Description of package TART

Wataru Yoshida and Kei Hirose

Kyushu University

The package TART contains functions ART-KF and TART-KF. These perform sequential sparse estimation of the linear-Gaussian state-space model rapidly. Please note that it is necessary to prepare an environment to compile C++ code to run them. It is possible to install the package TART and conduct a simple simulation by running the test code in the file "example" published on GitHub. The details of our methods are shown below.

1 Introduction to the model and existing methods

Let $y_1, y_2, \ldots, y_n$ be $d \times 1$ observation vectors. Assume that y_t follows the linear-Gaussian state-space model:

$$\begin{cases} y_t = Z_t \alpha_t + \varepsilon_t, \\ \alpha_{t+1} = T_t \alpha_t + \eta_t, \end{cases} \qquad (t = 1, 2, \ldots, n), \tag{1.1}$$

where α_t is a $p \times 1$ state vector, ε_t is a $d \times 1$ observation noise vector, η_t is a $p \times 1$ state noise vector, Z_t is a $d \times p$ design matrix, and T_t is a $p \times p$ transition matrix. We assume $\varepsilon_t \sim N(0, \Sigma_\varepsilon)$, $\eta_t \sim N(0, \Sigma_\eta)$, $\alpha_1 \sim N(a_1, P_1)$, and ε_t, $\varepsilon_k (t \neq k)$, η_t, and η_k are mutually uncorrelated. In model (1.1), α_t can be estimated sequentially with the Kalman filter [Kalman 1960] as follows:

$$\begin{cases} a_{t|t}^{[KF]} = a_t^{[KF]} + K_t(y_t - Z_t a_t^{[KF]}), \quad P_{t|t} = P_t - K_t(Z_t P_t Z_t^T + \Sigma_\varepsilon) K_t^T, \\ a_{t+1}^{[KF]} = T_t a_{t|t}^{[KF]}, \quad P_{t+1} = T_t P_{t|t} T_t^T + \Sigma_\eta, \end{cases} \qquad (t = 1, 2, \ldots, n),$$

where $K_t := P_t Z_t^{\mathrm{T}} (Z_t P_t Z_t^T + \Sigma_\varepsilon)^{-1}$. At time t, $a_{t|t}^{[KF]}$ and $a_{t+1}^{[KF]}$ denote the estimation of α_t and the forecast of α_{t+1}, respectively. Under the assumption of the linear-Gaussian state-space model (1.1), it holds that $a_{t|t}^{[KF]} = E(\alpha_t | y_1, \ldots, y_t)$, $a_{t+1}^{[KF]} = E(\alpha_{t+1} | y_1, \ldots, y_t)$, $P_{t|t} := \mathrm{Var}(\alpha_t | y_1, \ldots, y_t)$, and $P_{t+1} := \mathrm{Var}(\alpha_{t+1} | y_1, \ldots, y_t)$. The estimator $a_{t|t}^{[KF]}$ can be rewritten as follows:

$$a_{t|t}^{[KF]} = (Z_t^T \Sigma_\varepsilon^{-1} Z_t + P_t^{-1})^{-1} [Z_t^T \Sigma_\varepsilon^{-1} y_t + P_t^{-1} a_t^{[KF]}]. \tag{1.2}$$

Li et al. [2014] pointed out that the value of $(Z_t^T \Sigma_\varepsilon^{-1} Z_t + P_t^{-1})^{-1}$ is sometimes unstable, and proposed the following ridge-type Kalman filter as an improvement method.

$$a_{t|t}^{[RT]}(\lambda) = (Z_t^T \Sigma_\varepsilon^{-1} Z_t + P_t^{-1} + \lambda I_p)^{-1} [Z_t^T \Sigma_\varepsilon^{-1} y_t + P_t^{-1} a_t^{[KF]}]. \tag{1.3}$$

Here, we find that $a_{t|t}^{[RT]}(\lambda)$ can be expressed as the solution of the following minimization problem.

Theorem 1.1.
$a_{t|t}^{[RT]}(\lambda) = \arg\min_{\alpha_t} \Big\{ -\log f(\alpha_t | y_1, \ldots, y_t) + \frac{1}{2}\lambda \|\alpha_t\|_2^2 \Big\}$,
where $f(\alpha_t | y_1, \ldots, y_t)$ denotes the conditional distribution of α_t given $y_1, \ldots, y_t$.

The proof is detailed in Appendix A. From this theorem, the ridge-type Kalman filter can be generalized as follows:

$$\arg\min_{\alpha_t} \Big\{ -\log f(\alpha_t | y_1, \ldots, y_t) + \frac{1}{2} P(\alpha_t) \Big\}. \tag{1.4}$$

2 ART-KF

We can extend the ridge-type Kalman filter by changing the penalty term of sparse estimation in (1.4). In particular, we propose applying the adaptive ridge regression [Dai et al. 2018, Ho and Masuda 2024] (also, the results of applying lasso are shown in Assumption B).

$$\begin{cases} a_{t|t}^{[ART,0]}(\lambda) = (Z_t^T \Sigma_\varepsilon^{-1} Z_t + P_t^{-1} + \lambda I_p)^{-1} [Z_t^T \Sigma_\varepsilon^{-1} y_t + P_t^{-1} a_t^{[ART,S]}(\lambda)], \\ a_{t|t}^{[ART,s]}(\lambda) = \Big\{ Z_t^T \Sigma_\varepsilon^{-1} Z_t + P_t^{-1} + \lambda D\Big(a_{t|t}^{[ART,s-1]}(\lambda) \Big) \Big\}^{-1} [Z_t^T \Sigma_\varepsilon^{-1} y_t + P_t^{-1} a_t^{[ART,S]}(\lambda)], \end{cases}$$
$$(s = 1, 2, \ldots, S), \tag{2.1}$$

where, $a_t^{[ART,S]}(\lambda) := T_{t-1} a_{t-1|t-1}^{[ART,S]}(\lambda)$ and $D(x) := Diag(\frac{1}{x_1^2 + \delta_{AR}}, \ldots, \frac{1}{x_p^2 + \delta_{AR}})$ for $x = (x_1, \ldots, x_p)^T \in \mathbb{R}^p$. The δ_{AR} is a small value set by users to avoid dividing by 0. In this method, the estimator of the state vector α_t is $a_{t|t}^{[ART,S]}(\lambda)$. It can be sparse estimated under $S \to \infty$. In practical, by setting S large enough, some components of $a_{t|t}^{[ART,S]}(\lambda)$ are almost zero.

In the (2.1), $a_t^{[ART,S]}(\lambda)$ is used instead of $a_t^{[KF]}$. That is to say, the estimator $a_{t|t}^{[ART,S]}(\lambda)$ is used at the next point for deriving $a_{t+1|t+1}^{[ART,S]}(\lambda)$. This is different from the ridge-type Kalman filter (1.3). Note that if $\lambda = 0$, then this algorithm is equal to the Kalman filter. When the lasso is applied instead of the adaptive ridge, it is more accurate if $a_t^{[KF]}$ is used. These are discussed in more detail in Appendix B.

We call the above method the "adaptive ridge type Kalman filter" (ART-KF). The following is the summary of the ART-KF algorithm. The algorithm only requires some additional matrix computations to the Kalman filter.

Algorithm 1 ART-KF

```
**Input:** $y_{1:n}, Z_{1:n}, T_{1:n}, \Sigma_\varepsilon, \Sigma_\eta, a_1, P_1, \lambda, S, \delta_{AR}$
**Output:** ART-KF estimaters $a_{1|1:n|n}^{[ART,S]}(\lambda)$ and predictions $a_{1:n+1}^{[ART,S]}(\lambda)$
for $t = 1, \dots n$ do
    $a_{t|t}^{[KF]} \leftarrow a_t^{[KF]} + K_t(y_t - Z_t a_t^{[KF]}),\quad P_{t|t} \leftarrow P_t - K_t(Z_t P_t Z_t^T + \Sigma_\varepsilon)K_t^T,$
    $a_{t+1}^{[KF]} \leftarrow T a_{t|t}^{[KF]},\quad P_{t+1} \leftarrow T_t P_{t|t} T_t^T + \Sigma_\eta$ (Do Kalmdan filter)
    Compute $a_{t|t}^{[ART,S]}(\lambda)$ by (2.1)
    $a_{t+1}^{[ART,S]}(\lambda) \leftarrow T_t a_{t|t}^{[ART,S]}(\lambda)$
end for
```

3 TART-KF

3.1 Sequential optimization of λ

We next discuss how to set the parameter λ in the equation (2.1). Considering the situation where the model is time-varying, it is inferred that the optimal λ is different for each time. For toy example, the following model is considered:

$$\begin{cases} y_t = Z_t \times 0 + \varepsilon_t \quad (t = 1, \dots, 1000), \\ y_t = Z_t \alpha_t + \varepsilon_t,\ \alpha_{t+1} = T_t \alpha_t + \eta_t \quad (t = 1001, \dots, 2000). \end{cases} \tag{3.1}$$

In this model, for $t = 1, \dots, 1000$, it is optimal to set λ large and estimate all state variables to be 0. In contrast, for $t = 1001, \dots, 2000$, since all components of true α_t are nonzero, λ should be set small.

Now, we define λ_t as λ at time t and consider the optimization problem of $\lambda_1, \dots, \lambda_n$. Here, it is computationally difficult to verify different values for all λ_t. Therefore, we propose a method to optimize λ_t sequentially with the ART-KF. Specifically, we consider updating the one-step-ahead forecast error $e_{t+1}^{[ART]}(\lambda) = y_{t+1} - Z_{t+1} a_{t+1}^{[ART,S]}(\lambda)$ to be smaller using Adam [Kingma and Ba 2014]:

$$\begin{cases} g_{t+1} = \frac{1}{2\delta_g}\left\{L_t\left(e_{t+1}^{[ART]}(\lambda_t + \delta_g)\right) - L_t\left(e_{t+1}^{[ART]}(\lambda_t - \delta_g)\right)\right\}, \\ m_{t+1} = \beta_1 m_t + (1 - \beta_1) g_{t+1}, \quad \hat{m}_{t+1} = \frac{m_{t+1}}{(1 - \beta_1^t)}, \\ v_{t+1} = \beta_2 v_t + (1 - \beta_2) g_{t+1}^2, \quad \hat{v}_{t+1} = \frac{v_{t+1}}{(1 - \beta_2^t)}, \\ \lambda_{t+1} = \lambda_t - \gamma \frac{\hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \delta_v}, \end{cases} \quad (t = 1, \dots, n), \tag{3.2}$$

where, $L_t(e) := \frac{1}{d} e^T M_t^{-1} e$, $M_t := Diag(\bar{e}_{1,t}^{[ART]2}, \dots, \bar{e}_{d,t}^{[ART]2})$, $\bar{e}_{j,t}^{[ART]2} := \frac{1}{1+c_e} \sum_{s=t-c_e}^{t} e_{j,s}^{[ART]2}$. In (3.2), λ_t is updated in the direction that $L_t(e_{t+1}^{[ART]})$ becomes smaller. Here, $L_t(e_{t+1}^{[ART]})$ denotes the sum of squared forecast errors normalized by the squared mean M_t at the recent $1 + c_e$ time. Note that the derivative of $L_t(e_{t+1}^{[ART]})$ is approximated by g_{t+1}, since it is difficult to obtain it explicitly.

3.2 Control of λ_t with Kalman filter

It is risky to focus only on the forecast error $e^{[ART]}_{t+1}(\lambda_t)$ and update λ_t; this is because λ_t may become too large. Then, we consider controlling λ_t by comparing $e^{[ART]}_{t+1}(\lambda)$ and the forecast error of the Kalman filter $e^{[KF]}_{t+1} := y_{t+1} - Z_{t+1}a^{[KF]}_{t+1}$:

$$\lambda_{t+1} = \lambda_t - \Big(L_t(e^{[ART]}_{t+1}(\lambda_t)) - L_t(e^{[KF]}_{t+1})\Big) d_F F_t \bar{U}_t \quad \Big(\text{if } L_t(e^{[ART]}_{t+1}(\lambda_t)) > L_t(e^{[KF]}_{t+1})\Big), \tag{3.3}$$

where $\bar{U}_t$ denotes the absolute mean of the updated values by Adam up to time t, and F_t is the number of times $L_t(e^{[ART]}_{t+1}(\lambda_t)) > L_t(e^{[KF]}_{t+1})$ holds from time $t - c_F$ to t. That is, when the Kalman filter has better prediction accuracy, λ_t is reduced and its update width depends on the mean of Adam's update width. While Adam adjusts the update width according to the size of the gradient, this control method adjusts it according to the difference in prediction accuracy $L_t(e^{[ART]}_{t+1}(\lambda_t)) - L_t(e^{[KF]}_{t+1})$ and the number of times the Kalman filter's prediction is better at the recent $1 + c_F$ time. d_F is the parameter that controls this adjustment width.

3.3 TART-KF

The following is the summary of an algorithm that combines ART-KF with λ control. We call this algorithm the "twin direction adaptive ridge type Kalman filter" (TART-KF). As with ART-KF, the algorithm only adds some matrix computations to the Kalman filter.

Algorithm 2 TART-KF

```
Input: $y_{1:n}, Z_{1:n}, T_{1:n}, \Sigma_\varepsilon, \Sigma_\eta, a_1, P_1, \lambda_1, S, \delta_{AR}, \delta_g, \beta_1, \beta_2, \delta_v, \gamma, c_e, c_F, d_F$.
Output: TART-KF estimaters $a^{[TART]}_{1|1:n|n}$ and predictions $a^{[TART]}_{1:n+1}$
for $t = 1, \dots n$ do
    $a^{[KF]}_{t|t} \leftarrow a^{[KF]}_t + K_t(y_t - Z_t a^{[KF]}_t)$,   $P_{t|t} \leftarrow P_t - K_t(Z_t P_t Z_t^T + \Sigma_\varepsilon)K_t^T$,
    $a^{[KF]}_{t+1} \leftarrow T a^{[KF]}_{t|t}$,   $P_{t+1} \leftarrow T_t P_{t|t} T_t^T + \Sigma_\eta$ (Do Kalmdan filter)
    Compute $a^{[ART,S]}_{t|t}(\lambda_t)$ by (2.1)
    $a^{[TART]}_{t|t} \leftarrow a^{[ART,S]}_{t|t}(\lambda_t)$,   $a^{[TART]}_{t+1} = T_t a^{[TART]}_{t|t}$
    if $L_t(e^{[ART]}_{t+1}(\lambda_t)) \leq L_t(e^{[KF]}_{t+1})$ then
        Update $\lambda_t$ to $\lambda_{t+1}$ by the Adam (3.2)
        $U \leftarrow (U, |\lambda_{t+1} - \lambda_t|)$
    else
        Update $\lambda_t$ to $\lambda_{t+1}$ by (3.3)
    end if
end for
```

4 Numerical examples

4.1 Simple example

We now review the behavior of the TART-KF through a simple numerical simulation. The simulation model is set as

$$\begin{cases} y_t = \alpha_t + \varepsilon_t, \\ \alpha_{t+1} = \max\Big(4 - (\frac{1000-t}{1000})^2, 0\Big), \end{cases} \qquad (t = 1, \dots, 5000),$$

where $\varepsilon_t \sim N(0,1)$. The parameters required for the Kalman filter are set as $a_1 = 0$, $P_1 = 1000$, $\Sigma_\varepsilon = 1$, and $\Sigma_\eta = 10^{-4}$. We set the initial value of λ as $\lambda_1 = 0$. In (2.1), we set $S = 3$, and $\delta_{AR} = 10^{-8}$. Adam's parameters are set to $\delta_g = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\delta_v = 10^{-8}$, $\gamma = 0.002$, and $c_e = 100$. Finally, we set $c_F = 5$ and $d_F = 1$ in (3.3).

Figure 4.1 shows the estimation result of the TART-KF. The estimates are close to the true α_t and are shrunk to 0 after $t = 3000$. Figure 4.2 shows the transition of λ_t. From this figure, λ_t becomes large around $t = 3000$ where α_t becomes 0. After that, it is controlled not to get larger than necessary.

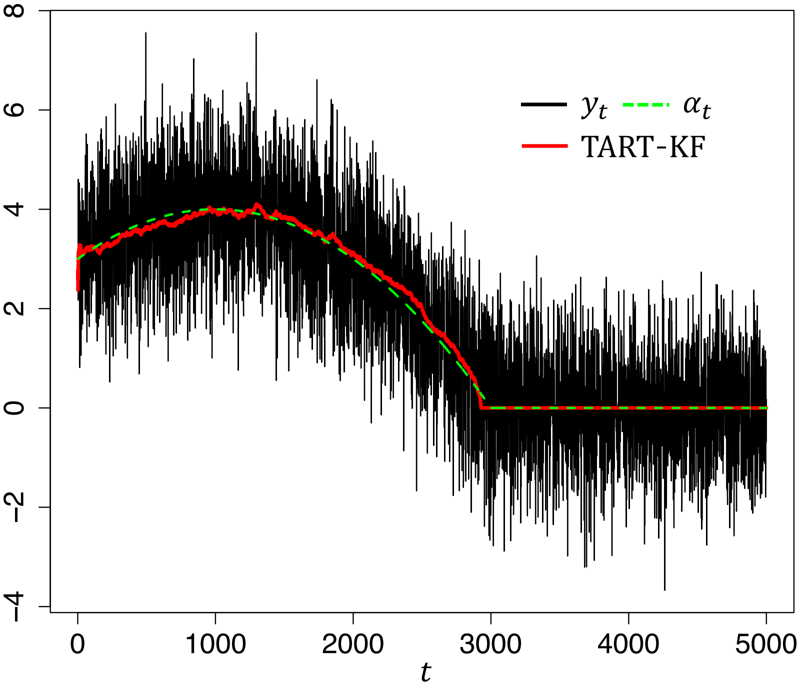

Figure 4.1 : Comparison of MSE between three methods

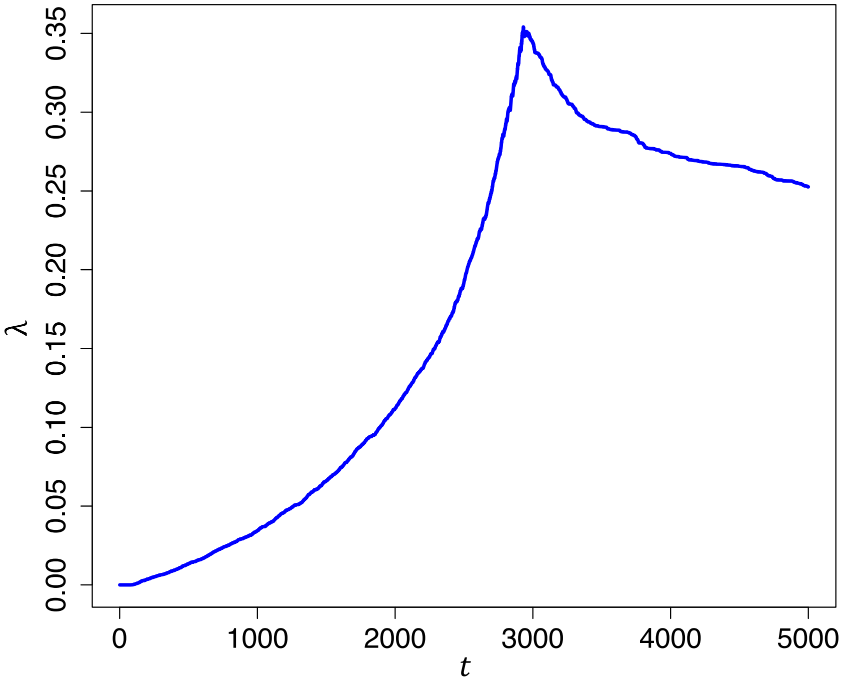

Figure 4.2 : Mean of λ_t for each time

4.2 Monte Carlo simulation

In this section, a Monte Carlo simulation is performed to demonstrate the validity of the proposed methods. The model used in the simulation is as follows:

$$\begin{cases} y_t = Z_t^{(1)}\alpha_t^{(1)} + Z_t^{(2)} \times 0 + \varepsilon_t, \\ \alpha_{t+1}^{(1)} = \alpha_t^{(1)} + \eta_t^{(1)}, & (t = 1, \dots, 2000), \\ \alpha_{t+1}^{(2)} = 0, \end{cases}$$

$$\begin{cases} y_t = Z_t^{(1)}\alpha_t^{(1)} + Z_t^{(2)}\alpha_t^{(2)} + \varepsilon_t, \\ \alpha_{t+1}^{(1)} = \alpha_t^{(1)} + \eta_t^{(1)}, & (t = 2001, \dots, 4000), \\ \alpha_{t+1}^{(2)} = \alpha_t^{(2)} + \eta_t^{(2)}, \end{cases}$$

where y_t is $d \times 1$ vector, $Z_t^{(1)}$ is $d \times 4$ matrix, $Z_t^{(2)}$ is $d \times (p-4)$ matrix, $\alpha_t^{(1)}$ is 4×1 vector, and $\alpha_t^{(2)}$ is $(p-4) \times 1$ vector, respectively. This model changes the number of nonzero variables at $t = 2000$. $Z_{t,j}^{(i)}$, ε_t, $\eta_t^{(1)}$, and $\eta_t^{(2)}$ are generated independently from $Unif(-5, 5)$, $N(0, \frac{1}{2}I_d + \frac{1}{2}\mathbf{1}_d\mathbf{1}_d^T)$, $N(0, I_4)$, and $N(0, I_{p-4})$, respectively. The initial values of $\alpha_1^{(1)}$ is 0. For the simulation data generated with the above settings, we compare the results of the estimations of the $p \times 1$ state vector $\alpha_t := (\alpha_t^{(1)T}, \alpha_t^{(2)T})^T$ by the Kalman filter, ART-KF, and TART-KF. The required parameters are set as $a_1 = 0$, $P_1 = 1000 \times I_p$, $\Sigma_\varepsilon = \frac{1}{2}I_d + \frac{1}{2}\mathbf{1}_d\mathbf{1}_d^T$, $\Sigma_\eta = I_p$, $S = 3$, $\delta_{AR} = 10^{-8}$, $\delta_g = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\delta_v = 10^{-8}$, $\gamma = 0.002$, $c_e = 100$, and $c_F = 5$. The dimensions of y_t and α_t are changed as $d = 1, 20$ and $p = 10, 20$. The initial value of λ and d_F in (3.3) are changed as

$(\lambda_1, d_F) = (0.05, 0.6), (0.02, 1), (1, 10), (0.2, 10)$ for $(d, p) = (1, 10), (1, 20), (20, 10), (20, 20)$, and in ART-KF, we set $\lambda = \lambda_1$.

The simulations were conducted 100 times. Table 4.1 shows the mean computation time for the Kalman filter, the ART-KF, and TART-KF. The Kalman filter is the fastest. However, our methods are not significantly slower. Figure 4.3 shows the MSE for the estimation of α_t at each time. For all case of p and d, the Kalman filter works well on the interval $t = 2001, \ldots, 4000$ where all variables are non-zero but not on $t = 1, \ldots, 2000$ where the simulation model is sparse. In contrast, the TART-KF is well adapted to both intervals. For $d = 20$, the ART-KF is also well adapted to both intervals. Whereas, for $d = 1$, the ART-KF does not works well in the interval $t = 2001, \ldots, 4000$. Figure 4.4 shows the mean of λ_t obtained by the TART-KF for each time. For $d = 1$, λ_t is adjusted to be large in the first half where the model is sparse and small in the second half where the model is nonsparse, thus TART-KF achieve good estimation accuracy in both intervals. For $d = 20$, although lambda continues to take large values, TART-KF works well. Thus, good accuracy is obtained without changing λ in ART-KF.

Table 4.1 : Comparison of computation time

	$(d, p) = (1, 10)$	$(d, p) = (1, 20)$	$(d, p) = (20, 10)$	$(d, p) = (20, 20)$
Kalman filter	0.011 sec	0.019 sec	0.060 sec	0.073 sec
ART-KF	0.072 sec	0.157 sec	0.168 sec	0.260 sec
TART-KF	0.111 sec	0.238 sec	0.210 sec	0.355 sec

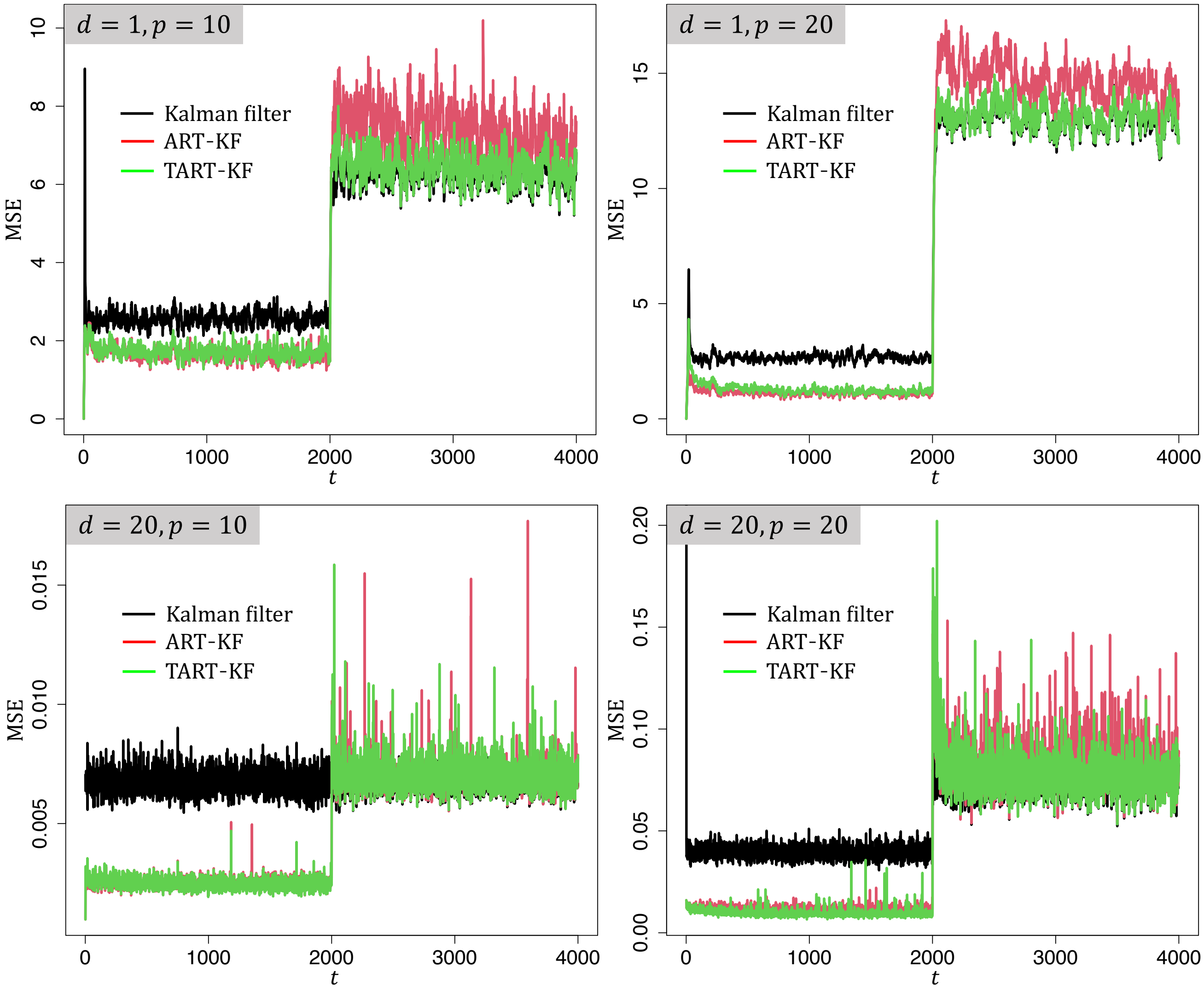

Figure 4.3 : Comparison of MSE between three methods

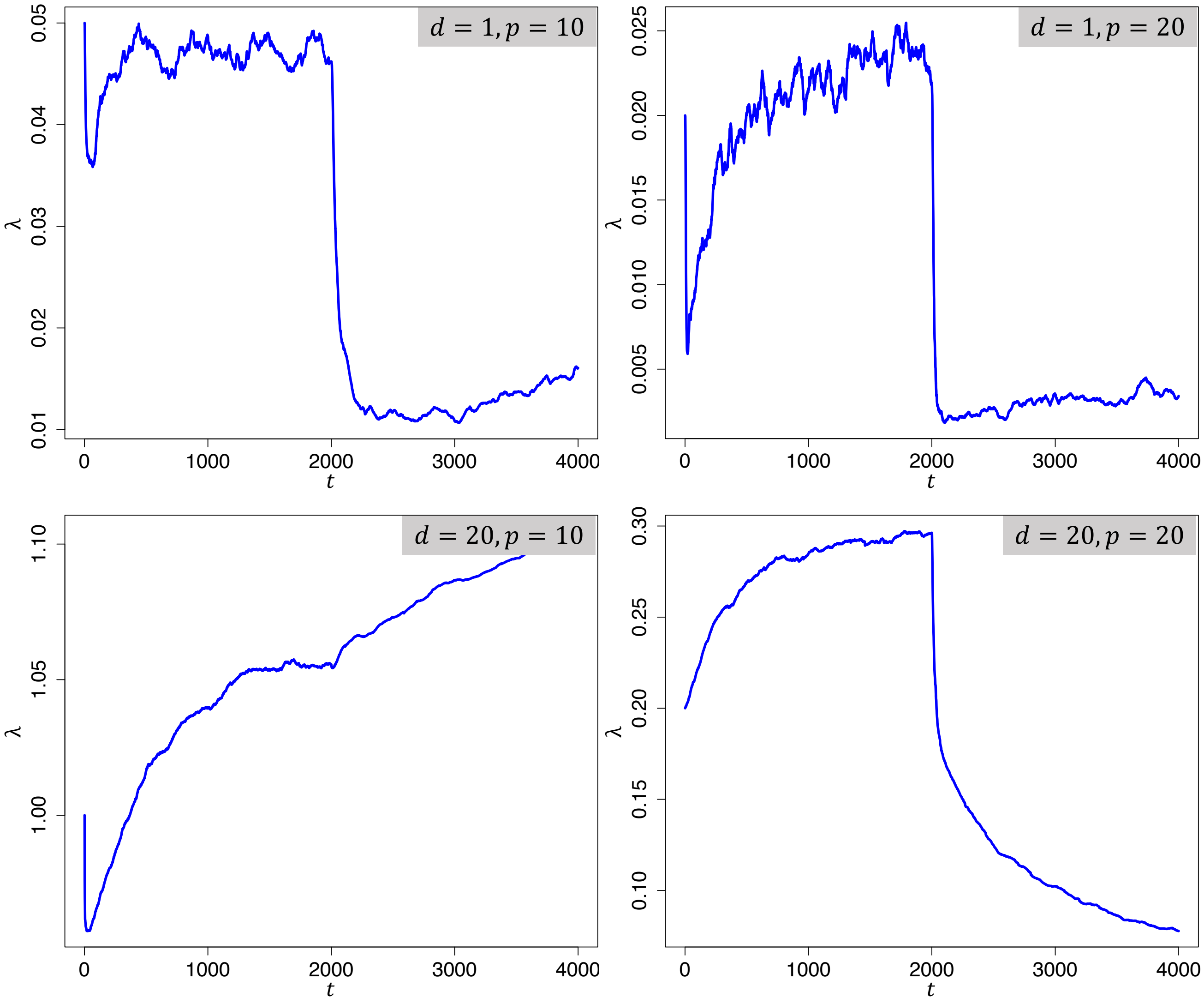

Figure 4.4 : Mean of λ_t for each time

Appendix

A Proof of theorem 1.1

In the linear-Gaussian state-space model (1.1), the following equality holds for the conditional distribution of α_t.

$$-\log f(\alpha_t|y_1,\ldots,y_t) = \frac{1}{2}(\alpha_t - a_{t|t}^{[KF]})^T P_{t|t}^{-1}(\alpha_t - a_{t|t}^{[KF]}) + const.$$

Hence,

$$\frac{\partial}{\partial \alpha_t}\Big\{-\log f(\alpha_t|y_1,\ldots,y_t)+\frac{1}{2}\lambda\|\alpha_t\|_2^2\Big\}=\frac{\partial}{\partial \alpha_t}\Big\{\frac{1}{2}(\alpha_t-a_{t|t}^{[KF]})^T P_{t|t}^{-1}(\alpha_t-a_{t|t}^{[KF]})+\frac{1}{2}\lambda\|\alpha_t\|_2^2\Big\}$$
$$=P_{t|t}^{-1}(\alpha_t-a_{t|t}^{[KF]})+\lambda P_{t|t}^{-1}P_{t|t}\alpha_t=P_{t|t}^{-1}\Big\{\alpha_t-a_{t|t}^{[KF]}+\lambda(Z_t^T\Sigma_\varepsilon^{-1}Z_t+P_t^{-1})^{-1}\alpha_t\Big\}.$$

Therefore, it holds that

$$\begin{aligned}
&\alpha_t=\arg\min_{\alpha_t}\Big\{-\log f(\alpha_t|y_1,\ldots,y_t)+\frac{1}{2}\lambda\|\alpha_t\|_2^2\Big\}\\
\iff\ & P_{t|t}^{-1}\Big\{\alpha_t-a_{t|t}^{[KF]}+\lambda(Z_t^T\Sigma_\varepsilon^{-1}Z_t+P_t^{-1})^{-1}\alpha_t\Big\}=0\\
\iff\ & \Big\{I_p+\lambda(Z_t^T\Sigma_\varepsilon^{-1}Z_t+P_t^{-1})^{-1}\Big\}\alpha_t=a_{t|t}^{[KF]}\\
\iff\ & (Z_t^T\Sigma_\varepsilon^{-1}Z_t+P_t^{-1})^{-1}(Z_t^T\Sigma_\varepsilon^{-1}Z_t+P_t^{-1}+\lambda I_p)\alpha_t=a_{t|t}^{[KF]}\\
\iff\ & \alpha_t=(Z_t^T\Sigma_\varepsilon^{-1}Z_t+P_t^{-1}+\lambda I_p)^{-1}(Z_t^T\Sigma_\varepsilon^{-1}Z_t+P_t^{-1})a_{t|t}^{[KF]}=a_{t|t}^{[RT]}(\lambda).
\end{aligned}$$

For the derivation of the last equality, we used (1.2).

B Lasso type Kalman filter and some discussion

We applied the adaptive ridge to the Kalman filter in ART-KF and TART-KF. It is also possible to apply the lasso regression. Specifically, the lasso estimator is obtained by the following minimization problem:

$$a_{t|t}^{[lasso]}:=\arg\min_{\alpha_t}\Big\{(\alpha_t-a_{t|t}^{[KF]})^T P_{t|t}^{-1}(\alpha_t-a_{t|t}^{[KF]})+\lambda\|\alpha_t\|_1\Big\}. \tag{B.1}$$

From this equation, $a_{t|t}^{[lasso]}$ can be regarded as an estimator of the lasso regression when the design matrix is $P_{t|t}^{-\frac{1}{2}}$ and the objective variable vector is $P_{t|t}^{-\frac{1}{2}}a_{t|t}^{[KF]}$. The regularized parameter λ can be dynamically optimized by the same method used in the TART-KF. In addition, as in TART-KF, we can consider using the value of the lasso estimator at the previous time point instead of $a_{t|t}^{[KF]}$ in (B.1):

$$\begin{aligned}
&a_{t|t}^{[lasso]'}=\arg\min_{\alpha_t}\Big\{(\alpha_t-a_{t|t}^{[lasso,0]'})^T P_{t|t}^{-1}(\alpha_t-a_{t|t}^{[lasso,0]'})+\lambda\|\alpha_t\|_1\Big\},\\
&\text{where } a_{t|t}^{[lasso,0]'}:=a_t^{[lasso]'}+K_t(y_t-Z_ta_t^{[lasso]'}),\ \text{and } a_t^{[lasso]'}:=T_{t-1}a_{t-1|t-1}^{[lasso]'}.
\end{aligned}$$

However, unlike the adaptive ridge (more precisely, the adaptive ridge for $S\to\infty$), the lasso does not have the oracle property, and thus biases are accumulated in this approach. Therefore, it would be appropriate to obtain the estimator by using the equation (B.1).

In fact, we compare the accuracy of the lasso estimators $a_{t|t}^{[lasso]}$, $a_{t|t}^{[lasso]'}$ by the same simulation in Section 4.2. The dimensions of y_t and α_t are set to $d=1$ and $p=20$. The required parameters are set as $a_1=0$, $P_1=1000\times I_p$, $\Sigma_\varepsilon=\frac{1}{2}I_d+\frac{1}{2}\mathbf{1}_d\mathbf{1}_d^T$, $\Sigma_\eta=I_p$, $S=3$, $\delta_g=0.01$, $\beta_1=0.9$,

$\beta_2 = 0.999$, $\delta_v = 10^{-8}$, $\gamma = 0.0002$, $c_e = 100$, $c_F = 5$, $d_F = 2$. The initial value of λ is changed as $\lambda_1 = 0.005, 0.001$ for $a_{t|t}^{[lasso]}$ and $a_{t|t}^{[lasso]'}$. For fixed λ, we set $\lambda = \lambda_1$.

The simulations were conducted 100 times. Figure B.1 shows the MSE for the estimation of α_t at each time for fixed and moving λ. In both cases, the lasso estimation achieves better accuracy than the Kalman filter in the first half where the model is sparse. The lasso estimator $a_{t|t}^{[lasso]}$ using $a_{t|t}^{[KF]}$ is better accurate than $a_{t|t}^{[lasso]'}$. Figure B.1 shows the mean of λ_t at each time for $a_{t|t}^{[lasso]}$ and $a_{t|t}^{[lasso]'}$. As in the TART-KF, large values are taken in the first half and small values are taken in the second half where the model is nonsparse. With this adjustment, the accuracy is better when the λ is moved than when it is fixed.

The simulation results confirm the effect of sparse estimation also in the case of applying lasso. However, ART-KF and TART-KF are more accurate, and the computation time for the ART-KF and the TART-KF is within 1 second, while that for the lasso takes several tens of seconds. For these reasons, we currently use the adaptive ridge.

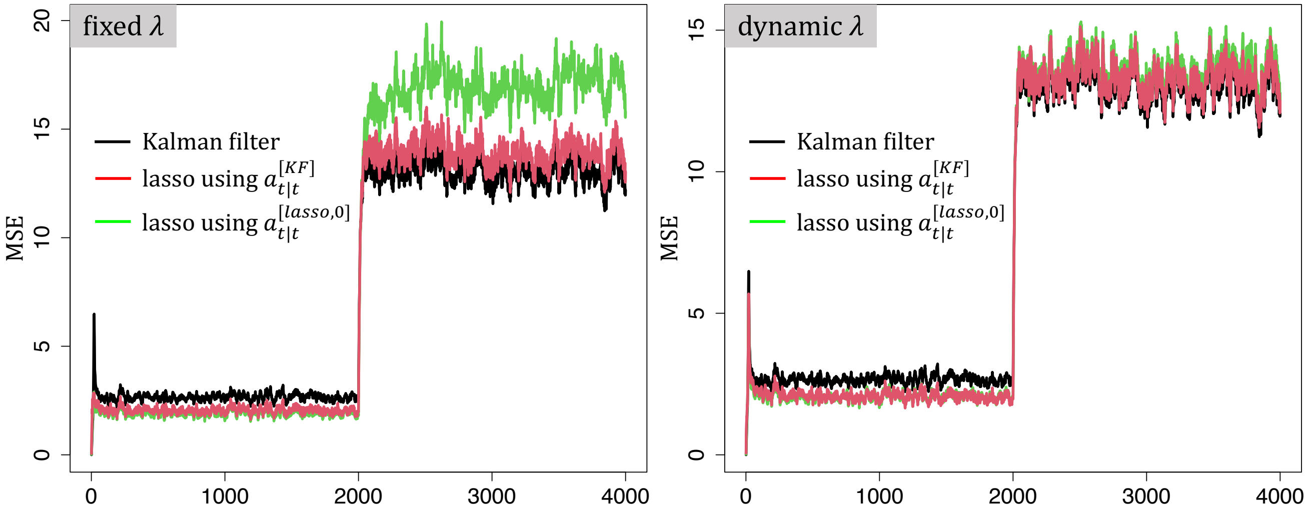

Figure B.1 : Comparison of MSE

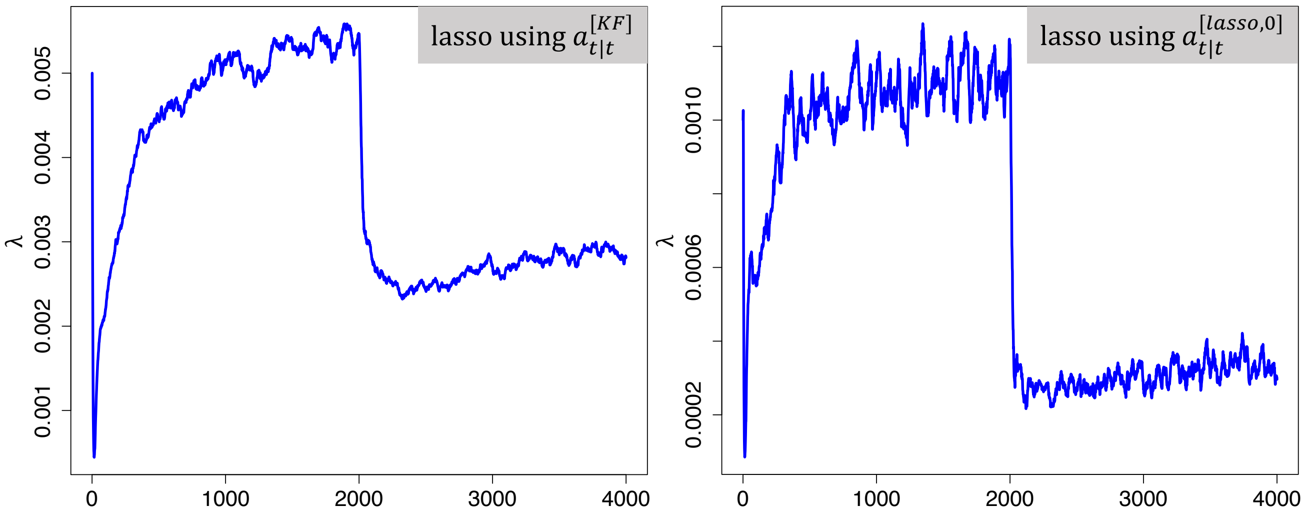

Figure B.2 : Mean of λ_t for each time

We discussed whether to use $a_{t|t}^{[KF]}$ in the lasso estimation (B.1). Furthermore, we can consider using $a_{t|t}^{[KF]}$ instead of $a_t^{[ART,S]}$ in the adaptive ridge estimation (2.1). We compare the accuracy of these approaches by conducting the same simulation as that of Lasso. The parameters set in ART-KF and TART-KF using $a_t^{[ART,S]}$ are the same as in Section 4.2. For the case where $a_{t|t}^{[KF]}$ is used, we set $\lambda_1 = 0.15$, $d_F = 0.8$, and all other parameters are unchanged.

Figure B.3 shows the MSE for the estimation of α_t at each time for fixed and moving λ. Better accuracy is obtained by using $a_t^{[ART,S]}$ for both the ART-KF and TART-KF. This is different results from when the lasso is applied. The adaptive ridge has the oracle property and does not accumulate biases as the lasso does, thus it would be more accurate by reflecting the information from the previous estimators.

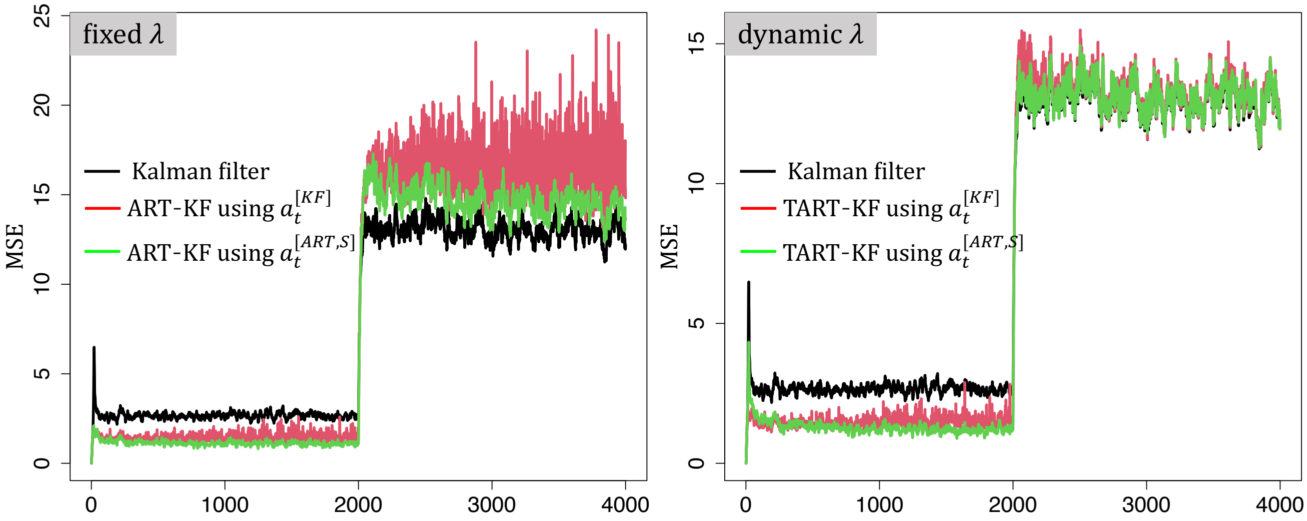

Figure B.3 : Comparison of MSE

We discussed whether to use $a_{t|t}^{[KF]}$ or $a_t^{[ART,S]}$, and it is also possible to take a weighted average of the two. Empirically, our methods are more accurate when $a_t^{[ART,S]}$ is used without taking the average. However, we have not studied this topic in detail, and this is a future work.

References

Linlin Dai, Kani Chen, Zhihua Sun, Zhenqiu Liu, and Gang Li. Broken adaptive ridge regression and its asymptotic properties. *Journal of multivariate analysis*, 168:334–351, 2018.

Ka Long Keith Ho and Hiroki Masuda. Adaptive ridge approach to heteroscedastic regression. *arXiv preprint arXiv:2402.13642*, 2024.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yongming Li, Qingming Gui, Yongwei Gu, Songhui Han, Kai Du, et al. Ridge-type kalman filter and its algorithm. *WSEAS Transactions on Mathematics*, 13:852–862, 2014.