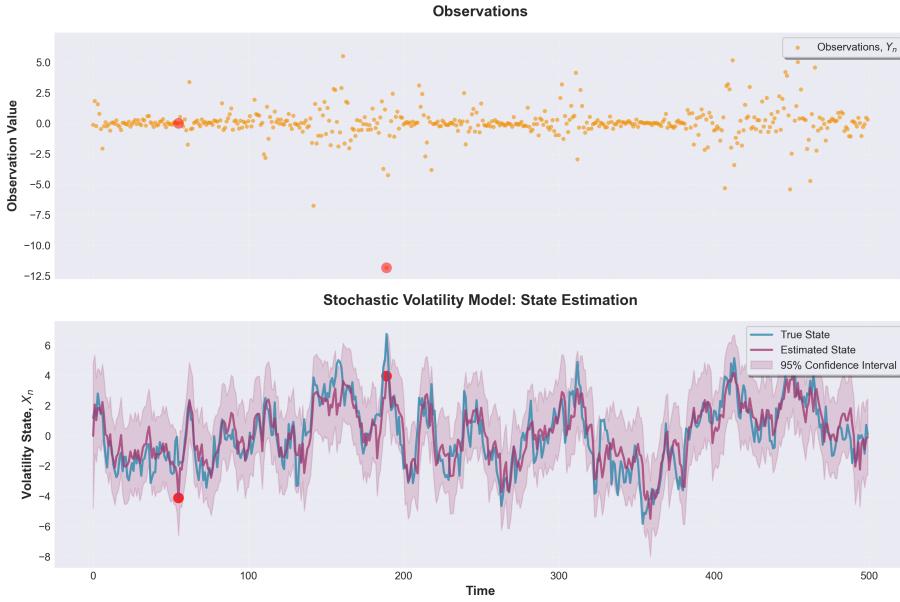


Figure 4: Results of the EKF for the SV model. The red dots marks the extreme points at $T=55$ and $T=189$.

EKF's imperfect performance lies in the linearization accuracy limits, which manifest in several critical ways:

1. Non-Gaussian Noise Distribution: The fundamental limitation lies in the fact that the observation noise $\log(W_n^2)$ is not Gaussian. Actually, $\log(W_n^2)$ follows a χ_1^2 distribution, which exhibits a heavy left tail. While the EKF assumes it to be $\mathcal{N}(-1.27, \pi^2/2)$ based on matching the first two moments, this Gaussian approximation fails to capture the true distribution's characteristics. The heavy-tailed nature means that extreme negative values occur more frequently than a Gaussian distribution would predict. When large observation noise is present, the EKF will be overconfident and the state estimate will be lag(eg. $T=189$ at figure 4).

2. Log-Transformation Amplification: The log-transformation approach, while necessary to linearize the multiplicative observation equation, introduces additional problems. When Y_n is close to zero, the transformation $\log(Y_n^2 + \epsilon)$ produces extreme negative values, creating outliers that dominate the filter's update in eq.(26). This means that if there happen to be some small observations in the true data, the filter will transform them into extreme values and overreact these events(eg. $T=55$ at figure 4).

3. Moment Matching vs. Distribution Matching: The EKF approach matches only the first two moments (mean and variance) of $\log(W_n^2)$ with a Gaussian distribution. However, the χ_1^2 distribution has significant skewness and kurtosis that are ignored. This moment-matching approach works well when the true distribution is approximately Gaussian, but breaks down when the distribution is highly non-Gaussian, as in this case.

4. First-Order Approximation Error: Here we didn't use the traditional way to implement the EKF, but we still briefly discuss the first-order approximation error. Traditonally, the EKF relies on first-order Taylor expansion around the predicted state. For the SV model, the observation

function $h(X_n) = \beta \exp(X_n/2)$ has a Jacobian that depends exponentially on the state. When the state uncertainty is large or the state deviates significantly from the linearization point, the first-order approximation becomes inaccurate.

2.3 Unscented Kalman Filter(UKF) and sigma point failures

It is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function.

A Appendix

A.1 Derivation of Covariance Update Equation for Kalman Filter

We want to derive $P_{n|n}$ from $P_{n|n-1}$ and K_n . By definition of $P_{n|n}$, we have

$$P_{n|n} = \text{cov} \left\{ X_n - \hat{X}_{n|n} \right\} \quad (30)$$

By taking (12) to replace $X_{n|n}$, we have:

$$P_{n|n} = \text{cov} \left\{ X_n - \left(\hat{X}_{n|n-1} + K_n \left(Y_n - C \hat{X}_{n|n-1} \right) \right) \right\} \quad (31)$$

Then plugging in eq.(8) to replace Y_n , we have:

$$\begin{aligned} P_{n|n} &= \text{cov} \left\{ X_n - \left(\hat{X}_{n|n-1} + K_n \left(C * X_n + D * W_n - C * \hat{X}_{n|n-1} \right) \right) \right\} \\ &= \text{cov} \left\{ (I - K_n * C) * (X_n - \hat{X}_{n|n-1}) + K_n * D * W_n \right\} \\ &= \text{cov} \left\{ (I - K_n * C) * (X_n - \hat{X}_{n|n-1}) \right\} + \text{cov} \{ K_n * D * W_n \} \\ &= (I - K_n * C) * \text{cov} \left\{ (X_n - \hat{X}_{n|n-1}) \right\} * (I - K_n * C)^T + K_n * D * D^T * K_n^T \\ &= (I - K_n * C) * \hat{P}_{n|n-1} * (I - K_n * C)^T + K_n * D * D^T * K_n^T \end{aligned} \quad (32)$$

This is the same as eq.(13).

A.2 Optimal Kalman Gain

The Kalman filter is a minimum mean square error estimator, which means that we want to minimize $E[(X_n - \hat{X}_{n|n})^2]$, which is exactly the trace of $P_{n|n}$.

Expanding Eq.(13) :

$$P_{n|n} = P_{n|n-1} - K_n * C * P_{n|n-1} - P_{n|n-1} * C^T * K_n^T + K_n * (C * C^T + D * D^T) * K_n^T \quad (33)$$

Taking the derivative of the trace of $P_{n|n}$ with respect to K_n and setting it to 0, we get:

$$\frac{\partial \text{tr}(P_{n|n})}{\partial K_n} = -2 * C * P_{n|n-1} + 2 * (C * C^T + D * D^T) * K_n = 0 \quad (34)$$

Here we use the property: $\frac{\partial \text{tr}(BAC)}{\partial A} = B^T C^T$ and $\frac{\partial \text{tr}(AB)}{\partial A} = B^T$. Solving for K_n , we get the same result as eq.(14).

References

- [1] Stephen J Taylor. *Modelling financial time series*. world scientific, 2008.
- [2] Andrew Harvey, Esther Ruiz, and Neil Shephard. Multivariate stochastic variance models. *The Review of Economic Studies*, 61(2):247–264, 1994.
- [3] Tsay, R.S. (2013) Multivariate Time Series Analysis With R and Financial Applications. Wiley, Hoboken. - References - Scientific Research Publishing. <https://www.scirp.org/reference/referencespapers?referenceid=2289642>.
- [4] Extended Kalman filter. *Wikipedia*, November 2025.
- [5] Milton Abramowitz. Handbook of mathematical functions, national bureau of standards. *Applied Mathematics Series*, (55), 1964.