The Rank-Reduced Kalman Filter: Approximate Dynamical-Low-Rank Filtering In High Dimensions

Appendix

A Proofs

A.1 Proof of Proposition 1

Proof. By standard results on Gaussian conditioning and marginalization

$$\begin{aligned} \mathrm{z} \mid \mathrm{y} &\sim \mathcal{N}(\mathrm{K_z}(\mathrm{y} - \mathrm{C}\mu), \Sigma_\mathrm{z}), &\text{(A.1a)}\\ \mathrm{y} &\sim \mathcal{N}(\mathrm{C}\mu, \mathrm{S}), &\text{(A.1b)} \end{aligned}$$

where

$$\begin{aligned} \Sigma_\mathrm{z}^{-1} &= \mathrm{I} + (\mathrm{R}^{-1/2}\mathrm{C}\Pi^{1/2})^*\mathrm{R}^{-1/2}C\Pi^{1/2}, &\text{(A.2a)}\\ \mathrm{K_z} &= \Sigma_z(\mathrm{R}^{-1/2}\mathrm{C}\Pi^{1/2})^*\mathrm{R}^{-1/2}, &\text{(A.2b)}\\ \mathrm{R}^{-1/2}\mathrm{S}\mathrm{R}^{-*/2} &= \mathrm{R}^{-1/2}\mathrm{C}\Pi^{1/2}(\mathrm{R}^{-1/2}\mathrm{C}\Pi^{1/2})^* + \mathrm{I}. &\text{(A.2c)} \end{aligned}$$

Substituting for the singular value decomposition of $(\mathrm{R}^{-1/2}\mathrm{C}\Pi^{1/2})^*$ gives the conditional covariance

$$\Sigma_\mathrm{z} = \big(\mathrm{I} + \mathrm{U}\mathrm{D}^2\mathrm{U}^*\big)^{-1} = \big(\mathrm{U}(\mathrm{I} + \mathrm{D}^2)\mathrm{U}^*\big)^{-1} = \mathrm{U}(\mathrm{I} + \mathrm{D}^2)^{-1}\mathrm{U}^*, \tag{A.3}$$

the Kalman gain

$$\mathrm{K_z} = \mathrm{U}(\mathrm{I} + \mathrm{D}^2)^{-1}\mathrm{U}^*\mathrm{U}\mathrm{D}\mathrm{V}^*\mathrm{R}^{-1/2} = \mathrm{U}(\mathrm{I} + \mathrm{D}^2)^{-1}\mathrm{D}\mathrm{V}^*\mathrm{R}^{-1/2}, \tag{A.4}$$

and the marginal measurement covariance matrix

$$\mathrm{R}^{-1/2}\mathrm{S}\mathrm{R}^{-*/2} = \mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I}. \tag{A.5}$$

This gives the result on marginalization and conditioning. To obtain the expression for the marginal likelihood, which is given by

$$\mathcal{N}(\mathrm{y}; \mathrm{C}\mu, \mathrm{S}) = -\frac{m}{2}\log 2\pi - \frac{1}{2}\log|\mathrm{S}| - \frac{1}{2}(\mathrm{y} - \mathrm{C}\mu)^*\mathrm{S}^{-1}(\mathrm{y} - \mathrm{C}\mu). \tag{A.6}$$

The log determinant is given by

$$\log|\mathrm{S}| = \log\Big|\mathrm{R}^{1/2}(\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I})\mathrm{R}^{*/2}\Big| = 2\log|\mathrm{R}^{1/2}| + \log|\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I}|\,. \tag{A.7}$$

Let V_k be the k-th column vector of V then it is an eigenvector of $\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I}$ with eigenvalue $\mathrm{D}_{kk}^2 + 1$. Furthermore, any vector in the orthogonal complement to the column space of V is also an eigenvector with eigenvalue 1. Therefore,

$$\log|\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I}| = \sum_{k=1}^{r}\log(\mathrm{D}_{kk}^2 + 1). \tag{A.8}$$

It remains to obtain the desired expression for the quadratic form. Start by inserting the expression for S

$$\begin{aligned} (\mathrm{y} - \mathrm{C}\mu)^*\mathrm{S}^{-1}(\mathrm{y} - \mathrm{C}\mu) &= (\mathrm{y} - \mathrm{C}\mu)^*\mathrm{R}^{-*/2}(\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I})^{-1}\mathrm{R}^{-1/2}(\mathrm{y} - \mathrm{C}\mu)\\ &= \mathrm{e}^*(\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I})^{-1}\mathrm{e}. \end{aligned} \tag{A.9}$$

Now $\mathrm{e}^*(\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I})^{-1}\mathrm{e} = \mathrm{e}^*\mathrm{b}$, where we define the vector b such that

$$(\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I})\mathrm{b} = \mathrm{e}. \tag{A.10}$$

Multiplying from the left by V^* gives

$$(\mathrm{D}^2 + \mathrm{I})\mathrm{V}^*\mathrm{b} = \mathrm{V}^*\mathrm{e}, \tag{A.11}$$

therefore,

$$\mathrm{V}^*\mathrm{b} = (\mathrm{D}^2 + \mathrm{I})^{-1}\mathrm{V}^*\mathrm{e}. \tag{A.12}$$

Inserting this in the original definition of b gives

$$\mathrm{V}\mathrm{D}^2(\mathrm{D}^2 + \mathrm{I})^{-1}\mathrm{V}^*\mathrm{e} + \mathrm{b} = \mathrm{e}, \tag{A.13}$$

from which it follows that

$$\mathrm{e}^*(\mathrm{V}\mathrm{D}^2\mathrm{V}^* + \mathrm{I})^{-1}\mathrm{e} = \mathrm{e}^*\mathrm{b} = \|\mathrm{e}\|^2 - \mathrm{e}^*\mathrm{V}\mathrm{D}(\mathrm{D}^2 + \mathrm{I})^{-1}\mathrm{D}\mathrm{V}^*\mathrm{e}. \tag{A.14}$$

□

A.2 Proof of Proposition 2

In the following, the computational complexity of the low-rank filtering recursion is analyzed in detail, which proves Proposition 2 and Corollary 2. In the best case, the cost of the proposed filtering algorithms scales linearly in the state dimension n and the measurement dimension m. For this to hold, we assume that

(a) the maps $\mathrm{x} \mapsto \mathrm{A}\mathrm{x}$, $\mathrm{x} \mapsto \Phi\mathrm{x}$, and $\mathrm{x} \mapsto \mathrm{B}\mathrm{B}^*\mathrm{x}$ can be evaluated in $\mathcal{O}(n)$,

(b) the map $\mathrm{x} \mapsto \mathrm{C}\mathrm{x}$ can be evaluated in $\mathcal{O}(m)$, and

(c) the map $\mathrm{x} \mapsto \mathrm{R}^{-1/2}\mathrm{x}$ and the log-determinant $\log|\mathrm{R}^{1/2}|$ can be evaluated in $\mathcal{O}(m)$.

We refer to the situation in which (a) or (b) do not apply as the "worst case". Assumption (c) is taken for granted as the measurement-noise covariance R is often a diagonal matrix, which implies that sensor errors are uncorrelated. This is not only realistic but also commonly imposed in modelling.

Proof. The best-case cost of the approximate filtering scales linear in the state dimension n and the measurement dimension m. In the worst case, the complexity scales quadratically in n and m.

Prediction We begin by analyzing the cost for the approximate integration of the low-rank process-noise covariance $\mathrm{Q}^{1/2}$. This amounts to the cost of the dynamical low-rank approximation (DLRA) algorithm for a symmetric Lyapunov equation. Appendix C gives a detailed description of how this algorithm is used as part of the RRKF recursions. Let the matrix-valued flow field be $F(\mathrm{Q}) = \mathrm{A}\mathrm{Q} + \mathrm{Q}\mathrm{A}^* + \mathrm{B}\mathrm{B}^*$. Given an initial factorization $\mathrm{Q}_0 \approx \mathrm{Y}_0 = \mathrm{U}_0\mathrm{D}_0^2\mathrm{U}_0^*$, with $\mathrm{U}_0 \in \mathbb{R}^{n\times r}$ and $\mathrm{D}_0 \in \mathbb{R}^{r\times r}$, the cost for integrating this matrix equation using DLRA is as follows:

1. **K-step**

   (a) Flow-field evaluation

$$F(t, \mathrm{K}(t)\mathrm{U}_0^*)\mathrm{U}_0 = \mathrm{A}(\mathrm{K}(t)\mathrm{U}_0^*)\mathrm{U}_0 + \mathrm{K}(t)\mathrm{U}_0^*\mathrm{A}^*\mathrm{U}_0 + \mathrm{B}\mathrm{B}^*\mathrm{U}_0 \tag{A.15a}$$
$$= \mathrm{A}\mathrm{K}(t) + \mathrm{K}(t)(\mathrm{U}_0^*(\mathrm{A}^*\mathrm{U}_0)) + \mathrm{B}\mathrm{B}^*\mathrm{U}_0, \tag{A.15b}$$

   (b) QR factorization of tall $n \times r$ matrix

$$\mathrm{QR}\left(\mathrm{K}(t+h)\right), \tag{A.16}$$

   (c) Compute

$$\mathrm{M} = \mathrm{U}_h^*\mathrm{U}_0. \tag{A.17}$$

2. **S-step**

   (a) Compute initial $\mathrm{D}(t_0)$

$$\mathrm{D}(t_0) = \mathrm{M}\mathrm{D}_0\mathrm{M}^*, \tag{A.18}$$

   (b) Flow-field evaluation

$$\mathrm{U}_h^*F(t, \mathrm{U}_h\mathrm{D}(t)\mathrm{U}_h^*)\mathrm{U}_h = \mathrm{U}_h^*\left(\mathrm{A}\mathrm{U}_h\mathrm{D}(t)\mathrm{U}_h^* + \mathrm{U}_h\mathrm{D}(t)\mathrm{U}_h^*\mathrm{A}^* + \mathrm{B}\mathrm{B}^*\right)\mathrm{U}_h \tag{A.19a}$$
$$= \mathrm{U}_h^*(\mathrm{A}(\mathrm{U}_h\mathrm{D}(t))) + (\mathrm{D}(t)\mathrm{U}_h^*)(\mathrm{A}^*\mathrm{U}_h) + \mathrm{U}_h^*(\mathrm{B}\mathrm{B}^*\mathrm{U}_h). \tag{A.19b}$$

Table A.1: Time complexity of reduced-rank filtering: The prediction step

Step			
Eq.	Operation	Best case	Worst case
(A.15)	$\mathrm{AK}(t)$ $\mathrm{K}(t)(\mathrm{U}_0^*(\mathrm{A}^*\mathrm{U}_0))$ $\mathrm{BB}^*\mathrm{U}_0$	$\mathcal{O}(nr)$ $\mathcal{O}(nr^2)$ $\mathcal{O}(nr)$	$\mathcal{O}(n^2r)$ $\mathcal{O}(n^2r+nr^2)$ $\mathcal{O}(n^2r)$
(A.16)	$\mathrm{QR}\big(\mathrm{K}(t+h)\big)$	$\mathcal{O}(nr^2)$	$\mathcal{O}(nr^2)$
(A.17)	$\mathrm{M}=\mathrm{U}_h^*\mathrm{U}_0$	$\mathcal{O}(nr^2)$	$\mathcal{O}(nr^2)$
(A.18)	$\mathrm{D}(t_0)=\mathrm{MD}_0\mathrm{M}^*$	$\mathcal{O}(r^3)$	$\mathcal{O}(r^3)$
(A.19)	$\mathrm{U}_h^*(\mathrm{A}(\mathrm{U}_h\mathrm{D}(t)))$ $(\mathrm{D}(t)\mathrm{U}_h^*)(\mathrm{A}^*\mathrm{U}_h)$ $\mathrm{U}_h^*(\mathrm{BB}^*\mathrm{U}_h)$	$\mathcal{O}(nr^2)$ $\mathcal{O}(nr^2)$ $\mathcal{O}(nr^2)$	$\mathcal{O}(n^2r+nr^2)$ $\mathcal{O}(n^2r+nr^2)$ $\mathcal{O}(n^2r+nr^2)$
(A.20)	$\mathrm{D}_l=\mathrm{D}_l^{1/2}\mathrm{D}_l^{*/2}$	$\mathcal{O}(r^3)$	$\mathcal{O}(r^3)$
(A.21)	$\Phi_l\mu_{l-1}$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$
(A.22)	$\Phi_l\Sigma_{l-1}^{1/2}$ $(\Phi_l\Sigma_{l-1}^{1/2}\quad \mathrm{Q}_l^{1/2})\approx\tilde{\mathrm{U}}_l\tilde{\mathrm{D}}_l\tilde{\mathrm{V}}_l^*$	$\mathcal{O}(nr)$ $\mathcal{O}(nr^2)$	$\mathcal{O}(n^2r)$ $\mathcal{O}(nr^2)$

The above steps 1. and 2. are repeated according to how many DLRA integration steps are performed in the respective prediction step. Therefore, their cost has to be multiplied by that (typically small) constant.

3. Using DLRA integration we obtain a low-rank factorization of the process-noise covariance matrix $\mathrm{Q}_l\approx\mathrm{Y}_l=\mathrm{U}_l\mathrm{D}_l\mathrm{U}_l^{?}$. It remains to compute a matrix square root $\mathrm{Q}_l^{1/2}=\mathrm{U}_l\mathrm{D}_l^{1/2}$ and thus a matrix square root of $\mathrm{D}_l\in\mathbb{R}^{r\times r}$

$$\mathrm{D}_l=\mathrm{D}_l^{1/2}\mathrm{D}_l^{*/2}. \tag{A.20}$$

Now, having obtained $\mathrm{Q}_l^{1/2}$, we proceed to

4. predict the mean

$$\mu_l^-=\Phi_l\mu_{l-1}, \tag{A.21}$$

5. and the low-rank factor $\Pi_l^{1/2}$ of the predicted covariance matrix. This amounts to building the rank-$2r$ square-root factor and truncating it at its r-th largest singular value (i.e., a truncated SVD of a tall $n\times 2r$ matrix):

$$\Big(\Phi_l\Sigma_{l-1}^{1/2}\quad \mathrm{Q}_l^{1/2}\Big)\approx\tilde{\mathrm{U}}_l\tilde{\mathrm{D}}_l\tilde{\mathrm{V}}_l^*. \tag{A.22}$$

The computational complexities of the respective steps can be found in Table A.1.

Correction step Now, the cost of the correction step is analyzed. First, we compute an SVD of a wide $r\times m$ matrix

$$(\mathrm{R}^{-1/2}\mathrm{C}\Pi_l^{1/2})^*=\mathrm{U}_l\mathrm{D}_l\mathrm{V}_l^*. \tag{A.23}$$

Given this decomposition and the whitened residual

$$\mathrm{e}_l=\mathrm{R}^{-1/2}(\mathrm{y}_l-\mathrm{C}\mu_l^-), \tag{A.24}$$

we proceed to

1. update the mean

$$\Delta\mu_l=\mathrm{K}_l\mathrm{e}_l=\Pi_l^{1/2}(\mathrm{U}_l((\mathrm{I}+\mathrm{D}_l^2)^{-1}(\mathrm{D}_l(\mathrm{V}_l^*\mathrm{e}_l)))), \tag{A.25}$$

Table A.2: Time complexity of reduced-rank filtering: The correction step

Step			
Eq.	Operation	Best case	Worst case
(A.23)	$R^{-1/2}(C\Pi_l^{1/2})$	$\mathcal{O}(mr)$	$\mathcal{O}(m^2 r)$
	$(R^{-1/2}C\Pi_l^{1/2})^* = U_l D_l V_l^*$	$\mathcal{O}(mr^2)$	$\mathcal{O}(mr^2)$
(A.24)	$e_l = R^{-1/2}(y_l - C\mu_l^-)$	$\mathcal{O}(m)$	$\mathcal{O}(mn + m^2)$
(A.25)	$\Pi_l^{1/2}(U_l((I + D_l^2)^{-1}(D_l(V_l^* e_l))))$	$\mathcal{O}(mr + nr^2)$	$\mathcal{O}(mr + nr^2)$
(A.26)	$\Pi_l^{1/2}(U_l(I + D_l^2)^{-1/2})$	$\mathcal{O}(nr^2)$	$\mathcal{O}(nr^2)$
(23)	$\log \lvert R^{1/2} \rvert$	$\mathcal{O}(m)$	$\mathcal{O}(m)$
	$\sum_{k=1}^{r} \log((D_l)_{kk}^2 + 1)$	$\mathcal{O}(r)$	$\mathcal{O}(r)$
	$\lVert e_l \rVert^2$	$\mathcal{O}(r)$	$\mathcal{O}(r)$
	$(e_l^* V_l)(D_l((D_l^2 + I)^{-1} D_l))(V_l^* e_l)$	$\mathcal{O}(mr)$	$\mathcal{O}(mr)$

2. and compute the low-rank factor of the filtering covariance

$$\Sigma_l^{1/2} = \Pi_l^{1/2}(U_l(I + D_l^2)^{-1/2}). \tag{A.26}$$

Finally, the marginal predictive log-likelihood $\log p(y_l \mid y_{1:l-1})$ is computed as in Eq. (23).

The computational complexities of the respective steps can be found in Table A.2. Together, Tables A.1 and A.2 prove the stated asymptotic complexities of the low-rank filtering recursion. □

A.3 Proof of Proposition 3

This section analyzes the computational complexity of the low-rank smoothing recursion in detail, which proves Proposition 3.

Proof. The best-case cost of the approximate smoothing scales linear in the state dimension n. In the worst case, the complexity is quadratic in n.

Approximate backwards kernel Factorizations of $\Sigma_{l-1}^{1/2}$ and $\Pi_l^{1/2}$ are already given from filtering. For the smoothing gain, it remains to compute

$$\Gamma_l = \Sigma_l^{*/2}(\Phi_{l+1}^*(\Pi_{l+1}^{1/2})^+). \tag{A.27}$$

Then, we proceed to compute the shift vector

$$v_l = \mu_l - G_l \mu_{l+1}^- = \mu_l - \Sigma_l^{1/2}(\Gamma_l((\Pi_{l+1}^{*/2})^+ \mu_{l+1}^-)), \tag{A.28}$$

and the low-rank factor $P_l^{1/2}$ of the backwards-process-noise covariance matrix. Therefore, a truncated SVD of a tall $n \times 2r$ matrix is computed:

$$\Big((I - G_l \Phi_{l+1})\Sigma_l^{1/2} \quad G_l Q_{l+1}^{1/2} \Big) \approx \widehat{U}_l \widehat{D}_l \widehat{V}_l^*. \tag{A.29}$$

Backwards prediction (smoothing) Smoothing amounts to consecutive predictions with the backwards kernel. For this prediction, the low-rank factor of the process-noise covariance matrix does not come from DLRA, but can be directly computed from the smoothing gain and the process-noise covariance matrix (see Eqs. (29) and (30)).

The smoothing mean is therefore given as

$$\xi_l = G_l \xi_{l+1} + v_l = \Sigma_l^{1/2}(\Gamma_l((\Pi_{l+1}^{*/2})^+ \xi_{l+1})) + v_l, \tag{A.30}$$

Table A.3: Time complexity of reduced-rank smoothing

Step			
Eq.	Operation	Best case	Worst case
(A.27)	$\Gamma_l = \Sigma_{l-1}^{*/2}(\Phi_l^*(\Pi_l^{1/2})^+)$	$\mathcal{O}(nr^2)$	$\mathcal{O}(n^2r + nr^2)$
(A.28)	$\mathrm{G}_l\mu_{l+1}^- = \Sigma_l^{1/2}(\Gamma_l((\Pi_{l+1}^{*/2})^+\mu_{l+1}^-))$	$\mathcal{O}(nr + r^2)$	$\mathcal{O}(nr + r^2)$
(A.29)	$\mathrm{G}_l\Phi_{l+1}\Sigma_l^{1/2} = \Sigma_l^{1/2}(\Gamma_l((\Pi_{l+1}^{*/2})^+(\Phi_{l+1}\Sigma_l^{1/2})))$	$\mathcal{O}(nr^2 + r^3)$	$\mathcal{O}(n^2r + nr^2 + r^3)$
	$\mathrm{G}_l\mathrm{Q}_l^{1/2} = \Sigma_{l-1}^{1/2}(\Gamma_l((\Pi_l^{*/2})^+\mathrm{Q}_l^{1/2}))$	$\mathcal{O}(nr^2 + r^3)$	$\mathcal{O}(nr^2 + r^3)$
	$((\mathrm{I} - \mathrm{G}_l\Phi_{l+1})\Sigma_l^{1/2} \quad \mathrm{G}_l\mathrm{Q}_{l+1}^{1/2}) \approx \widehat{\mathrm{U}}_l\widehat{\mathrm{D}}_l\widehat{\mathrm{V}}_l^*$	$\mathcal{O}(nr^2)$	$\mathcal{O}(nr^2)$
(A.30)	$\mathrm{G}_l\xi_l = \Sigma_{l-1}^{1/2}(\Gamma_l((\Pi_l^{*/2})^+\xi_l))$	$\mathcal{O}(nr + r^2)$	$\mathcal{O}(nr + r^2)$
(A.31)	$\mathrm{G}_l\Lambda_{l+1}^{1/2} = \Sigma_l^{1/2}(\Gamma_l((\Pi_{l+1}^{*/2})^+\Lambda_{l+1}^{1/2}))$	$\mathcal{O}(nr^2 + r^3)$	$\mathcal{O}(nr^2 + r^3)$
	$(\mathrm{G}_l\Lambda_{l+1}^{1/2} \quad \mathrm{P}_l^{1/2}) \approx \bar{\mathrm{U}}_l\bar{\mathrm{D}}_l\bar{\mathrm{V}}_l^*$	$\mathcal{O}(nr^2)$	$\mathcal{O}(nr^2)$

and the low-rank factor of the smoothing covariance as $\Lambda_l^{1/2} = \bar{\mathrm{U}}_l\bar{\mathrm{D}}_l$, where

$$\begin{pmatrix}\mathrm{G}_l\Lambda_{l+1}^{1/2} & \mathrm{P}_l^{1/2}\end{pmatrix} \approx \bar{\mathrm{U}}_l\bar{\mathrm{D}}_l\bar{\mathrm{V}}_l^* \tag{A.31}$$

is the truncated SVD of a tall $n \times 2r$ matrix.

The computational complexities of the respective steps can be found in Table A.3, which shows that—in the best case—the cost of approximate low-rank smoothing never exceeds $\mathcal{O}(nr^2 + r^3)$, as stated by Proposition 3. It also shows that in the worst case, the complexity is quadratic in n. □

B Efficient inference for the case r > m

This section gives an inference scheme that can be used in the correction step in case the low-rank dimension r exceeds the measurement dimension m at a time point t_l. This is not generally the designated use case of the algorithm, since in most applications, we assume $r \ll m \leq n$. However, if at a given time point t_l, there are less than usual measurements available (e.g. due to missing data at that time), it is still useful to have an inference scheme in place for this case. This is provided by the following proposition.

Proposition B.1. *Let $m < r \leq n$ and $\Pi_l^{1/2} \in \mathbb{R}^{n \times r}$ and $\mathrm{R}^{1/2} \in \mathbb{R}^{m \times m}$. An approximate update is then obtained according to the standard Kalman filter update rules:*

$$\mathrm{S}_l = \mathrm{C}\Pi_l^{1/2}(\mathrm{C}\Pi_l^{1/2})^* + \mathrm{R}^{1/2}\mathrm{R}^{*/2}, \tag{B.1a}$$

$$\mathrm{K}_l = \Pi_l^{1/2}(\mathrm{C}\Pi_l^{1/2})^*\mathrm{S}^+, \tag{B.1b}$$

$$\Delta\mu_l = \mathrm{K}_l(\mathrm{y}_l - \mathrm{C}\mu_l), \tag{B.1c}$$

$$\Sigma_l = \Pi_l - \mathrm{K}_l\mathrm{S}_l\mathrm{K}_l^*. \tag{B.1d}$$

The marginal measurement covariance, S_l, is obtained by a singular value decomposition

$$\mathrm{U}_l^s\mathrm{D}_l^s(\mathrm{V}_l^s)^* = \begin{pmatrix}\mathrm{C}\Pi_l^{1/2} & \mathrm{R}^{1/2}\end{pmatrix}, \tag{B.2a}$$

$$\mathrm{S} = \mathrm{U}_l^s(\mathrm{D}_l^s)^2(\mathrm{U}_l^s)^*, \tag{B.2b}$$

where the matrix on the right-hand side of the first equation is $m \times (r+m)$. Hence, $\mathrm{U}_l^s \in \mathbb{R}^{m \times (r+m)}$, and $\mathrm{D}_l^s, \mathrm{V}_l^s \in \mathbb{R}^{(r+m) \times (r+m)}$. The Moore–Penrose pseudoinverse S_l^+ is then given by

$$\mathrm{S}_l^+ = \mathrm{U}_l^s(\mathrm{D}_l^s)^{-2}(\mathrm{U}_l^s)^*. \tag{B.3}$$

The gain matrix is thus obtained as

$$\widetilde{\mathrm{K}}_l = (\mathrm{C}\Pi_l^{1/2})^*\mathrm{U}^s(\mathrm{D}^s)^{-1}, \tag{B.4a}$$

$$\mathrm{K}_l = \Pi_l^{1/2}\widetilde{\mathrm{K}}_l(\mathrm{D}^s)^{-1}(\mathrm{U}^s)^*, \tag{B.4b}$$

where $\widetilde{\mathrm{K}}_l \in \mathbb{R}^{r\times(r+m)}$. The updated covariance is obtained by

$$\begin{aligned}\Sigma_l &= \Pi_l - \mathrm{K}_l \mathrm{S}_l \mathrm{K}_l^* \\ &= \Pi_l - \Pi_l^{1/2}\widetilde{\mathrm{K}}_l(\mathrm{D}_l^s)^{-1}(\mathrm{U}_l^s)^*\mathrm{U}_l^s\mathrm{D}_l^s(\mathrm{U}_l^s)^*(\Pi_l^{1/2}\widetilde{\mathrm{K}}_l(\mathrm{D}_l^s)^{-1}(\mathrm{U}_l^s)^*)^* \\ &= \Pi_l - \Pi_l^{1/2}\widetilde{\mathrm{K}}_l(\Pi_l^{1/2}\widetilde{\mathrm{K}}_l)^* \\ &= \Pi_l^{1/2}\big(\mathrm{I} - \widetilde{\mathrm{K}}_l\widetilde{\mathrm{K}}_l^*\big)\Pi_l^{*/2}.\end{aligned} \tag{B.5}$$

Consider the singular value decomposition of $\widetilde{\mathrm{K}}_l$

$$\widetilde{\mathrm{K}}_l = \mathrm{U}_l^k\mathrm{D}_l^k(\mathrm{V}_l^k)^*, \tag{B.6}$$

where $\mathrm{U}_l^k, \mathrm{D}_l^k \in \mathbb{R}^{r\times r}$ and $\mathrm{V}_l^k \in \mathbb{R}^{r\times r+m}$. The covariance update in low-rank form is then given by

$$\begin{aligned}\Sigma^{1/2} &= \Pi^{1/2}\big(\mathrm{I} - \mathrm{U}_l^k(\mathrm{D}_l^k)^2(\mathrm{U}_l^k)^*\big)^{1/2}, \\ &= \Pi^{1/2}\big(\mathrm{U}_l^k(\mathrm{U}_l^k)^* - \mathrm{U}_l^k(\mathrm{D}_l^k)^2(\mathrm{U}_l^k)^*\big)^{1/2}, \\ &= \Pi^{1/2}\mathrm{U}_l^k\big(\mathrm{I} - (\mathrm{D}_l^k)^2\big)^{1/2}.\end{aligned} \tag{B.7}$$

The marginal predictive log likelihood $p(\mathrm{y}_l \mid \mathrm{y}_{1:l-1}) = \mathcal{N}(\mathrm{C}\mu_l^-, \mathrm{S}_l)$ can be cheaply evaluated given the factorization of S_l from Eq. (B.2).

C Dynamical-low-rank approximation algorithm for Lyapunov equations

Following Ceruti and Lubich [8], we give the procedure for one DLRA integration step, adapted to our specific case of a symmetric Lyapunov equation

$$F(t, \mathrm{Q}(t)) = \mathrm{A}\mathrm{Q}(t) + \mathrm{Q}(t)\mathrm{A}^* + \mathrm{B}(t)\mathrm{B}(t)^*. \tag{C.1}$$

Let an initial rank-r factorization $\mathrm{Q}_0 \approx \mathrm{Y}_0 = \mathrm{U}_0\mathrm{D}_0\mathrm{U}_0^*$, with an orthogonal matrix $\mathrm{U}_0 \in \mathbb{R}^{n\times r}$ and a symmetric matrix $\mathrm{D}_0 \in \mathbb{R}^{r\times r}$, be given at time t_0. For a temporal step size h, a rank-r factorization $\mathrm{Q}(t+h) \approx \mathrm{Y}_h = \mathrm{U}_h\mathrm{D}_h\mathrm{U}_h^*$ at the next integration step $t_0 + h$ is computed as follows.

K-step: Update $\mathrm{U}_0 \to \mathrm{U}_h$.
Integrate from $t = t_0$ to $t_0 + h$ the $n \times r$ matrix differential equation

$$\dot{\mathrm{K}}(t) = F(t, \mathrm{K}(t)\mathrm{U}_0^*)\mathrm{U}_0, \qquad \mathrm{K}(t_0) = \mathrm{U}_0\mathrm{D}_0. \tag{C.2}$$

Orthogonalize $K(t+h)$ by computing a QR decomposition, yielding the orthogonal matrix U_h. Then, compute the $r \times r$ matrix $\mathrm{M} = \mathrm{U}_h^*\mathrm{U}_0$.

S-step: Update $\mathrm{D}_0 \to \mathrm{D}_h$.
Integrate from $t = t_0$ to $t_0 + h$ the $r \times r$ matrix differential equation

$$\dot{\mathrm{D}}(t) = \mathrm{U}_h^*F(t, \mathrm{U}_h\mathrm{D}(t)\mathrm{U}_h^*)\mathrm{U}_h, \qquad \mathrm{D}(t_0) = \mathrm{M}\mathrm{D}_0\mathrm{M}^*, \tag{C.3}$$

and set $\mathrm{D}_h = \mathrm{D}(t+h)$

Remark C.1. *The terms "K-step" and "S-step" can be confusing due to conflicting notation conventions in the Kalman-filtering and dynamical-low-rank-approximation literature, respectively. The matrix K and the term "S-step" are not related to the Kalman gain K and the marginal measurement covariance matrix S from the Kalman filter update step described in Section 3.2 and Appendix B.*

The $n \times r$ and $r \times r$ matrix differential equations in the K-step and S-step can be solved using a standard numerical integrator, e.g., a Runge-Kutta method or an exponential integrator.

It is useful to note that the matrix equation in the S-step itself is a symmetric Lyapunov equation:

$$\begin{aligned}\dot{\mathrm{D}}(t) &= \mathrm{U}_h^*F(t, \mathrm{U}_h\mathrm{D}(t)\mathrm{U}_h^*)\mathrm{U}_h && \text{(C.4a)} \\ &= \mathrm{U}_h^*\left(\mathrm{A}\mathrm{U}_h\mathrm{D}(t)\mathrm{U}_h^* + \mathrm{U}_h\mathrm{D}(t)\mathrm{U}_h^*\mathrm{A}^* + \mathrm{B}\mathrm{B}^*\right)\mathrm{U}_h && \text{(C.4b)} \\ &= (\mathrm{U}_h^*\mathrm{A}\mathrm{U}_h)\mathrm{D}(t) + \mathrm{D}(t)(\mathrm{U}_h^*\mathrm{A}\mathrm{U}_h)^* + \mathrm{U}_h^*\mathrm{B}\mathrm{B}^*\mathrm{U}_h && \text{(C.4c)} \\ &= \mathrm{A}_\mathrm{D}\mathrm{D}(t) + \mathrm{D}(t)\mathrm{A}_\mathrm{D}^* + \mathrm{B}_\mathrm{D}\mathrm{B}_\mathrm{D}^*, && \text{(C.4d)}\end{aligned}$$

where the final step defines the parameters of the Lyapunov equation for $\mathrm{D}(t)$ as $\mathrm{A}_\mathrm{D} = \mathrm{U}_h^*\mathrm{A}\mathrm{U}_h$ and $\mathrm{B}_\mathrm{D} = \mathrm{U}_h^*\mathrm{B}$. Since $\mathrm{D}(t) \in \mathbb{R}^{r\times r}$ is small, this equation can be solved exactly with little computational cost using matrix-fraction decomposition [40, 3].

D Details on experimental setups

This section gives more details regarding the experimental setups used in Section 4. For all results of the proposed RRKF algorithm, only a single DLRA step was used in the prediction step to compute the low-rank factorization of the process noise covariance matrix $\mathrm{Q}^{1/2}$.

D.1 Linear advection model

This section provides more details on the data assimilation setting that is considered in Section 4.1.

We consider a spatial grid of $n = 1024$ uniformly spaced points and a temporal grid of 800 uniformly spaced time points. Unit step sizes $\Delta t = \Delta x = 1$ are assumed on the spatio-temporal grid. To generate an initial ground-truth state we sample an initial sinusoidal curve $\psi(0)$ according to the description by Sakov and Oke [33]

$$\psi_i(0) = \sum_{k=0}^{25} a_k \sin\left(\frac{2\pi k}{1000} i + \varphi_k\right), \tag{D.1}$$

where $i = 1, \dots, 1024$ is an index into the spatial grid and $a_k \sim \mathrm{Unif}(0, 1)$, $\varphi_k \sim \mathrm{Unif}(0, 2\pi)$. The initial state is normalized as described by Sakov and Oke [33]. To generate a ground-truth trajectory from this initial state, the linear-advection dynamics $\frac{\partial \psi}{\partial t} = -\alpha \frac{\partial \psi}{\partial x}$ are simulated on the finite spatial grid. We assume constant unit velocity $\alpha = 1$ and periodic boundary conditions in space. As data, 10 equidistant state components are observed every 5 time steps. Each observation is corrupted by additive Gaussian noise.

An initial ensemble of size $r = n$ is built by successive sampling according to Eq. (D.1). The exact sampling process follows the more detailed description by Evensen [14, Section 3.1]. From this $\mathbb{R}^{n \times n}$ ensemble matrix, the sample covariance matrix is computed and used as the initial covariance for the Kalman filter. The initial factorization of the RRKF is obtained by a spectral decomposition of the sample covariance matrix truncated at the r-th largest eigenvalue. Selecting the r first ensembles serves as an initial ensemble for the EnKF and the ETKF.

Figure 2a shows the deviations of the individual low-rank approximations to the optimal KF estimate on the described data assimilation problem.

D.2 London air-quality regression

The experimental setup and the data used in Section 4.2 is provided by Hamelijnck et al. [20]. The model is selected via the log-likelihood estimation of the *exact* Kalman filter. The RMSE of the Kalman filter mean to the test data is 9.96791 (cf. Hamelijnck et al. [20]). For $r = n$ the RRKF also obtains this RMSE up to numerical error, as shown in Fig. 2b.

D.3 Spatio-temporal Matérn process with varying spatial lengthscale

Section 4.3 evaluates the low-rank approximation quality in approximate on-model spatio-temporal GP regression with varying strength of interaction between the state components. In this experiment, we consider a spatio-temporal GP with separable covariance structure, as described in Section 4.3 with hand-picked hyperparameters. The time domain is a uniform grid on the interval $[0.1, 10]$ with step size $\Delta t = 0.1$. The spatial domain is a uniform grid on $[0, 2] \times [0, 2]$ with step size $\Delta x = 0.1$.

Over the experiment, the spatial lengthscale is varied in order to evaluate low-rank approximation quality given how much correlation between the respective state components is encoded in the prior. The smaller the spatial lengthscale, the less interactions between the spatial point and the worse the low-rank approximation is expected to be for small r. We test the values $\ell_x \in \{0.01, 0.1, 0.25, 1.0\}$. For each of those values, we proceed as follows. First, a realization of the prior dynamics is drawn from the spatio-temporal Matérn process. At every temporal grid point, the entire state vector of the prior draw at that point is corrupted by additive Gaussian noise to generate data. Then, the Kalman filter estimate and the respective low-rank filtering estimates are computed for increasing values of r. Figure 3 shows for increasing spatial lengthscales and varying r (i) the resulting RMSE of the approximate filter means to the Kalman filter mean and (ii) the time-averaged Frobenius distance between the approximate filter covariance matrices to the Kalman filter covariance matrix.

D.4 Runtime

This section details the respective experimental setups used to evaluate the best-case and worst-case computational complexity of the approximate low-rank filters with respect to the state dimension n, as described in Section 4.4. We analyze the asymptotic computational complexity with respect to the state dimension n both for the linear-in-n case (Fig. 4, left) and for the quadratic-in-n case (Fig. 4, right). The elapsed wall time is always measured using the BenchmarkTools.jl software package[3] with default settings, which computes sample statistics over multiple runs automatically to prevent distortions by background processes. The setups for both analyzed cases differ only marginally from already encountered setups.

Linear-in-n case We use a setting that is similar to what is described in Section 4.1 and Appendix D.1. The state dimension is varied in order to evaluate computational complexity with respect to n. The measurement dimension is fixed at $m = 100$ and the low-rank dimension at $r = 5$. The temporal domain is a grid on the interval $[0, 100]$ with unit step size $\Delta t = 1$. Noisy observations are generated from the ground truth trajectory at every 5 time points.

The discretized linear advection dynamics with periodic boundary conditions amount to multiplication with a circulant matrix. The shift direction depends on the advection velocity, which we chose as $\alpha = 1$. Multiplication with a circulant matrix can be carried out in linear complexity via multiplication in the Fourier space. Since there is no process noise assumed, the matrix $\mathrm{B}\mathrm{B}^*$ is the zero matrix. Hence, Assumption 1 is satisfied by the dynamics model.

The measurement operator selects $m = 100$ uniformly spaced spatial points from the state vector and the measurement noise covariance matrix is a diagonal matrix. Hence, the measurement model fulfills Assumptions 2 and 3.

Quadratic-in-n case Here, we use a setting that is similar to what is described in Section 4.3 and Appendix D.3. The measurement dimension is fixed at $m = 100$ and the low-rank dimension at $r = 5$. The temporal domain is a grid on the interval $[0, 10]$ with step size $\Delta t = 0.1$. The spatial domain are n equidistant points in an interval in $\mathbb{R}$, where n varies. The generated data are noisy observations from a ground-truth realization of the prior at 20 time points, each measuring $m = 100$ state components. We evaluate the runtime of solving a spatio-temporal regression problem with this prior and the generated data.

Assumption 1 is *not* fulfilled by the dynamics model in this setting, in that evaluating the map $x \mapsto \mathrm{B}\mathrm{B}^* x$ costs $\mathcal{O}(n^2)$. By modeling spatial diffusion with a dense spatial kernel matrix, the inhomogeneity in the Lyapunov equation for the process-noise covariance matrix is also a dense matrix. Concretely, $\mathrm{B}\mathrm{B}^* = (\widetilde{\mathrm{B}}\mathcal{K}_x^{1/2})(\widetilde{\mathrm{B}}\mathcal{K}_x^{1/2})^*$, where $\widetilde{\mathrm{B}}$ is the temporal dispersion matrix and $\mathcal{K}_x$ denotes the spatial kernel matrix. Since $\mathcal{K}_x$ is dense, so is $\mathrm{B}\mathrm{B}^*$, which violates a part of Assumption 1. The cost of the low-rank prediction step thus scales quadratically in the state dimension n.

The observation model is a sparse selection operator that projects the state onto $m = 100$ points and the measurement noise covariance is a diagonal matrix. Hence, Assumptions 2 and 3 are still fulfilled by the observation model.

Figure 4 demonstrates that the theoretical complexities are empirically verified.

D.5 Large-scale spatio-temporal GP regression on rainfall data

This section provides further details on the large-scale low-rank approximation to spatio-temporal GP regression, described in Section 4.5.

We select the prior model via the log-likelihood estimate of the RRKF, as given in Eq. (23). For the model selection, the data set is distinct from the data used in the filtering/smoothing problem (Fig. 5). We use the time period from 25 January, 2010 through 5 March, 2010 and each data point is downsampled to a lower spatial resolution using cubic-spline interpolation. This reduces the state dimension to $n = 4350$ spatial points during model selection. Further, for model selection the low-rank dimension is set to $r = 200$.

[3] `https://github.com/JuliaCI/BenchmarkTools.jl`

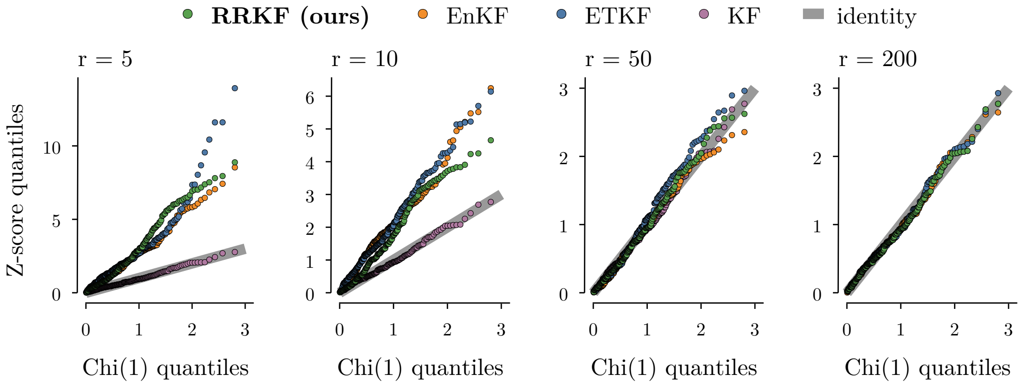


Figure E.1: *Z-score distribution for the low-rank filters with varying low-rank dimensions* r. In this on-model GP regression problem, we assume the Z-scores to be Chi(1) distributed. For the approximate low-rank filters, the Z-score distribution has too much mass in the high regimes, indicating overconfident estimates. For $r = n$, the Z-scores of the RRKF and the KF align.

E Z-scores of (approximate) Gaussian state estimates

This sections investigates a limitation of the proposed algorithm, which is addressed in Section 5. When truncating covariance information between the state components, we expect this to be reflected by a *higher* uncertainty in the resulting estimate. However, the algorithm does not account for that missing information and tends to return overconfident estimates for small values of r.

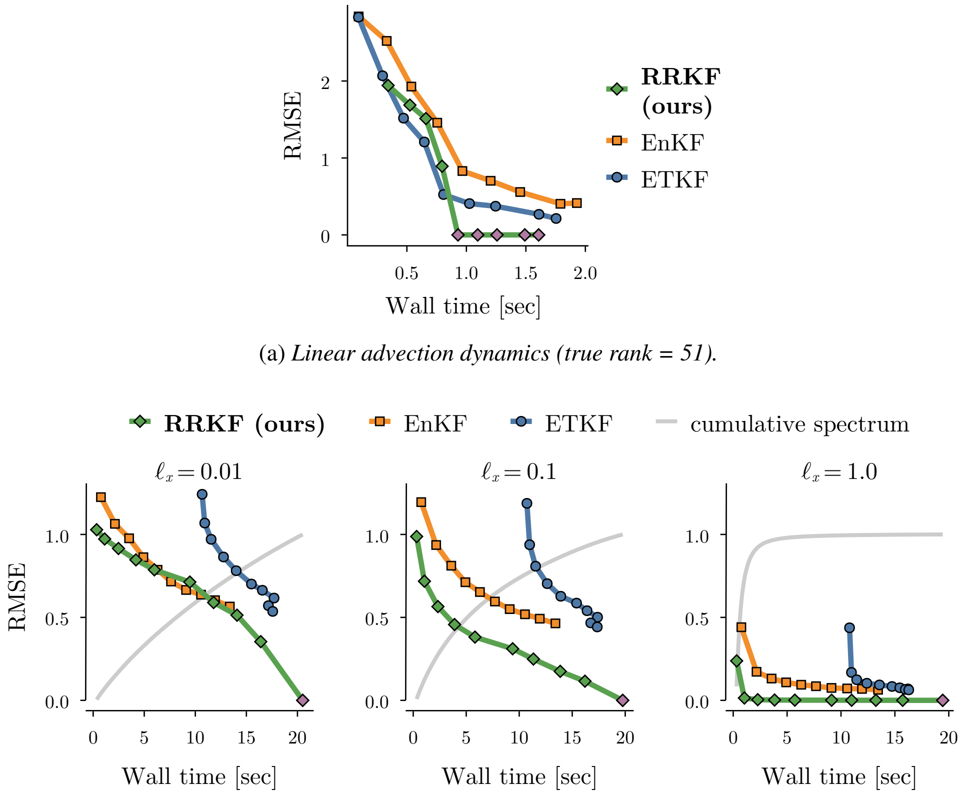
We analyze the distribution of the Z-scores of Gaussian state estimates on a Gaussian model. We expect the absolute values of the Z-scores to be distributed according to a Chi(1) distribution. A spatio-temporal GP regression problem in an on-model setting, similar to the setting described in Appendix D.3, is solved to investigate this. A ground truth realization is drawn from a spatio-temporal Matérn-1/2 process in the temporal domain $[0, 50]$ with a step size of $\Delta t = 0.1$. Data is generated by adding zero-mean Gaussian noise to $m = 150$ state components at 100 randomly sampled time points. The spatial grid is a uniformly spaced grid in the interval $[0, 20]$ with spatial step size of $\Delta x = 0.1$.

Figure E.1 visualizes the distribution of the vector of Z-scores $\frac{\mu_N - \mathrm{x}_N^\star}{\sigma_N}$, where μ_N and σ_N are the final-step filtering/smoothing mean and standard deviation, respectively. $\mathrm{x}_N^\star$ is the ground-truth state at time t_N. The Z-scores are computed for increasing values of r. The first $r = 50$ eigenvalues of the final-step Kalman filter covariance matrix account for around 97.8% of the spectrum. No covariance inflation [6, Section 4.4] is used for the ensemble methods.

It becomes apparent that—especially for very small values of r—there are significantly more high Z-scores than expected. This indicates that too many states are estimated poorly and divided by a small standard deviation. In their naïve implementation, all examined low-rank algorithms exhibit this behavior. For the RRKF, it is left for future work to find a principled solution to account for missing covariance information in a computationally efficient manner.

F Error as a Function of Raw Computation Time

In addition to the previous anaylsis of the error in the approximate filtering estimate with varying low-rank dimensions, Figures F.1a and F.1b show the error as a function of wall-clock computation times for the linear-advection model (Section 4.1) and the spatio-temporal Matérn model (Section 4.3), respectively.



(a) *Linear advection dynamics (true rank = 51).*

(b) *London air-quality regression.*

Figure F.1: *Error of the low-rank filters as a function of wall-clock computation time.* The raw computational expenses are comparable to those of the ensemble methods. The lower-left corner is the optimal setting with low error and low computation time. All methods approach this region of the plot for faster decays of the spectrum, while the RRKF performs better most of the time.