

# The Kalman filter and the square root Kalman filter using only QR decompositions

Koichi (Koiti) Yano

## Introduction

In this note, I outline the algorithms for the Kalman filter and the square root Kalman filter using only QR decompositions. The Kalman filter is a recursive algorithm that estimates the state vector of a linear Gaussian state space model. The square root Kalman filter using only QR decompositions is an alternative to the Kalman filter, which is proposed by Tracy (2022). The square root Kalman filter is numerically more stable than the Kalman filter.

## The Kalman filter and the square root Kalman filter using only QR decompositions

### A linear Gaussian state space model

A linear Gaussian state space model is defined by the following two equations.

$$x_t = Fx_{t-1} + Eu_t + w_t$$

and

$$y_t = Hx_t + v_t,$$

where  $x_t$  is the  $(k \times 1)$  state vector,  $y_t$  is the  $(l \times 1)$  observation vector,  $u_t$  is the  $(n \times 1)$  exogenous vector,  $v_t$  is the  $(k \times 1)$  state noise,  $w(t)$  is the  $(l \times 1)$  observation noise,  $t$  is a time index. The matrices  $F$ ,  $E$ , and  $H$  are  $(k \times k)$ ,  $(k \times n)$ , and  $(l \times k)$ , respectively. The two innovations  $w_t$  and  $v_t$  are sampled from  $N(0, W)$  and  $N(0, V)$ , where  $W$  is a  $(k \times k)$  covariance matrix and  $V$  is an  $(l \times l)$  covariance matrix. The Kalman Filter is used to estimate the state vector  $x_t$  given the observations  $y_t$ .

The Kalman filter is implemented using the following recursion:

1. Prediction step:

1.  $x_{t|t-1} = Fx_{t-1|t-1} + Eu_t$
2.  $P_{t|t-1} = FP_{t-1|t-1}F^t + V$

2. Innovation step:

1.  $e_t = y_t - Hx_{t|t-1}$
2.  $s_t = HP_{t-1|t-1}H^t + W$

3. Update step:

1.  $K_t = P_{t|t-1}H^ts_t^{-1}$
2.  $x_{t|t} = x_{t|t-1} + K_te_t$
3.  $P_{t|t} = (I - K_tH)P_{t|t-1}$

where  $x_{t|t}$  is the estimated state vector at,  $P_{t|t}$  is the estimated state covariance matrix, and  $K_t$  is the Kalman gain at time t. See Kitagawa, (2010) for more details.

### **The square root Kalman filter using only QR decompositions**

The square root Kalman filter is an alternative to the Kalman filter that is numerically more stable. Tracy (2022) proposes the following recursion for The square root Kalman filter using only QR decompositions (hereafter the QR Kalman filter).

The QR Kalman filter is implemented using the following recursion

1. Prediction step

1.  $x_{t|t-1} = Fx_{t-1|t-1} + Eu_t$
2.  $\Sigma_{t|t-1} = gr_r(\Sigma_{t-1|t-1}F^t, \Gamma_v)$

2. Innovation step:

1.  $e_t = y_t - Hx_{t|t-1}$
2.  $G_t = gr_r(\Sigma_{t|t-1}H^t, \Gamma_w)$

3. Update step:

1.  $K_t = [G_t^{-1}(G_t^{-t}H)\Sigma_{t|t-1}^t(\Sigma_{t|t-1})]^t$
2.  $x_{t|t} = x_{t|t-1} + K_te_t$

$$3. \Sigma_{t|t} = gr_r(\Sigma(t|t-1)(I - K(t)H)^t, \Gamma_w K(t)^t),$$

where  $\Sigma_{t|t} = \sqrt{P_{t|t}}$ . The function  $gr_r$  returns the matrix  $R$  of the QR decomposition of the matrix. The matrices  $\Gamma_v$  and  $\Gamma_w$  are the Cholesky decompositions of the covariance matrices of the state noise and the observation noise, respectively. See Tracy (2022) for more details.

## Running Code

**To be added:** When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).