Volatile vs. Standard Kalman Filters: A Didactic Overview

This document consolidates our discussion of **standard Kalman filters**, **volatile Kalman filters**, **Kalman learning rates**, and their relationships to **Hierarchical Gaussian Filters (HGF)** and **Behrens et al. (2007)** models. It includes equations in LaTeX, MATLAB code, and simulation outputs.

1. Standard Kalman Filter

A **standard Kalman filter** assumes a *fixed process noise* Q and *fixed observation noise* R.

State dynamics

$$x_{t+1} = x_t + w_t, \qquad w_t \sim \mathcal{N}(0, Q)$$

Observation model

$$y_t = x_t + v_t, \qquad v_t \sim \mathcal{N}(0, R)$$

Prediction step

$$\hat{x}_{t|t-1} = \hat{x}_{t-1|t-1}$$

$$P_{t|t-1} = P_{t-1|t-1} + Q$$

Kalman gain (trial-wise learning rate)

$$K_t = \frac{P_{t|t-1}}{P_{t|t-1} + R}$$

Update step

$$\delta_t = y_t - \hat{x}_{t|t-1}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \delta_t$$

$$P_{t|t} = (1 - K_t) P_{t|t-1}$$

Learning rate **decreases and stabilizes** in stationary environments.

2. Volatile Kalman Filter (vKF)

A **volatile Kalman filter** assumes that volatility itself changes across time.

State dynamics

$$x_{t+1} = x_t + w_t, \qquad w_t \sim \mathcal{N}(0, Q_t)$$

Volatility update rule (simple form)

$$Q_{t+1} = Q_t + \eta \left(\delta_t^2 - Q_t\right)$$

Where

- δ_t is the prediction error,
- η is a volatility learning rate.

Updated predicted variance

$$P_{t|t-1} = P_{t-1|t-1} + Q_t$$

Kalman gain

$$K_t = \frac{P_{t|t-1}}{P_{t|t-1} + R}$$

Learning rate increases when:

- Volatility (Q_t) increases
- Prediction errors are large

3. Covariance Relationships

Volatility increases learning rate

$$\frac{\partial K_t}{\partial Q_t} > 0$$

Observation noise decreases learning rate

$$\frac{\partial K_t}{\partial R} < 0$$

Prediction error increases volatility

$$\mathrm{Cov}(\delta_t^2, Q_{t+1}) > 0$$

These relationships underpin adaptive learning.

4. Relationship to HGF (Hierarchical Gaussian Filter)

The **HGF** (Mathys et al. 2011) is a hierarchical Bayesian update model with:

- Level 1: observations
- Level 2: beliefs about contingencies
- Level 3: volatility
- Higher levels encode uncertainty over volatility

The vKF corresponds to a **two-level HGF**:

- KF state estimate ↔ HGF level 2
- Volatility estimate Q_t ↔ HGF level 3
- Prediction-error-driven volatility updates ↔ precision-weighted surprise in HGF

HGF learning rate:

$$\alpha_t = \sigma_{2,t}\,\pi_{1,t}$$

Mirrors KF gain:

$$K_t = \frac{P_{t|t-1}}{P_{t|t-1} + R}$$

5. Relationship to Behrens et al. (2007)

Behrens et al. introduced a model where:

- Beliefs update proportional to **surprise**
- Volatility increases after surprising events
- Learning rate increases when volatility is high

This is mathematically equivalent to:

$$Q_{t+1} \propto \delta_t^2$$

and yields a dynamic learning rate similar to the vKF.

6. MATLAB Code: Standard and Volatile Kalman Filters

```
function [x_est, K_hist, Q_hist] = kalman_standard(y, Q, R)
    n = length(y);
    x_est = zeros(1,n);
    P = 1;
    for t = 1:n
        % Prediction
        P_pred = P + Q;

        % Kalman gain
        K = P_pred / (P_pred + R);
        K_hist(t) = K;

        % Update
        if t == 1
            x_pred = 0;
        else
            x_pred = x_est(t-1);
        end
        delta = y(t) - x_pred;
        x_est(t) = x_pred + K * delta;
        P = (1 - K) * P_pred;
    end
    Q_hist = Q * ones(1,n);
end
```

```
function [x_est, K_hist, Q_hist] = kalman_volatile(y, Q0, R, eta)
    n = length(y);
    x_est = zeros(1,n);
    Q = Q0;
    P = 1;
    for t = 1:n
        % Prediction variance
        P_pred = P + Q;

        % Kalman gain
        K = P_pred / (P_pred + R);
        K_hist(t) = K;
        Q_hist(t) = Q;

        % Update
        if t == 1
            x_pred = 0;
        else
            x_pred = x_est(t-1);
        end
        delta = y(t) - x_pred;
        x_est(t) = x_pred + K * delta;
        P = (1 - K) * P_pred;

        % Volatility update
        Q = Q + eta * (delta^2 - Q);
    end
end
```

7. MATLAB Simulation: Learning-Rate Comparison

```
% Simulated data with volatility
T = 200;
x_true = zeros(1,T);
Q_true = [0.01*ones(1,70), 0.2*ones(1,60), 0.01*ones(1,70)];
for t = 2:T
    x_true(t) = x_true(t-1) + sqrt(Q_true(t))*randn;
end
y = x_true + 0.1*randn(1,T);

% Run filters
[~, K_std, ~] = kalman_standard(y, 0.01, 0.1);
[~, K_vol, Qvol] = kalman_volatile(y, 0.01, 0.1, 0.1);

% Plot
figure; hold on;
plot(K_std, 'LineWidth', 2);
plot(K_vol, 'LineWidth', 2);
legend('Standard KF', 'Volatile KF');
xlabel('Trial'); ylabel('Learning Rate (Kalman Gain)');
title('Learning Rate Comparison');
```

In this simulation:

- Standard KF shows declining/stable learning rate
- Volatile KF shows spikes in learning rate where volatility increases
- This effect mirrors Behrens et al. (2007) and HGF-level behavior

8. Conceptual Summary

Feature	Standard KF	Volatile KF	HGF
State noise Q	Fixed	Dynamic	Dynamic at multiple levels
Learning rate	Stabilizes	Tracks volatility	Precision-weighted
Surprise effect	Weak	Strong	Strong
Matches	Stable tasks	Volatile environments	Psychophysics, psychiatry

9. References

- Behrens et al., Nature 2007
- Mathys et al., Frontiers 2011 (HGF)