Stock Option Pricing Using Bayes Filters

Author: Ziyang Bob Ding

April 12, 2020

Abstract

In this project, I utilize EKF (extended Kalman Filter), which is an extension of Kalman Filter, with a combination of the GARCH(1,1) model that generates the dynamics for implied volatilities, to model SPX500 options (European) and its latent volatility dynamics. For variable estimation in GARCH(1,1) model and latent volatility process, the MCMC algorithm is used together with Extended Kalman Smoothing, Linear Regression, and Inverse Gamma update. The finalized model is able to conduct volatility prediction up to days in which the stock price is observed.

1 GARCH(1,1)

1.1 Variable Explanation

- p_t Stock price on day t.
- u_t Stock return on day t. $u_t = \frac{p_t - p_{t-1}}{p_{t-1}}$
- θ_t Stock volatility on day t

1.2 Sub-model Specification

$$\begin{aligned} \theta_t &= \gamma V_L + \alpha u_t^2 + \beta\theta_{t-1} + w_t \\ w_t &\sim \mathcal{N}(0, W) \\ \gamma, \alpha, \beta &> 0 \\ \gamma + \alpha + \beta &= 1 \end{aligned}$$

Where V_L is the long-run average variance rate V_L. To better help our computation, we may relax our boundary constraint condition by treating γV_L as ω, therefore the model becomes:

$$\theta_t = \omega + \alpha u_t^2 + \beta\theta_{t-1} + w_t \tag{1}$$
$$w_t \sim \mathcal{N}(0, W) \tag{2}$$
$$\omega, \alpha, \beta > 0 \tag{3}$$
$$\alpha + \beta < 1 \tag{4}$$

2 Black-Schole's Formula

In this project, I will be dealing with European Option solely, considering it is easier for computation. Black-Schole's Formula is used to measure an European vanilla option's price based on the related assets (can be a portfolio of assets) return and volatility. The formula assumes the following:

1. Stock return follows quasi geometric Brownian Motion.
2. **The risk-free rate and volatility of the underlying are known and constant.**
3. **The returns on the underlying are normally distributed.**

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

The BS formula is the closed form solution of the above SPDE. It takes the following form:

$$y_{ti} = S\Phi\left(d_+\right) - K_{ti}e^{-r_t T_{ti}}\Phi\left(d_-\right)$$

where

$$\begin{aligned} d_+ &= \frac{\ln\left(\frac{p_t}{K_{ti}}\right) + \left(r_t + \frac{\theta_t}{2}\right)T_{ti}}{\sqrt{\theta_t T_{ti}}} \\ d_- &= d_+ - \sqrt{\theta_t T_{ti}} \end{aligned}$$

Where p_t is asset price; r_t is risk-free interest rate; T_{ti} is exercise time; K_{ti} is strike price; θ_t is volatility; Φ is CDF of normal.

3 Extended Kalman Filter

3.1 Sub-model Specification

$$\begin{aligned} \theta_t &= g_t(\theta_{t-1}, u_t, w_t) \\ y_{it} &= f_{ti}(\theta_t, p_t, K_{ti}, T_{ti}) + \nu_t \\ \nu_t &\sim N(0, v) \end{aligned}$$

Where

$$g(\theta_{t-1}, u_t, w_t) = \omega + \alpha u_t^2 + \beta\theta_{t-1} + w_t$$

is the GARCH(1,1) process. In other word, the GARCH(1,1) is representing the dynamics inside the EKF. And

$$f_{ti}(\theta_t, S, K_{ti}, T_{ti}) = BS(\theta_t, S, K_{ti}, T_{ti})$$

is the Black Scholes formula. Here, because on each day, the stock market has several several different Option choices. In order to incorporate all of their information, we have multiple outputs on each timestep as our target variable y_{it}, the option price on day t.

3.2 Forward Filtering

The theory of EKF lies in linearlizing the EKF at $E[\theta_{t-1}|\mathcal{D}_{t-1}] = m_{t-1}$ and $E[\theta_t|\mathcal{D}_{t-1}] = a_{t0}$ using Taylor first degree expansion. Therefore, we have

$$\begin{aligned}
\mathbb{P}(\theta_{t-1}|\mathcal{D}_{t-1}) &\sim \mathcal{N}(m_{t-1}, C_{t-1}) \\
\mathbb{P}(\theta_t|\mathcal{D}_{t-1}) &\sim \mathcal{N}(a_t, R_t) \\
a_t &= g_t(m_{t-1}) \\
R_t &= G_t C_{t-1} G_t^T + W \\
m_t &= \frac{R_t \sum_{i=1}^n F_{ti}\xi_{ti} + va_t}{v + \sum_{i=1}^2 F_{ti}} \\
C_t &= \frac{vR}{v + \sum_{i=n}^n F_{ti}}
\end{aligned}$$

Where

- $G_t = \frac{\partial g_t}{\partial \theta_{t-1}}\Big|_{\theta_{t-1}=m_{t-1}}$
- $F_{ti}^{\top} = \frac{\partial f_{ti}}{\partial \theta_t}\Big|_{\theta_t = a_t}$
- $\xi_{ti} = y_{ti} - h_{ti}(a_t)$
- $h_{ti}(a_t) = f_{ti}(a_t) - F_{ti}^T(a_t)$

(**Note:**) The update equation is not available for vector case as the closed form inverse matrix doesn't exist. While in our case, all the transitions are scalar, we're able to write out the correction step in closed form.

A detailed derivation can be found in Appendix.

3.3 Backward Smoothing

Backward Smoothing is particularly important when we're sampling the latent $\theta_{1:T}|\mathcal{D}_T$ for the MCMC parameter estimation part. Similar to forward filtering step, the backward smoothing also uses Taylor first degree expansion. Therefore we have:

$$\begin{aligned}
\mathbb{P}(\theta_t|\theta_{t+1:T}\mathcal{D}_T) &\sim \mathcal{N}(m_t^*, C_t^*) \\
\mathbb{P}(\theta_{t-1}|\theta_{t:T}, \mathcal{D}_T) &\sim \mathcal{N}(m_{t-1}^*, C_{t-1}^*) \\
m_{t-1}^* &= m_{t-1} + J_{t-1}(\theta_t - g_t(m_{t-1})) \\
C_{t-1}^* &= C_{t-1} - J_{t-1} R_t J_{t-1}^{\top}
\end{aligned}$$

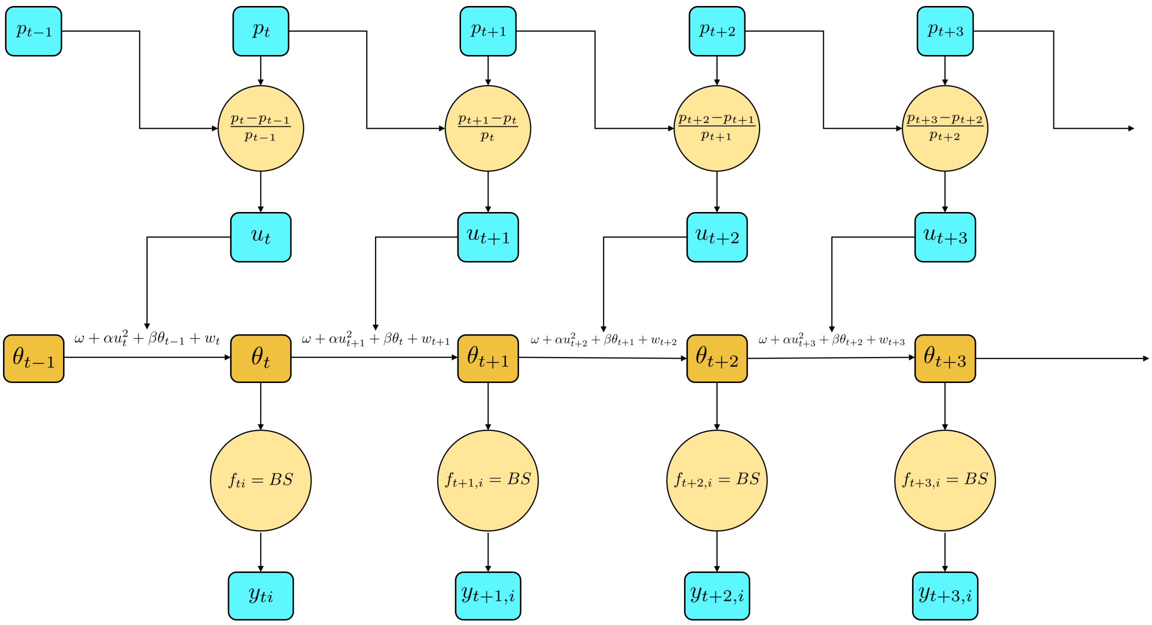
Where

- $m_{t-1} = \mathbb{E}[\theta_{t-1}|\mathcal{D}_{t-1}]$
- $C_{t-1} = \mathrm{Var}[\theta_{t-1}|\mathcal{D}_{t-1}]$
- $R_t = \mathrm{Var}[\theta_t|\mathcal{D}_t] = G_t C_{t-1} G_t^{\top} + W$
- $J_{t-1} = C_{t-1} G_t^{\top} R_t^{-1}$

A detailed derivation can be found in Appendix.

4 Model Illustration

Attached is a representation of the project model.



5 Parameter Inference

The Model requires statistical inference uppon the following parameters:

$$\{\omega, \alpha, \beta, W, v\}$$

We develop the MCMC algorithm that

1. Set priors $\mathbb{P}((\omega, \alpha, \beta)^\top), \mathbb{P}(W), \mathbb{P}(v)$. Random initialize.
2. Forward Filtering: Update all $m_t, C_t, \forall t \in 1:T$
3. Backward Smoothing: Sample $\theta_{1:T}$ from $\mathbb{P}(\theta_{1:T}|\mathcal{D}_T)$ by recursively sample θ_{t-1} from $\mathbb{P}(\theta_{t-1}|\theta_t, \mathcal{D}_T)$ using EKF smoothing.
4. Linear Regression with Constrains: Sample $(\omega, \alpha, \beta)^\top, W$ from $\mathbb{P}(\omega, \alpha, \beta|\theta_{1:T}, \mathcal{D}_T, W, v) = \mathbb{P}(\omega, \alpha, \beta, W|\theta_{1:T})$ with rejection sampling for constraints of equation (3) and (4).
5. Inverse Gamma: Sample v from $\mathbb{P}(v|\theta_{1:T}, \mathcal{D}_T, W, \omega, \alpha, \beta) = \mathbb{P}(W|\theta_{1:T}, \mathcal{D}_T)$
6. Repeat 2,3,4,5,6 until converges

5.1 Sampling $(\omega, \alpha, \beta)^\intercal$, and $W = \phi^{-1}$

This is a simple linear regression $\boldsymbol{\theta} = \mathbf{X}\boldsymbol{\beta} + w_t$ with design matrices as

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \vdots \\ \theta_T \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & u_1^2 & \theta_0 \\ 1 & u_2^2 & \theta_1 \\ 1 & u_3^2 & \theta_2 \\ 1 & u_4^2 & \theta_3 \\ 1 & u_5^2 & \theta_4 \\ 1 & \vdots & \vdots \\ 1 & u_T^2 & \theta_{T-1} \end{bmatrix} \qquad \boldsymbol{\beta} = \begin{bmatrix} \omega \\ \alpha \\ \beta \end{bmatrix}$$

Therefore,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\beta}; \boldsymbol{\theta}, \mathbf{X}) &\propto \phi^{\frac{T}{2}} \exp\{-\frac{1}{2}\phi(\boldsymbol{\theta} - \mathbf{X}\boldsymbol{\beta})^\intercal(\boldsymbol{\theta} - \mathbf{X}\boldsymbol{\beta})\} \\
\boldsymbol{\beta}|\phi &\sim \mathcal{N}\left(\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0^{-1}/\phi\right) \\
\phi &\sim \mathbf{G}\left(a_0 = \frac{v_0}{2}, b_0 = \frac{v_0 s_0^2}{2}\right) \\
\boldsymbol{\mu}_n &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \boldsymbol{\Lambda}_0\right)^{-1}\left(\boldsymbol{\Lambda}_0\boldsymbol{\mu}_0 + \mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}\right) \\
\boldsymbol{\Lambda}_n &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \boldsymbol{\Lambda}_0\right) \\
a_n &= a_0 + \frac{T}{2} \\
b_n &= b_0 + \frac{1}{2}(\boldsymbol{\theta}^\intercal\boldsymbol{\theta} + \boldsymbol{\mu}_0^\intercal\boldsymbol{\Lambda}_0\boldsymbol{\mu}_0 - \boldsymbol{\mu}_n^\intercal\boldsymbol{\Lambda}_n\boldsymbol{\mu}_n) \\
\boldsymbol{\beta}|\phi, \boldsymbol{\theta} &\sim \mathcal{N}\left(\boldsymbol{\mu}_n, \boldsymbol{\Lambda}_n^{-1}/\phi\right) \\
\phi|\boldsymbol{\theta} &\sim \mathbf{G}\left(a_n, b_n\right)
\end{aligned}$$

During sampling, once the condition (3),(4) is not satisfied, reject and resample until we reach one. First sample variance, then sample coefficient.

5.2 Sampling $v = \tau^{-1}$

This is a standard inverse gamma update

$$\begin{aligned}
y_{ti} &\sim \mathcal{N}(f_{ti}(\theta_t, p_t, K_{ti}, T_{ti}), \tau^{-1}) \\
\mathcal{L}(\tau; \{y\}_{1:T}^{1:n_t}) &\propto \tau^{\frac{\sum_{t=1}^T n_t}{2}} \exp\left\{-\frac{1}{2}\tau\sum_{t=1}^T\sum_{i=1}^{n_t}(y_{ti} - f_{ti})^2\right\} \\
\tau &\sim \mathbf{G}(\delta, \epsilon) \\
\tau|\{y\}_{1:T}^{1:n_t} &\sim \mathbf{G}(\delta + \frac{\sum_{t=1}^T n_t}{2}, \epsilon + \frac{1}{2}\sum_{t=1}^T\sum_{i=1}^{n_t}(y_{ti} - f_{ti})^2)
\end{aligned}$$

6 Experiment

- **Data Sources**: The data I will use are the SP 500 index option (or more) from year 1991 to 2002. The database is huge. For each day, it has option quotes for both call and put options.

Then for each of them, there exist a number of different expiration dates. And for each given expiration date, there are a few possible strike prices. Then for the given expiration date and strike price, it includes the price quotes for ”ask”,”bid”,”high”,”low” and ”closing”.

- **Computation**: As direct closed form solution of updates might not be available, I will use EM algorithm for optimization. The derivation of a closed form M step is not guaranteed. If so, I will utilize ECIM in replace of EM algorithm. But generally, using EM algorithm will be the majorly considered algorithm for optimization.
- **Control**: The design of control vector $\mathbf{u}_t$ will highly likely to be eliminated in this project. However, if the project proceeds well, I will link this project to a Chaos Theory project that I've been working on, in which a robust control vector $\mathbf{u}_t$ will be designed.
- **Progress**: I've already coded up the DLM python object that I can use directly. In fact, I've used the class in another project I have finished. As the previous DLM python object is only for singular variable, I will make it multivariate.

References

[1] G. Rigatos *A Kalman filtering approach for detection of option mispricing in the Black-Scholes PDE model.* 2014 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr), London, 2014, pp. 378-383.

[2] K. Liu and X. Wang *A Pragmatical Option Pricing Method Combining Black-Scholes Formula, Time Series Analysis and Artificial Neural Network.* 2013 Ninth International Conference on Computational Intelligence and Security, Leshan, 2013, pp. 149-153.

Appendix

EKF Forward Filtering

$$\begin{aligned}
h_{ti}(a_i) &= f_{ti}(a_t) - F_{ti}^T(a_t) \\
F_{ti}^T &= \frac{\partial f_{ti}}{\partial \theta_t}\Big|_{\theta_t = a_t} \\
d_t(m_{t-1}) &= g_t(m_{t-1}) - G_t m_{t-1} \\
G_t &= \frac{\partial g_t}{\partial \theta_{t-1}}\Big|_{\theta_{t-1} = m_{t-1}} \\
\xi_{ti} &= y_{ti} - h_{ti}(a_t) \\
a_t &= g_t(m_{t-1}) \\
R_t &= G_t C_{t-1} G_t^T + W_t \\
&\text{(Scalar Case)} \\
m_t &= \frac{R_t \sum_{i=1}^n F_{ti}\xi_{ti} + va}{v + \sum_{i=1}^2 F_{ti}} \\
C_t &= \frac{vR}{v + \sum_{i=1}^2 F_{ti}}
\end{aligned}$$

$$\begin{aligned}
\mathbb{P}(\theta_t|\mathcal{D}_t) &= \frac{\prod_i^n \mathbb{P}(y_{ti}|\theta_t, \mathcal{D}_{t-1})\mathbb{P}(\theta_t|\mathcal{D}_{t-1})}{\prod_i^n \mathbb{P}(y_{ti}|\mathcal{D}_{t-1})} \\
&\propto \prod_i^n \mathbb{P}(y_{ti}|\theta_t, \mathcal{D}_{t-1})\mathbb{P}(\theta_t|\mathcal{D}_{t-1}) \\
&= \prod_i^n N(f_{ti}(\theta_t), v) N(a_t, R_t) \approx \prod_i^n N(h_i(a_t) + F_{ti}^T \theta_t, v) N(a_t, R_t) \\
&\propto \exp\left\{-\frac{1}{2v}\sum_i^n ((y_{ti} - h_i(a_t)) - F_{ti}^T\theta_t))^2\right\} \exp\left\{-\frac{1}{2}(\theta_t - a_t)^T R_t^{-1}(\theta_t - a_t)\right\} \\
&\propto \exp\left\{-\frac{1}{2v}\sum_i^n (\xi_{ti} - F_{ti}^T\theta_t)^2\right\} \exp\left\{-\frac{1}{2}(\theta_t - a_t)^T R_t^{-1}(\theta_t - a_t)\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\frac{1}{v}\sum_i^n (\xi_{ti} - F_{ti}^T\theta_t)^2 + (\theta_t - a_t)^T R_t^{-1}(\theta_t - a_t)\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\frac{1}{v}\theta_t^T(\sum_{i=1}^n F_{ti}F_{ti}^T)\theta_t - 2\theta_t^T(\sum_i^n F_{ti}\xi_{ti})(v)^{-1} + \theta_t^T R_t^{-1}\theta_t - 2\theta_t^T R_t^{-1} a_t\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\theta_t^T\left(\frac{1}{v}(\sum_{i=1}^n F_{ti}F_{ti}^T) + R_t^{-1}\right)\theta_t - 2\theta_t^T\left(\frac{1}{v}(\sum_i^n F_{ti}\xi_{ti}) + R_t^{-1}a_t\right)\right]\right\} \\
&\sim N\left(\left(\frac{1}{v}(\sum_{i=1}^n F_{ti}F_{ti}^T) + R_t^{-1}\right)^{-1}\left(\frac{1}{v}(\sum_i^n F_{ti}\xi_{ti}) + R_t^{-1}a_t\right), \left(\frac{1}{v}(\sum_{i=1}^n F_{ti}F_{ti}^T) + R_t^{-1}\right)^{-1}\right) \\
&= N(m_t, C_t)
\end{aligned}$$

EKF Backward Smoothing

$$\begin{aligned}\log P\left(\theta_{t+1},\theta_t|\{y\}_1^T\right) &= \log P\left(\theta_{t+1}|\theta_t\right) + \log P\left(\theta_t|\{y\}_1^t\right) - \log P\left(\theta_{t+1}|\{y\}_1^t\right) + \log P\left(\theta_{t+1}|\{y\}_1^T\right)\\
&= -\frac{1}{2}\left(\theta_{t+1} - G_{t+1}\theta_t - d_{t+1}(m_t)\right)^\top W_{t+1}^{-1}\left(\theta_{t+1} - G_{t+1}\theta_t - d_{t+1}(m_t)\right)\\
&- \frac{1}{2}\left(\theta_t - m_t\right)^\top\left(C_t\right)^{-1}\left(\theta_t - m_t\right) + \frac{1}{2}\left(\theta_{t+1} - a_{t+1}\right)^\top\left(R_{t+1}\right)^{-1}\left(\theta_{t+1} - a_{t+1}\right)\\
&- \frac{1}{2}\left(\theta_{t+1} - m_{t+1}^*\right)^\top\left(C_{t+1}^*\right)^{-1}\left(\theta_{t+1} - m_{t+1}^*\right) + \dots\\
&= -\frac{1}{2}\theta_{t+1}^\top\left(W_{t+1}^{-1} - \left(R_{t+1}\right)^{-1} + \left(C_{t+1}^*\right)^{-1}\right)\theta_{t+1}\\
&- \frac{1}{2}\theta_{t+1}^\top\left(-W_{t+1}^{-1}G_{t+1}\right)\theta_t - \frac{1}{2}\theta_t^\top\left(-G_{t+1}^\top W_{t+1}^{-1}\right)\theta_{t+1}\\
&- \frac{1}{2}\theta_t^\top\left(G_{t+1}^\top W_{t+1}^{-1}G_{t+1} + \left(C_t\right)^{-1}\right)\theta_t + \theta_t^\top\left(-G_{t+1}^\top W_{t+1}^{-1}d_{t+1}(m_t) + \left(C_t\right)^{-1}m_t\right) + \dots\end{aligned}$$

Therefore, we have

$$\begin{bmatrix}\theta_t\\ \theta_{t-1}\end{bmatrix}\Big|\mathcal{D}_T = \mathcal{N}\left(\begin{bmatrix}\mu_t\\ \mu_{t-1}\end{bmatrix},\begin{bmatrix}\Phi_{t,t} & \Phi_{t,t-1}\\ \Phi_{t-1,t} & \Phi_{t-1,t-1}\end{bmatrix}^{-1}\right) = \mathcal{N}\left(\begin{bmatrix}\mu_t\\ \mu_{t-1}\end{bmatrix},\begin{bmatrix}\Sigma_{t,t} & \Sigma_{t,t-1}\\ \Sigma_{t-1,t} & \Sigma_{t-1,t-1}\end{bmatrix}\right)$$

$$\begin{aligned}\Phi_{t,t} &= W_t^{-1} - \left(R_t\right)^{-1} + \left(C_t^*\right)^{-1}\\
\Phi_{t,t-1} &= -W_t^{-1}G_t\\
\Phi_{t-1,t} &= -G_t^\top W_t^{-1}\\
\Phi_{t-1,t-1} &= G_t^\top W_t^{-1}G_t + \left(C_{t-1}\right)^{-1}\end{aligned}$$

$$\begin{aligned}\Phi_{t-1,t-1}^{-1} &= C_{t-1} - J_{t-1}R_tJ_{t-1}^\top\\
J_{t-1} &= C_{t-1}G_t^\top R_t^{-1}\\
\Phi_{t-1,t-1}^{-1}\Phi_{t,t-1} &= -J_{t-1}\\
C_{t-1}^* &= C_{t-1} + J_{t-1}(C_t^* - R_t)J_{t-1}^\top\end{aligned}$$

$$\begin{aligned}\Sigma_{t,t} &= C_t^*\\
\Sigma_{t,t-1} &= C_t^*J_{t-1}^\top\\
\Sigma_{t-1,t} &= J_{t-1}C_t^{*\top}\\
\Sigma_{t-1,t-1} &= C_{t-1} + J_{t-1}(C_t^* - R_t)J_{t-1}^\top\end{aligned}$$

$$\begin{aligned}\mu_t &= m_t^*\\
\mu_{t-1} &= m_{t-1} + J_{t-1}(m_t^* - g_t(m_{t-1}))\end{aligned}$$

So the conditional sampling distribution should be:

$$\begin{aligned}
\theta_{t-1}|\theta_t, \mathcal{D}_t &\sim \mathcal{N}(\mu_{t-1} + \Sigma_{t,t-1}^T \Sigma_{t,t}^{-1}(\theta_{t-1} - \mu_t), \Sigma_{t-1,t-1} - \Sigma_{t,t-1}^T \Sigma_{t,t}^{-1} \Sigma_{t-1,t}^T) \\
&= \mathcal{N}(m_{t-1} + J_{t-1}(m_t^* - g_t(m_{t-1})) + (C_t^* J_{t-1}^\top)^\top (C_t^*)^{-1}(\theta_{t-1} - m_t^*), \\
&\quad C_{t-1} + J_{t-1}(C_t^* - R_t)J_{t-1}^\top - (C_t^* J_{t-1}^\top)^\top (C_t^*)^{-1}(J_{t-1} C_t^{*\top})^\top) \\
&= \mathcal{N}\Big(m_{t-1} + J_{t-1}(\theta_t - g_t(m_{t-1})), \quad C_{t-1} - J_{t-1} R_t J_{t-1}^\top\Big)
\end{aligned}$$

Last Revised: April 12, 2020