Algorithm 1 VB-AKF algorithm

- Initialize $\nu_0, \mathbf{m}_0, \mathbf{P}_0$ and Σ_0.
- For $k = 1, 2, \ldots$
  - **Prediction**: compute the parameters of the predicted distribution:

$$
\begin{aligned}
\mathbf{m}_k^- &= \mathbf{A}_{k-1}\, \mathbf{m}_{k-1}, \\
\mathbf{P}_k^- &= \mathbf{A}_{k-1}\, \mathbf{P}_{k-1}\, \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}, \\
\nu_k^- &= \rho(\nu_{k-1} - d - 1) + d + 1, \\
\Sigma_k^- &= \mathbf{B}\, \Sigma_{k-1}\, \mathbf{B}^T,
\end{aligned}
$$

  - **Update**: set $\nu_k = \nu_k^- + 1$ and $\Sigma_k^{(1)} = \Sigma_k^-$. Iterate the following until the convergence (say, N times for $j = 1, \ldots, N$):

$$
\begin{aligned}
\mathbf{S}_k^{(j+1)} &= \mathbf{H}_k\, \mathbf{P}_k^-\, \mathbf{H}_k^T + \Sigma_k^{(j)}, \\
\mathbf{K}_k^{(j+1)} &= \mathbf{P}_k^-\, \mathbf{H}_k^T \left(\mathbf{S}_k^{(j+1)}\right)^{-1}, \\
\mathbf{m}_k^{(j+1)} &= \mathbf{m}_k^- + \mathbf{K}_k^{(j+1)} \left(\mathbf{y}_k - \mathbf{H}_k\, \mathbf{m}_k^-\right), \\
\mathbf{P}_k^{(j+1)} &= \mathbf{P}_k^- - \mathbf{K}_k^{(j+1)}\, \mathbf{S}_k^{(j+1)} \left(\mathbf{K}_k^{(j+1)}\right)^T, \\
\Sigma_k^{(j+1)} &= \left(\frac{\nu_{k-1} - d - 1}{\nu_k - d - 1}\right) \Sigma_k^- + \left(\frac{1}{\nu_k - d - 1}\right) \mathbf{H}_k\, \mathbf{P}_k^{(j+1)}\, \mathbf{H}_k^T \\
&\quad + \left(\frac{1}{\nu_k - d - 1}\right) \left(\mathbf{y}_k - \mathbf{H}_k\, \mathbf{m}_k^{(j+1)}\right) \left(\mathbf{y}_k - \mathbf{H}_k\, \mathbf{m}_k^{(j+1)}\right)^T.
\end{aligned}
\tag{22}
$$

  - Set $\Sigma_k = \Sigma_k^{(N)}$, $\mathbf{m}_k = \mathbf{m}_k^{(N)}$ and $\mathbf{P}_k = \mathbf{P}_k^{(N)}$.

In Algorithm 1, the choice of the number of iterations N depends on the problem at hand. However, in the numerical examples we tested, we found out that the algorithm requires only a few iterations to converge (we used $N = 5$). However, it would also be possible to use a stopping criterion which determines a suitable time to stop by monitoring the changes in the estimates at each iteration. In the next section we will use the above algorithm to compute the covariance matrix Σ_k for a proposal distribution.