Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline 54-7 (2021) 821–826

Algorithms for Block Tridiagonal Systems: Stability Results for Generalized Kalman Smoothing

Aleksandr Y. Aravkin * **James V. Burke** ** **Bradley M. Bell** *** **Gianluigi Pillonetto** ****

* *Applied Mathematics, University of Washington, Seattle, WA 98195 USA (e-mail: saravkin@uw.edu).*
** *Mathematics, University of Washington, Seattle, WA 98195 USA (e-mail: jvburke@uw.edu)*
*** *Applied Physics Lab, University of Washington, Seattle, WA 98195 USA (e-mail: bradbell@seanet.com)*
**** *Department of Information Engineering, University of Padova, Padova, Italy (e-mail: giapi@dei.unipd.it)*

Abstract: Block tridiagonal systems appear in classic Kalman smoothing problems, as well in generalized Kalman smoothing, where problems may have nonsmooth terms, singular covariance, constraints, nonlinear models, and unknown parameters. In this paper, first we interpret all the classic smoothing algorithms as different approaches to solve positive definite block tridiagonal linear systems. Then, we obtain new results on their numerical stability. Our outcomes apply to all systems with dynamic structure, informing both classic and modern inference for generalized Kalman smoothing.

Keywords: generalized Kalman smoothing; linear algebra; numerical stability; optimization

1. INTRODUCTION

Kalman filtering (Kalman, 1960; Kalman and Bucy, 1961) and smoothing methods form a broad category of computational algorithms used for inference on noisy dynamical systems. The classic linear Gaussian model is as follows:

$$\begin{aligned} x_1 &\sim N(x_0, Q_1), \\ x_k &= G_k x_{k-1} + w_k \quad k = 2, \ldots, N, \\ z_k &= H_k x_k + v_k \quad\;\; k = 1, \ldots, N\,, \end{aligned} \tag{1}$$

where x_0 is known, $x_k, w_k \in \mathbf{R}^n$, $z_k, v_k \in \mathbf{R}^{m(k)}$, $G_k \in \mathbf{R}^{n\times n}$ and $H_k \in \mathbf{R}^{m(k)\times n}$, and w_k, v_k are mutually independent zero-mean Gaussian random variables with known positive definite covariance matrices Q_k and R_k, respectively.

Rauch-Tung-Striebel (RTS) and the Mayne-Fraser (MF) algorithm are used to obtain the minimum variance estimates of the states given $\{z_1, \ldots, z_N\}$. RTS (Rauch et al., 1965; Ansley and Kohn, 1982) computes the state estimates running forward and then back through the data, while MF (Mayne, 1966; Fraser and Potter, 1969; Wall et al., 1981) uses combination of two independent filters. A third algorithm proposed by Mayne (Mayne, 1966), hereby called the M procedure, is first run backward and then forward. We analyze the numerical stability of these smoothers from a single unified perspective.

The structure of linear systems in the classic linear case are pervasive in generalized Kalman smoothing (Aravkin et al., 2017), and apply to smoothing systems with sparse innovations, constraints, and outliers (Aravkin et al., 2014). The same structure appears in systems with singular covariance (Jonker et al., 2019), nonlinear systems (Bell et al., 2009; Aravkin et al., 2011) and systems with unknown parameters. Results derived here are applicable to all of these settings. We concentrate on the least squares case to make the ideas maximally clear.

The paper proceeds as follows. In Section 2 we formulate Kalman smoothing as a least squares problem where the system is symmetric block tridiagonal (SBT). In Section 3 we obtain bounds on the eigenvalues of SBT systems in terms of the behavior of the individual blocks, and show how these bounds are related to the stability of Kalman smoothing formulations. In Sections 4, 5, and 6, we characterize the RTS, M, and MF smoothers as solvers for SBT systems, and analyze their numerical stability. We conclude with a discussion of these results and their consequences.

2. LEAST SQUARES KS AND SBT SYSTEMS

Obtaining the maximum a posteriori (MAP) estimate for linear systems with Gaussian process and measurement noise is equivalent to solving the weighted least squares problem

$$\begin{aligned} \min_{\{x_k\}} f(\{x_k\}) := \sum_{k=1}^{N} \frac{1}{2}(z_k - H_k x_k)^\top R_k^{-1}(z_k - H_k x_k) \\ + \frac{1}{2}(x_k - G_k x_{k-1})^\top Q_k^{-1}(x_k - G_k x_{k-1}) \end{aligned} \tag{2}$$

where $G_1 = I$ captures the initial condition. Given a sequence of column vectors $\{v_k\}$ and matrices $\{T_k\}$ we use the notation

$$\mathrm{vec}(\{v_k\}) = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}, \; \mathrm{diag}(\{T_k\}) = \begin{bmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & T_N \end{bmatrix}.$$

⋆ This research was partially supported by the National Science Foundation grant no. DMS-1908890 and by the PROACTIVE project *Personalized whole brain models for neuroscience: Inference and validation* and by NVIDIA Corporation trough the GPU Grant Program.

10.1016/j.ifacol.2021.08.463

We make the following definitions:

$$\begin{aligned} R &= \mathrm{diag}(\{R_k\}) \qquad & x &= \mathrm{vec}(\{x_k\}) \\ Q &= \mathrm{diag}(\{Q_k\}) \qquad & \zeta &= \mathrm{vec}(\{x_0, 0, \ldots, 0\}) \\ H &= \mathrm{diag}(\{H_k\}) \qquad & z &= \mathrm{vec}(\{z_1, z_2, \ldots, z_N\}) \end{aligned} \tag{3}$$

$$G = \begin{bmatrix} \mathrm{I} & 0 & & \\ -G_2 & \mathrm{I} & \ddots & \\ & \ddots & \ddots & 0 \\ & & -G_N & \mathrm{I} \end{bmatrix} . \tag{4}$$

With definitions in (3) and (4), problem (2) can be written

$$\min_x f(x) = \frac{1}{2}\|Hx - z\|^2_{R^{-1}} + \frac{1}{2}\|Gx - \zeta\|^2_{Q^{-1}} , \tag{5}$$

where $\|a\|^2_M = a^\top M a$. Minimizing (5) is equivalent to solving

$$(H^\top R^{-1} H + G^\top Q^{-1} G)x = H^\top R^{-1} z + G^\top Q^{-1} \zeta . \tag{6}$$

The linear system in (6) is a symmetric positive definite SBT system. Let Φ denote a generic SBT matrix, with Φ_S denoting the Kalman smoothing case. We have

$$\Phi_S := H^\top R^{-1} H + G^\top Q^{-1} G = \begin{bmatrix} B_1 & C_2^{\mathrm{T}} & 0 & \cdots & 0 \\ C_2 & B_2 & C_3^{\mathrm{T}} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & & C_N^T \\ 0 & \cdots & 0 & C_N & B_N \end{bmatrix} , \tag{7}$$

with $C_k \in \mathbf{R}^{n\times n}$ and $B_k \in \mathbf{R}^{n\times n}$ defined as follows:

$$\begin{aligned} C_k &= -Q_k^{-1} G_k , \\ B_k &= Q_k^{-1} + G_{k+1}^\top Q_{k+1}^{-1} G_{k+1} + H_k^\top R_k^{-1} H_k \end{aligned} \tag{8}$$

where

$$G_{N+1} = 0 \quad \text{and} \quad G_{N+1}^\top Q_{N+1}^{-1} G_{N+1} = 0 . \tag{9}$$

This SBT structure was noted early on by Wright (1990); Fahrmeir and Kaufmann (1991); Wright (1993). SBT linear systems arise in all extensions to inference for dynamic systems, and are solved repeatedly by iterative algorithms in these settings. For detailed examples, please see Aravkin et al. (2017) for a survey both methods and types of smoothing problems. That survey also previews some of the results presented here without proofs or stability results, citing an older unpublished preprint (Aravkin et al., 2013).

3. CHARACTERIZING SBT SYSTEMS

Consider systems of form

$$g^{\mathrm{T}} q^{-1} g \tag{10}$$

where

$$q = \mathrm{diag}\{q_1, \ldots q_N\}, \quad g = \begin{bmatrix} \mathrm{I} & 0 & & \\ g_2 & \mathrm{I} & \ddots & \\ & \ddots & \ddots & 0 \\ & & g_N & \mathrm{I} \end{bmatrix} ,$$

q_i are positive definite, and g_i are square.

Let $\lambda_{\min}$, $\lambda_{\max}$, and $\sigma_{\min}$, $\sigma_{\max}$ denote the minimum and maximum eigenvalues and singular values, respectively. Simple upper bounds on the lower and upper eigenvalues of $g^{\mathrm{T}} q^{-1} g$ are derived in the following theorem.

Theorem 3.1. Consider matrix (10). Then, one has

$$\frac{\sigma^2_{\min}(g)}{\lambda_{\max}(q)} \le \lambda_{\min}(g^{\mathrm{T}} q^{-1} g) \le \lambda_{\max}(g^{\mathrm{T}} q^{-1} g) \le \frac{\sigma^2_{\max}(g)}{\lambda_{\min}(q)} . \tag{11}$$

This gives the following simple bound on the condition number κ of (10):

$$\kappa(g^{\mathrm{T}} q^{-1} g) = \frac{\lambda_{\max}(g^{\mathrm{T}} q^{-1} g)}{\lambda_{\min}(g^{\mathrm{T}} q^{-1} g)} \le \frac{\lambda_{\max}(q)\sigma^2_{\max}(g)}{\lambda_{\min}(q)\sigma^2_{\min}(g)} . \tag{12}$$

Since we typically have bounds on the eigenvalues of q, all that remains is to characterize the singular values of g in terms of the individual g_k. This is done in the next result which uses the relation

$$g^{\mathrm{T}} g = \begin{pmatrix} I + g_2^{\mathrm{T}} g_2 & g_2^{\mathrm{T}} & 0 & \cdots \\ g_2 & I + g_3^{\mathrm{T}} g_3 & & \vdots \\ \vdots & & \ddots & g_N^{\mathrm{T}} \\ 0 & & g_N & I + g_{N+1}^{\mathrm{T}} g_{N+1} \end{pmatrix} \tag{13}$$

where we define $g_{N+1} := 0$, so that the bottom right entry is the identity matrix.

Theorem 3.2. The following bounds hold for the singular values of g:

$$\begin{aligned} \max\big(0, \min_k \big\{1 + \sigma^2_{\min}(g_{k+1}) - \sigma_{\max}(g_k) - \sigma_{\max}(g_{k+1})\big\}\big) \\ \le \quad \sigma^2_{\min}(g) \quad \le \quad \sigma^2_{\max}(g) \quad \le \\ \max_k \big\{1 + \sigma^2_{\max}(g_{k+1}) + \sigma_{\max}(g_k) + \sigma_{\max}(g_{k+1})\big\} \end{aligned} \tag{14}$$

Corollary 3.1. Let $v^{\min}$ be the eigenvector corresponding to $\lambda_{\min}(g^{\mathrm{T}} g)$, and suppose that N is the index of the subvector of $v^{\min}$ with the largest norm. Then the lower bound is given by

$$\max\{0, 1 - \sigma_{\max}(g_N)\} \le \sigma^2_{\min}(g) . \tag{15}$$

4. FORWARD BLOCK TRIDIAGONAL (FBT) ALGORITHM AND THE RTS SMOOTHER

We now present the FBT algorithm. Suppose for $k = 1, \ldots, N$, $b_k \in \mathbf{R}^{n\times n}$, $e_k \in \mathbf{R}^{n\times \ell}$, $r_k \in \mathbf{R}^{n\times \ell}$, and for $k = 2, \ldots, N$, $c_k \in \mathbf{R}^{n\times n}$. We define the corresponding SBT system of equations

$$\begin{pmatrix} b_1 & c_2^{\mathrm{T}} & 0 & \cdots & 0 \\ c_2 & b_2 & & & \vdots \\ \vdots & & \ddots & & 0 \\ 0 & & c_{N-1} & b_{N-1} & c_N^{\mathrm{T}} \\ 0 & \cdots & 0 & c_N & b_N \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N-1} \\ e_N \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{N-1} \\ r_N \end{pmatrix} \tag{16}$$

Let Φ denotes the generic SBT matrix in this system. For positive definite systems, the FBT algorithm is defined in Algorithm 1 (Bell, 2000, algorithm 4). We now show that the RTS smoother is an implementation of FBT when Φ is set to the matrix Φ_S in (7). We use r to denote the column vector $[r_1^T \ldots r_N^T]^T$ on the rhs of (16). The proof is in (Aravkin, 2010, Chapter 1), and the result is also noted in Aravkin et al. (2017).

Theorem 4.1. When applied to Φ_S in (7) with $r = H^\top R^{-1} z + G^\top Q^{-1} \zeta$, Algorithm 1 is equivalent to the RTS Rauch et al. (1965) smoother.

We now relate stability of the forward block tridiagonal algorithm to the stability of the system (10).

Algorithm 1 Forward Block Tridiagonal (FBT)

The inputs to this algorithm are $\{c_k\}_{k=2}^N$, $\{b_k\}_{k=1}^N$, and $\{r_k\}_{k=1}^N$ where, for each k, $c_k \in \mathbf{R}^{n\times n}$, $b_k \in \mathbf{R}^{n\times n}$, and $r_k \in \mathbf{R}^{n\times \ell}$. The output is the sequence $\{e_k\}_{k=1}^N$ that solves equation (16), with each $e_k \in \mathbf{R}^{n\times\ell}$.

(1) Set $d_1^f = b_1$ and $s_1^f = r_1$.
For $k = 2$ To N :
- Set $d_k^f = b_k - c_k(d_{k-1}^f)^{-1}c_k^{\mathrm{T}}$.
- Set $s_k^f = r_k - c_k(d_{k-1}^f)^{-1}s_{k-1}$.

(2) Set $e_N = (d_N^f)^{-1}s_N$.
For $k = N-1$ To 1 :
- Set $e_k = (d_k^f)^{-1}(s_k^f - c_{k+1}^{\mathrm{T}}e_{k+1})$.

Theorem 4.2. Consider any SBT system $\Phi \in \mathbf{R}^{Nn}$ of form (16). Suppose we are given a lower bound α_L and an upper bound α_U on the eigenvalues of this system:

$$0 < \alpha_L \le \lambda_{\min}(\Phi) \le \lambda_{\max}(\Phi) \le \alpha_U \ . \tag{17}$$

If we apply the FBT iteration

$$d_k^f = b_k - c_k(d_{k-1}^f)^{-1}c_k^{\mathrm{T}},$$

then

$$0 < \alpha_L \le \lambda_{\min}(d_k^f) \le \lambda_{\max}(d_k^f) \le \alpha_U \quad \forall k \ . \tag{18}$$

In other words, the FBT iteration preserves eigenvalue bounds (and hence the condition number) for each block, and hence will be stable when the full system is well conditioned.

5. BACKWARD BLOCK TRIDIAGONAL ALGORITHM AND THE M SMOOTHER

In this section, we discuss the *backward block tridiagonal* (BBT) algorithm, and show it is equivalent to the M smoother (Mayne, 1966) when applied to the Kalman smoothing setting.

Let us again begin with a generic SBT system $\Phi \in \mathbf{R}^{Nn}$ of form (16). Now, starting at the lower right corner, we subtract $c_N^{\mathrm{T}}b_N^{-1}$ times row N from row $N-1$, and iterate this procedure up until we reach the first row of the matrix, using d_k to denote the resulting diagonal blocks, and s_k the corresponding right hand side of the equations:

$$d_N^b = b_N \ , \ d_k^b = b_k - c_{k+1}^{\mathrm{T}}(d_{k+1}^b)^{-1}c_{k+1} \quad (k = N-1, \cdots, 1)$$
$$s_N^b = e_N \ , \ s_k^b = r_k - c_{k+1}^{\mathrm{T}}(d_{k+1}^b)^{-1}s_{k+1} \quad (k = N-1, \cdots, 1) \ .$$

We now have a lower triangular system:

$$\begin{pmatrix} d_1^b & 0 & \cdots & & 0 \\ c_2 & d_2^b & 0 & \dots & 0 \\ & & \ddots & & \vdots \\ \vdots & & c_{N-1} & d_{N-1}^b & 0 \\ 0 & & & c_N & d_N^b \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N-1} \\ e_N \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{N-1} \\ r_N \end{pmatrix} \tag{19}$$

Next, we solve for the *first* block vector and then proceed back down, doing back substitution. The entire procedure is summarized in Algorithm 2. We have the following results.

Theorem 5.1. When applied to Φ_S in (7) with $r = H^{\top}R^{-1}z + G^{\top}Q^{-1}\zeta$, BBT is equivalent to the M smoother, i.e. (Mayne, 1966, Algorithm A).

Next, we show that the BBT algorithm has the same stability result as the FBT algorithm.

Theorem 5.2. Consider any SBT system $\Phi \in \mathbf{R}^{Nn}$ of form (16) and suppose we are given the bounds α_L and α_U for the lower and upper bounds of the eigenvalues of this system, so (17) is satisfied. If we apply the BBT iteration

$$d_k^b = b_k - c_{k+1}^{\mathrm{T}}(d_{k+1}^b)^{-1}c_{k+1}$$

then

$$0 < \alpha_L \le \lambda_{\min}(d_k^b) \le \lambda_{\max}(d_k^b) \le \alpha_U \quad \forall k \ .$$

Algorithm 2 Backward Block Tridiagonal (BBT)

The inputs to this algorithm are $\{c_k\}$, $\{b_k\}$, and $\{r_k\}$. The output is a sequence $\{e_k\}$ that solves equation (16).

(1) Set $d_N^b = b_N$ and $s_N^b = r_N$.
For $k = N-1, \dots, 1$,
- Set $d_k^b = b_k - c_{k+1}^{\mathrm{T}}(d_{k+1}^b)^{-1}c_{k+1}$.
- Set$s_k^b = r_k - c_{k+1}^{\mathrm{T}}(d_{k+1}^b)^{-1}s_{k+1}$.

(2) Set $e_1 = (d_1^b)^{-1}s_1^b$.
For $k = 2, \dots, N$,
- Set $e_k = (d_k^b)^{-1}(s_k^b - c_k e_{k-1})$.

Theorems 4.2 and 5.2 show that both forward and backward tridiagonal algorithms are stable when the SBT systems they are applied to are well conditioned.

6. TWO FILTER BLOCK TRIDIAGONAL ALGORITHM AND THE MF SMOOTHER

In this section, we discuss the MF (Mayne, 1966; Fraser and Potter, 1969) smoother for (16) and characterize its stability. Let d_k^f, s_k^f denote the forward matrix and vector terms obtained after step 1 of Algorithm 1, and d_k^b, s_k^b denote the terms obtained after step 1 of Algorithm 2, and let b_k, r_k refer to the diagonal terms and right hand side of system (16). Algorithm 3 uses elements of both forward and backward algorithms.

Algorithm 3 Two Filter Block Tridiagonal

The inputs to this algorithm are $\{c_k\}$, $\{b_k\}$, and $\{r_k\}$. The output is a sequence $\{e_k\}$ (that is shown to solve equation (16) in Theorem 6.1).

(1) Set $d_1^f = b_1$, $s_1^f = r_1$.
For $k = 2$ To N :
- Set $d_k^f = b_k - c_k(d_{k-1}^f)^{-1}c_k^{\mathrm{T}}$.
- Set $s_k^f = r_k - c_k(d_{k-1}^f)^{-1}s_{k-1}$.

(2) Set $d_N^b = b_N$ and $s_N^b = r_N$.
For $k = N-1, \dots, 1$,
- Set $d_k^b = b_k - c_{k+1}^{\mathrm{T}}(d_{k+1}^b)^{-1}c_{k+1}$.
- Set $s_k^b = r_k - c_{k+1}^{\mathrm{T}}(d_{k+1}^b)^{-1}s_{k+1}$.

(3) For $k = 1, \dots, N$
- Set $e_k = (d_k^f + d_k^b - b_k)^{-1}(s_k^f + s_k^b - r_k)$.

Theorem 6.1. The solution e to (16) is given by Algorithm 3. Furthermore, when applied to the Kalman smoothing system (6), i.e. when $\Phi = \Phi_S$, Algorithm 3 is equivalent to the Mayne-Fraser smoother. In particular, the MF update can be written

$$\hat{x}_k = (d_k^f + d_k^b - b_k)^{-1}(s_k^f + s_k^b - r_k) \ . \tag{20}$$

We now discuss the numerical stability of Algorithm 3, and of the MF scheme, see (20).

Theorem 6.2. Consider any SBT system $\Phi \in \mathbf{R}^{Nn}$ of form (16) and suppose we are given the bounds α_L and α_U for the lower and upper bounds of the eigenvalues of this system, so that (17) is satisfied. Then we also have

$$0 < \alpha_L \leq \lambda_{\min}(d_k^f + d_k^b - b_k) \leq \lambda_{\max}(d_k^f + d_k^b - b_k) \leq \alpha_U \quad \forall k\,. \tag{21}$$

Thus, when considered in the Kalman smoothing setting, the above result shows that the MF smoother has the same stability guarantees as the RTS smoother for well-conditioned systems.

7. CONCLUSIONS

We have characterized the numerical stability of SBT systems that arise in Kalman smoothing, see Theorems 3.1 and 3.2. We then showed that any well-conditioned symmetric SBT system can be solved in a stable manner with the FBT, which is equivalent to RTS in the Kalman smoothing context, and derived analogous results for M smoother, i.e. Algorithm A in Mayne (1966), and for the MF scheme. These results apply to both classic algorithms and newer optimization routines used in all generalized Kalman smoothing applications.

8. APPENDIX

8.1 Proof of Theorem 3.1

For the upper bound, note that for any vector v,

$$v^{\mathrm{T}} g^{\mathrm{T}} q^{-1} g v \leq \lambda_{\max}(q^{-1})\|gv\|^2 \leq \frac{\sigma_{\max}^2(g)}{\lambda_{\min}(q)}\|v\|^2.$$

Applying this inequality to a unit eigenvector for the maximum eigenvalue of $g^{\mathrm{T}} q^{-1} g$ gives the result. The lower bound is obtained analogously:

$$v^{\mathrm{T}} g^{\mathrm{T}} q^{-1} g v \geq \lambda_{\min}(q^{-1})\|gv\|^2 \geq \frac{\sigma_{\min}^2(g)}{\lambda_{\max}(q)}\|v\|^2.$$

Applying this inequality to a unit eigenvector for the minimum eigenvalue of $g^{\mathrm{T}} q^{-1} g$ completes the proof.

8.2 Proof of Theorem 3.2

Let $v = \mathrm{vec}(\{v_1, \ldots, v_N\})$ be any eigenvector of $g^{\mathrm{T}} g$, so that

$$g^{\mathrm{T}} g v = \lambda v\,. \tag{22}$$

Without loss of generality, let k denote the index such that subvector v_k has largest norm, i.e. $\|v_k\| = \max_{i\in[1,\ldots,N]}\{\|v_i\|\}$. Then from the kth block of (22), we get

$$g_k v_{k-1} + (I + g_{k+1}^{\mathrm{T}} g_{k+1}) v_k + g_{k+1}^{\mathrm{T}} v_{k+1} = \lambda v_k\,, \tag{23}$$

where we take $v_0 = 0$, and $g_1 = 0$. Let $u_k = \frac{v_k}{\|v_k\|}$. Multiplying (23) on the left by v_k^{T}, dividing by $\|v_k\|^2$, and rearranging terms, we get

$$\begin{aligned} 1 + {u_k}^{\mathrm{T}} g_{k+1}^{\mathrm{T}} g_{k+1} u_k - \lambda &= -u_k g_k \frac{v_{k-1}}{\|v_k\|} - u_k g_{k+1}^{\mathrm{T}} \frac{v_{k+1}}{\|v_k\|} \\ &\leq \sigma_{\max}(g_k) + \sigma_{\max}(g_{k+1})\,. \end{aligned} \tag{24}$$

This relationships in (24) yield the upper bound

$$\lambda \leq 1 + \sigma_{\max}^2(g_{k+1}) + \sigma_{\max}(g_k) + \sigma_{\max}(g_{k+1}) \tag{25}$$

and the lower bound

$$\begin{aligned} \lambda &\geq 1 + u_k g_{k+1}^{\mathrm{T}} g_{k+1} u_k - \sigma_{\max}(g_k) - \sigma_{\max}(g_{k+1}) \\ &\geq 1 + \sigma_{\min}^2(g_{k+1}) - \sigma_{\max}(g_k) - \sigma_{\max}(g_{k+1})\,. \end{aligned} \tag{26}$$

Taking the minimum over all indices, we obtain a lower bound that does not depend on a particular index:

$$\begin{aligned} \min_{j\in\{1,\ldots,N\}} &\left[1 + \sigma_{\min}^2(g_{j+1}) - \sigma_{\max}(g_j) - \sigma_{\max}(g_{j+1})\right] \\ &\leq 1 + \sigma_{\min}^2(g_{k+1}) - \sigma_{\max}(g_k) - \sigma_{\max}(g_{k+1}). \end{aligned}$$

Note this follows immediately because the particular index k on the right is a member of the set over which the minimum is taken. By an analogous argument, we also obtain an index-independent upper bound:

$$\begin{aligned} 1 &+ \sigma_{\max}^2(g_{k+1}) + \sigma_{\max}(g_k) + \sigma_{\max}(g_{k+1}) \\ &\leq \max_{j\in\{1,\ldots,N\}} \left[1 + \sigma_{\max}^2(g_{j+1}) + \sigma_{\max}(g_j) + \sigma_{\max}(g_{j+1})\right]. \end{aligned}$$

The expression $\max(0,\cdots)$ in (14) arises since the singular values are nonnegative.

Applying the computation to a generic index k, we have $s_k^f = P_{k|k}^{-1} x_{k|k}$. From these results, it immediately follows that e_N computed in step 2 of Algorithm 1 is the Kalman filter estimate (and the RTS smoother estimate) for time point N:

$$e_N = (d_N^f)^{-1} s_N^f = \left(P_{N|N}^{-1} + 0\right)^{-1} P_{N|N}^{-1} x_{N|N} = x_{N|N}\,. \tag{27}$$

We now establish the iteration in step 2 of Algorithm 1. First, following (Rauch et al., 1965, (3.29)), we define

$$C_k = P_{k|k} G_{k+1}^{\mathrm{T}} P_{k+1|k}^{-1} \tag{28}$$

for $k = 1, \ldots, N-1$.

To save space, we also use shorthand

$$\hat{P}_k := P_{k|k}, \quad \hat{x}_k := x_{k|k}\,. \tag{29}$$

At the first step, we obtain

$$\begin{aligned} e_{N-1} &= (d_{N-1}^f)^{-1}(s_{N-1}^f - c_N^{\mathrm{T}} e_N) \\ &= (\hat{P}_{N-1}^{-1} + G_N^{\mathrm{T}} Q_N^{-1} G_N)^{-1}(\hat{P}_{N-1}^{-1}\hat{x}_{N-1} - G_N^{\mathrm{T}} Q_N^{-1}\hat{x}_N) \\ &= (\hat{P}_{N-1}^{-1} + G_N^{\mathrm{T}} Q_N^{-1} G_N)^{-1}\hat{P}_{N-1}^{-1}\hat{x}_{N-1} - C_{N-1}\hat{x}_N \\ &= \hat{x}_{N-1} - C_{N-1}(G_n x_{N-1} - \hat{x}_N) \\ &= x_{N-1|N-1} + C_{N-1}(x_{N|N} - G_N x_{N-1|N-1})\,, \end{aligned} \tag{30}$$

where the Sherman-Morrison-Woodbury (SMW) formula was used to get from line 3 to line 4. Comparing this to (Rauch et al., 1965, (3.28)), we find that $e_{N-1} = x_{N-1|N}$, i.e. the RTS smoothed estimate. The computations above, when applied to the general tuple $(k, k+1)$ instead of $(N-1, N)$, show that every e_k is equivalent to $x_{k|N}$, which completes the proof.

8.3 Proof of Theorem 4.1

Looking at the very first block, we plug (8) into step 1 of Algorithm 1, obtaining

$$\begin{aligned} d_2^f =& b_2 - c_2^{\mathrm{T}}(d_1^f)^{-1} c_2 \\ =& Q_2^{-1} - \left(Q_2^{-1} G_2\right)^{\top} \left(Q_1^{-1} + H_1^{\top} R_1^{-1} H_1 + G_2^{\top} Q_2^{-1} G_2\right)^{-1} \\ &\times \left(Q_2^{-1} G_2\right) + H_2^{\top} R_2^{-1} H_2 + G_3^{\top} Q_3^{-1} G_3 \\ =& Q_2^{-1} - \left(Q_2^{-1} G_2\right)^{\top} \left(P_{1|1}^{-1} + G_2^{\top} Q_2^{-1} G_2\right)^{-1} \left(Q_2^{-1} G_2\right) \\ &+ H_2^{\top} R_2^{-1} H_2 + G_3^{\top} Q_3^{-1} G_3 \\ =& P_{2|1}^{-1} + H_2^{\top} R_2^{-1} H_2 + G_3^{\top} Q_3^{-1} G_3 = P_{2|2}^{-1} + G_3^{\top} Q_3^{-1} G_3, \end{aligned} \tag{31}$$

where $P_{1|0} = Q_1$, $P_{k|k} := \left(P_{k|k-1}^{-1} + H_k^{\top} R_k^{-1} H_k\right)^{-1}$ for $k = 1, \ldots, N$, and $(P_{k+1|k})^{-1}$ is given by

$$Q_{k+1}^{-1} - \left(Q_{k+1}^{-1} G_{k+1}\right)^{\top} \left(P_{k|k}^{-1} + G_{k+1}^{\top} Q_{k+1}^{-1} G_{k+1}\right)^{-1} \left(Q_{k+1}^{-1} G_{k+1}\right)$$

for $k = 1, \ldots, N-1$. The matrices $P_{k|k}$, $P_{k|k-1}$ are represent covariances of $x_{k|k}$ (the state at time k given the the measurements $\{z_1, \ldots, z_k\}$), and $x_{k|k-1}$ (the state estimate at time k given measurements $\{z_1, \ldots, z_{k-1}\}$). Using the same computation for the generic tuple $(k, k+1)$ establishes $d_k^f = P_{k|k}^{-1} + G_{k+1}^\top Q_{k+1}^{-1} G_{k+1}$. We now perform a similar computation for the right hand side of (6), $r = H^\top R^{-1} z + G^\top Q^{-1} \zeta$. We have

$$\begin{aligned} s_2^f =& r_2 - c_2^{\mathrm{T}} (d_1^f)^{-1} r_1 \\ =& \left(Q_2^{-1} G_2\right)^\top \left(P_{1|1}^{-1} + G_2^\top Q_2^{-1} G_2\right)^{-1} \left(H_1^\top R_1^{-1} z_1 + G_1^\top P_{1|0}^{-1} x_0\right) \\ &+ H_2^\top R_2^{-1} z_2 \\ =& P_{2|1}^{-1} x_{2|1} + H_2^\top R_2^{-1} z_2 = P_{2|2}^{-1} x_{2|2}. \end{aligned} \tag{32}$$

8.4 Proof of Theorem 4.2

For simplicity, we will focus only on the lower bound, since the same arguments apply for the upper bound. Note that $b_1 = d_1^f$, and the eigenvalues of d_1^f must satisfy $\alpha_L \leq \lambda_{\min}(d_1^f)$ since otherwise we can produce a unit-norm eigenvector $v_1 \in \mathbf{R}^n$ of d_1^f with $v_1^{\mathrm{T}} d_1^f v_1 < \alpha_L$, and then form the augmented unit vector $\widetilde{v}_1 \in \mathbf{R}^N$ with v_1 in the first block, and every other entry 0. Then we have $\widetilde{v}_1^{\mathrm{T}} \Phi \widetilde{v}_1 < \alpha L$, which violates (17). Next, define the elementary block row operation matrx S_1 to satisfy

$$S_1 \Phi S_1^{\mathrm{T}} = \begin{pmatrix} b_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2^f & c_3^{\mathrm{T}} & & \vdots \\ & c_3 & b_3 & & \\ \vdots & & \ddots & \ddots & \\ 0 & \cdots & 0 & c_N & b_N \end{pmatrix} \tag{33}$$

where $d_2^f = b_2 - c_2 (d_1^f)^{-1} c_2^{\mathrm{T}}$. Suppose now that d_2^f has an eigenvalue that is less than α_L. Then we can produce a unit eigenvector v_2 of d_2^f with $v_2^{\mathrm{T}} d_2^f v_2 < \alpha_L$, and create an augmented unit vector $\widetilde{v}_2 = \begin{bmatrix} 0_{1\times n} & v_2^{\mathrm{T}} & 0_{1\times n(N-2)} \end{bmatrix}^{\mathrm{T}}$ which satisfies

$$\widetilde{v}_2^{\mathrm{T}} S_1 \Phi S_1^{\mathrm{T}} \widetilde{v}_2 < \alpha_L \,. \tag{34}$$

Next, note that $\hat{v}_2^{\mathrm{T}} := \widetilde{v}_2^{\mathrm{T}} S_1 = \begin{bmatrix} -v_2^{\mathrm{T}} c_2 (d_1^f)^{-1} & v_2^{\mathrm{T}} & 0_{1\times n(N-2)} \end{bmatrix}^{\mathrm{T}}$, so in particular $\|\hat{v}_2\| \geq 1$. From (34), we now have

$$\hat{v}_2^T A \hat{v}_2 < \alpha_L \leq \alpha_L \|v_2\|^2 \,,$$

which violates (17). To complete the proof, note that the lower $n(N-1) \times n(N-1)$ block of $S_1 A S_1^{\mathrm{T}}$ is identical to that of A, with (17) holding for this modified system. The reduction technique can now be repeatedly applied.

8.5 Proof of Theorem 5.1

First, it is useful to state the following linear algebraic result.

Lemma 8.1. Let P, Q and $Q^{-1} + P$ be invertible matrices. Then

$$P - P(Q^{-1} + P)^{-1} P = Q^{-1} - Q^{-1}(Q^{-1} + P)^{-1} Q^{-1} \,. \tag{35}$$

Proof: Starting with the left hand side, write $P = P + Q^{-1} - Q^{-1}$. Then we have

$$\begin{aligned} P - P(Q^{-1} + P)^{-1} P &= P - P(Q^{-1} + P)^{-1}(P + Q^{-1} - Q^{-1}) \\ &= P(Q^{-1} + P)^{-1} Q^{-1} \\ &= (P + Q^{-1} - Q^{-1})(Q^{-1} + P)^{-1} Q^{-1} \\ &= Q^{-1} - Q^{-1}(Q^{-1} + P)^{-1} Q^{-1} \end{aligned}$$

End of proof

The recursion in (Mayne, 1966, Algorithm A), translated to our notation, is

$$J_k = G_{k+1}^T \left[I - J_{k+1} C_{k+1} \Delta_{k+1} C_{k+1}^T\right] J_{k+1} G_{k+1} + H_k^T R_k^{-1} H_k$$

$$\Delta_k = \left[I + \Gamma_k{}^T J_{k+1} \Gamma_k\right]^{-1} \tag{36}$$

$$\phi_k = -H_k^T R_k^{-1} z_k + G_{k+1}^T \left[I - J_{k+1} \Gamma_k \Delta_k \Gamma_k{}^T\right] \phi_{k+1} \tag{37}$$

where $Q_k = \Gamma_k \Gamma_k{}^T$. Note that in Mayne (1966), the quantities Γ_k and J_k are denoted C_k and P_k, respectively. The recursion is initialized by setting

$$J_N = H_N^T R_N^{-1} H_N \quad \text{and} \quad \phi_N = -H_N^T R_N^{-1} z_n \,. \tag{38}$$

We show that d_k^b in Algorithm 2 corresponds to $J_k + Q_k^{-1}$, while s_k^b in Algorithm 2 is precisely $-\phi_k$ in recursion (36)—(37). Recall that c_k and b_k in Algorithm 2 correspond to C_k and B_k in (8). Using this relationship, the correspondence claimed in the theorem is seen immediately to hold for step N. We show the next step of the recursion. From (36), we have

$$\begin{aligned} J_{N-1} &= H_{N-1}^T R_{N_1}^{-1} H_N + G_N^T \Phi_S G_N \\ \Phi_S &= J_N - J_N (\Gamma_N \Delta_N \Gamma_N{}^T) J_N \\ &= J_N - J_N (Q_N - Q_N (J_N{}^{-1} + Q_N^{-1})^{-1} Q_N) J_N \\ &= J_N - J_N (Q_N^{-1} + J_N)^{-1} J_N \\ &= Q_N^{-1} - Q_N^{-1} (d_N^b)^{-1} Q_N^{-1} \end{aligned} \tag{39}$$

where the SMW formula was used twice to get from line 2 to line 4, and Lemma 8.1 together with the definition of d_N^b was used to get from line 4 to line 5.

Therefore, we immediately have

$$\begin{aligned} J_{N-1} &= H_{N-1}^T R_{N_1}^{-1} H_N + G_N^T (Q_N^{-1} - Q_N^{-1} d_N^{-1} Q_N^{-1}) G_N \\ &= d_{N-1}^b - Q_{N-1} \end{aligned}$$

as claimed. Next, by Lemma 8.1, we have

$$\begin{aligned} \phi_{N-1} &= -H_{N-1}^T R_{N-1}^{-1} z_N + G_N^T (I - J_N (\Gamma_N \Delta_N \Gamma_N{}^T)) q_N \\ &= -H_{N-1}^T R_{N-1}^{-1} z_N - G_N^T (I - J_N (Q_N^{-1} + J_N)^{-1}) s_N \\ &= -H_{N-1}^T R_{N-1}^{-1} z_N - G_N^T (J_N{}^{-1} + Q_N)^{-1} J_N{}^{-1}) s_N \\ &= -H_{N-1}^T R_{N-1}^{-1} z_N - G_N^T (Q_N^{-1} (J_N + Q_N^{-1})^{-1}) s_N \\ &= -s_{N-1}^b \,. \end{aligned} \tag{40}$$

Finally, note that the smoothed estimate given in (Mayne, 1966, (A.8)) (translated to our notation)

$$\hat{x}_1 = -(J_1 + Q_1^{-1})^{-1} (-s_1^b - Q_1^{-1} x_0)$$

is precisely $(d_1^b)^{-1} r_1$, which is the estimate e_1 in step 2 of Algorithm 2. The reader can check that the forward recursion in (Mayne, 1966, (A.9)) is equivalent to the recursion in step 2 of Algorithm 2.

8.6 Proof of Theorem 5.2

Note first that $d_N^b = b_N$, and satisfies (21) by the same argument as in the proof of Theorem 4.2. Define the elementary block row operation matrix S_N to satisfy

$$S_N^{\mathrm{T}} \Phi S_N = \begin{pmatrix} d_1^b & c_2^{\mathrm{T}} & \cdots & & 0 \\ c_2 & d_2^b & c_3^{\mathrm{T}} & \ldots & 0 \\ 0 & & \ddots & c_{N-1}^{\mathrm{T}} & \vdots \\ \vdots & & c_{N-1} & d_{N-1}^b & 0 \\ 0 & \cdots & & 0 & d_N^b \end{pmatrix}$$

An analogous proof to that of Theorem 4.2 shows the upper $n(N-1)\times n(N-1)$ block of $S_N^{\mathrm{T}}\Phi S_N$ satisfies (21). Applying this reduction iteratively completes the proof.

8.7 Proof of Theorem 6.1

Given the linear system Φ in (16), let F denote the matrix whose action is equivalent to step 1 of Algorithm 3, so that $F\Phi$ is upper block triangular, and Fr recovers blocks $\{s_k^f\}$. Let B denote the matrix whose action is equivalent to steps 2 of Algorithm 3, so that $B\Phi$ is lower block triangular, and Br recovers blocks $\{s_k^b\}$. The solution e returned by Algorithm 3 can be written as follows:

$$e = ((F+B-I)\Phi)^{-1}(F+B-I)r\,. \tag{41}$$

To see this, note that $F\Phi$ has the same blocks above the diagonal as Φ, and zero blocks below the diagonal. Analogously, $B\Phi$ has the same blocks below the diagonal as Φ. Then $F\Phi+B\Phi-\Phi$ is block diagonal, with diagonal blocks given by $d_k^f+d_k^b-b_k$, which are invertible by Theorem (6.2). Since Φ is invertible, and $(F+B-I)\Phi$ is invertible, we also have $F+B-I$ is invertible.

Applying the system $F+B-I$ to r yields the blocks $s_k^f+s_k^b-r_k$. The fact that e solves (16) follows from the following calculation:

$$\begin{aligned}\Phi e &= \Phi((F+B-I)\Phi)^{-1}(F+B-I)r\\ &= \Phi\Phi^{-1}(F+B-I)^{-1}(F+B-I)r = r\end{aligned}$$

The MF smoother given in (Mayne, 1966, (B.9)) is equivalent to

$$\hat{x}_k = -(P_k+\sigma_{k|k-1}^{-1})^{-1}(q_k+g_k)\,,$$

where, translating to our notation, $q_k=-s_k^b$, $P_k=d_k^b-Q_k^{-1}$, $g_k := -\sigma_{k|k-1}^{-1}x_{k|k-1}$ from (Mayne, 1966, (B.7)), and $\sigma_{k|k}^{-1}$ is given by $\sigma_{k|k}^{-1} := d_k^f - G_{k+1}^T Q_{k+1}^{-1}G_{k+1}$. We now obtain

$$\sigma_{k|k-1}^{-1} = \sigma_{k|k}^{-1} - H_k^T R_k^{-1} H_k \quad \text{(Aravkin, 2010, Chapter 2)}$$

and therefore,

$$\begin{aligned}P_k+\sigma_{k|k-1}^{-1} &= d_k^b + d_k^f - Q_k^{-1} - G_{k+1}^T Q_{k+1}^{-1}G_{k+1} - H_k^T R_k^{-1}H_k\\ &= d_k^b + d_k^f - b_k \quad \text{by (8)}\,.\end{aligned}$$

Finally, we have

$$\begin{aligned}g_k = -\sigma_{k|k-1}^{-1}x_{k|k-1} &= -(s_k^f + H_k^T R_k^{-1} z_k)\\ &= -(s_k^f - r_k) \quad \text{by (6)}\,.\end{aligned}$$

This gives $-(q_k+g_k) = s_k^f + s_k^b - r_k$, and the lemma is proved.

8.8 Proof of Theorem 6.2

At every intermediate step, it is easy to see that

$$\begin{aligned}d_k^f + d_k^b - b_k &= b_k - c_k(d_{k-1}^f)^{-1}c_k^T + b_k - c_{k+1}^T(d_{k+1}^b)^{-1}c_{k+1} - b_k\\ &= b_k - c_k(d_{k-1}^f)^{-1}c_k^T - c_{k+1}^T(d_{k+1}^b)^{-1}c_{k+1}\end{aligned}$$

This corresponds exactly to isolating the middle block of the three by three system

$$\begin{pmatrix} d_{k-1}^f & c_k^T & 0\\ c_k & b_k & c_{k+1}^T\\ 0 & c_{k+1} & d_{k+1}^b\end{pmatrix}\,.$$

By Theorems 4.2 and 5.2, the eigenvalues of this system are bounded by the eigenvalues of the full system. Applying these theorems to the middle block shows that the system in (20) also satisfies such a bound.

REFERENCES

Ansley, C.F. and Kohn, R. (1982). A geometrical derivation of the fixed interval smoothing algorithm. *Biometrika*, 69, 486–487.

Aravkin, A.Y. (2010). *Robust Methods with Applications to Kalman Smoothing and Bundle Adjustment*. Ph.D. thesis, University of Washington, Seattle, WA.

Aravkin, A.Y., Bell, B.M., Burke, J.V., and Pillonetto, G. (2011). An ℓ_1-Laplace robust Kalman smoother. *IEEE Transactions on Automatic Control*, 56(12), 2898–2911.

Aravkin, A., Burke, J.V., Ljung, L., Lozano, A., and Pillonetto, G. (2017). Generalized kalman smoothing: Modeling and algorithms. *Automatica*, 86, 63–86.

Aravkin, A.Y., Bell, B.B., Burke, J.V., and Pillonetto, G. (2013). Kalman smoothing and block tridiagonal systems: new connections and numerical stability results. *arXiv preprint arXiv:1303.5237*.

Aravkin, A.Y., Burke, J.V., and Pillonetto, G. (2014). Robust and trend-following student's t kalman smoothers. *SIAM Journal on Control and Optimization*, 52(5), 2891–2916.

Bell, B.M., Burke, J.V., and Pillonetto, G. (2009). An inequality constrained nonlinear Kalman-Bucy smoother by interior point likelihood maximization. *Automatica*, 45(1), 25–33.

Bell, B. (2000). The marginal likelihood for parameters in a discrete Gauss-Markov process. *IEEE Transactions on Signal Processing*, 48(3), 870–873.

Fahrmeir, L. and Kaufmann, H. (1991). On Kalman filtering, posterior mode estimation, and Fisher scoring in dynamic exponential family regression. *Metrika*, 37–60.

Fraser, D.C. and Potter, J.E. (1969). The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on Automatic Control*, 387–390.

Jonker, J., Aravkin, A., Burke, J.V., Pillonetto, G., and Webster, S. (2019). Fast robust methods for singular state-space models. *Automatica*, 105, 399–405.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the AMSE - Journal of Basic Engineering*, 82(D), 35–45.

Kalman, R.E. and Bucy, R.S. (1961). New results in linear filtering and prediction theory. *Trans. ASME J. Basic Eng*, 83, 95–108.

Mayne, D.Q. (1966). A solution of the smoothing problem for linear dynamic systems. *Automatica*, 4, 73–92.

Rauch, H.E., Tung, F., and Striebel, C.T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA J.*, 3(8), 1145–1150.

Wall, J.E., Willsky, A.S., and Sandell, N.R. (1981). On the fixed-interval smoothing problem. *Stochastics*, 5, 1–41.

Wright, S.J. (1990). Solution of discrete-time optimal control problems on parallel computers. *Parallel Computing*, 16, 221–238.

Wright, S.J. (1993). Interior point methods for optimal control of discrete-time systems. *Journal of Optimization Theory and Applications*, 77, 161–187.