A Factorization Approach to Smoothing of Hidden Reciprocal Models

Francesca Paola Carli *
University of Cambridge
Cambridge, UK
fpc23@cam.ac.uk

Anna Caterina Carli
Istituto Superiore delle Comunicazioni e delle Tecnologie dell'Informazione
Rome, Italy
annacaterina.carli@mise.gov.it

***Abstract*—Acausal signals are ubiquitous in science and engineering. These processes are usually indexed by space, instead of time. Similarly to Markov processes, reciprocal processes (RPs) are defined in terms of conditional independence relations, which imply a rich sparsity structure for this class of models. In particular, the smoothing problem for Gaussian RPs can be traced back to the problem of solving a linear system with a cyclic block tridiagonal matrix as coefficient matrix. In this paper we propose two factorization techniques for the solution of the smoothing problem for Gaussian hidden reciprocal models (HRMs). The first method relies on a clever split of the problem in two subsystems where the matrices to be inverted are positive definite block tridiagonal matrices. We can thus rely on the rich literature for this kind of sparse matrices to devise an iterative procedure for the solution of the problem. The second approach, applies to scalar valued stationary reciprocal processes, in which case the coefficient matrix becomes circulant tridiagonal (and symmetric), and is based on the direct factorization of the coefficient matrix into the product of a circulant lower bidiagonal matrix and a circulant upper bidiagonal matrix. The computational complexity of both algorithms scales linearly with the length of the observation interval.**

***Index Terms*—Markov processes, acausal models, reciprocal processes, hidden Markov models, inference and learning, signal processing.**

I. Introduction

Acausal signals are ubiquitous in science and engineering. These processes are usually indexed by space, instead of time. Think for example of an image, where the intensity of a pixel at a given location is likely to be related to all surrounding pixels intensities or to a tracking task, where the destination (when known) is likely to influence the future position of a tracked target in the very same way of the path travelled so far.

In this context, particularly promising is the class of *reciprocal processes* (RPs). RPs were first introduced by Bernstein in 1932 [1] and studied in statistics and probability notably by B. Jamison [2]. Similarly to Markov processes, RPs are defined in terms of conditional independence: a $\mathbb{R}^n$–valued stochastic process $\mathbf{x}_k$ defined over the interval $\mathcal{I} = [0, N]$ is said to be reciprocal if, for any subinterval $[K, L] \subset \mathcal{I}$, the process in the interior of $[K, L]$ is conditionally independent of the process in $\mathcal{I} - [K, L]$ given $\mathbf{x}_K$ and $\mathbf{x}_L$. The conditional independence relations in the definition determine a rich sparsity structure for this kind of acausal processes that makes them very attractive for modeling of acausal signals.

In [3] it has been shown that a discrete–time *Gaussian* RP admits a second–order nearest–neighbor model – the acausal analog of autoregressive models for Markov processes – where the driving noise is locally correlated, with noise correlation structure specified by the model dynamics. This model recalls state–space models for Markov processes but it is *acausal* and the driving noise is not white. State–space models for discrete–time *finite state–space* RPs have been recently derived in [4].

In recent years a lot of attention has been devoted to solving the smoothing problem for finite state–space reciprocal models (see [4] and references therein) and to the problem of learning the model parameters of a reciprocal process (see e.g. [5] and references therein). A double sweep solution of the inference problem for Gaussian RPs that requires the solution of a certain Riccati equation has been derived in [3], while a message passing algorithm has been recently proposed in [6].

In this paper, we address the smoothing problem for Gaussian hidden reciprocal models (HRMs). The problem can be reduced to the solution of a linear system whose coefficient matrix has a cyclic block tridiagonal structure. Two factorization approaches are proposed. The first approach traces back the original problem to the solution of a banded block tridiagonal system, for which efficient solutions have been devised in the literature (e.g. Thomas algorithm [7]). This approach is suitable for non-stationary as well as stationary vector-valued Gaussian reciprocal processes and has a computational complexity that scales linearly with the length of the observed interval. The second approach only applies to stationary scalar–valued processes. For stationary reciprocal processes, the coefficient matrix turns out to be circulant. In this second approach, the circulant tridiagonal coefficient matrix is directly factorized into the product of circulant bidiagonal factors. This approach relies on an algorithm first developed in [8] for the solution of certain linear systems arising in numerical methods for the solution of partial differential equations with cyclic boundary conditions. The computational complexity of this second approach also scales linearly with the length of the observed interval.

The paper is organized as follows. In Section II the smoothing problem for Gaussian reciprocal processes is introduced

*This work was done while the author was with the Department of Engineering, University of Cambridge, United Kingdom.

ISBN 978-90-827970-1-5

and its relation with the solution of a cyclic block tridiagonal linear system is highlighted. In Section III the first algorithm for the solution of the smoothing problem for Gaussian hidden reciprocal processes is introduced. The scalar-valued stationary case is analyzed in Section IV where a second procedure for the evaluation of the optimal smoother is introduced. The two algorithms are compared on a numerical example in Section V. Section VI concludes the paper.

Notation. Throughout the paper, matrices will be denoted by bold face capital letters like $\mathbf{A}$, vectors by bold face lower case letters, e.g. $\mathbf{a}$. Non-bold lower case like a will be used to denote scalars. Given a sequence of column vectors $\{\mathbf{u}_k\}$ and matrices $\{\mathbf{T}_k\}$ we will use the notation

$$\text{vec}\left(\{\mathbf{u}_k\}\right) = \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_N \end{bmatrix}, \ \text{diag}\left(\{\mathbf{T}_k\}\right) = \begin{bmatrix} \mathbf{T}_0 & 0 & \dots & 0 \\ 0 & \mathbf{T}_1 & \ddots & \vdots \\ \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{T}_N \end{bmatrix}.$$

II. Smoothing for Gaussian HRMs and Block Cyclic Tridiagonal Systems

Let $\{\mathbf{x}_k\}$ be a zero-mean Gaussian stochastic process defined over the finite interval $\mathcal{I} = [0, N]$ and taking values in $\mathbb{R}^m$. $\{\mathbf{x}_k\}$ is reciprocal if and only if it admits the following representation

$$-\mathbf{M}_k^- \mathbf{x}_{k-1} + \mathbf{M}_k^0 \mathbf{x}_k - \mathbf{M}_k^+ \mathbf{x}_{k+1} = \mathbf{e}_k , \quad 1 \le k \le N-1 \quad (1)$$

where $\mathbf{M}_k^0$, $\mathbf{M}_k^+$, $\mathbf{M}_k^-$ are such that

$$\mathbf{M}_k^0 = (\mathbf{M}_k^0)^\top, \qquad \mathbf{M}_k^+ = (\mathbf{M}_{k+1}^-)^\top \quad (2)$$

and the driving noise $\mathbf{e}_k$ satisfies

$$\mathbb{E}\left[\mathbf{e}_k \mathbf{x}_l^\top\right] = \mathbf{I}\,\delta_{kl} \quad (3)$$

and is locally correlated with covariance $\mathbf{\Sigma}_e$

$$[\mathbf{\Sigma}_e]_{k,l} = \begin{cases} \mathbf{M}_k^0, & \text{for } l = k \\ -\mathbf{M}_k^+ & \text{for } l = k+1 \\ 0 & \text{otherwise}. \end{cases} \quad (4)$$

Equations (1)–(4) specify a second–order nearest–neighbor model for the discrete–time reciprocal process $\{\mathbf{x}_k\}$. The model recalls standard first–order state–space models for Markov processes but it is *acausal* (the system does not evolve recursively in the direction of increasing or decreasing values of k). Also, the driving noise $\mathbf{e}_k$ is not white, but locally correlated. Notice that, in order to completely specify $\mathbf{x}_k$ over the interval $\mathcal{I} = [0, N]$, some boundary conditions must be provided. Following [3], in this paper we consider *cyclic boundary conditions*, namely we assume

$$\mathbf{x}_{-1} = \mathbf{x}_N, \qquad \mathbf{x}_{N+1} = \mathbf{x}_0 . \quad (5)$$

These conditions are equivalent to extending cyclically the model (1), (2) and the noise structure (3), (4) to the whole interval $\mathcal{I} = [0, N]$, provided that, in these identities, $k-1$ and $k+1$ are defined modulo $N+1$. Equation (1) with cyclic boundary conditions (5) can be written in matrix form as

$$\mathbf{M}\mathbf{x} = \mathbf{e} \quad (6)$$

where

$$\mathbf{x} = \text{vec}\left(\{\mathbf{x}_k\}\right), \qquad \mathbf{e} = \text{vec}\left(\{\mathbf{e}_k\}\right), \quad (7)$$

and $\mathbf{M}$ is cyclic block tridiagonal and symmetric (see (2))

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_0^0 & -\mathbf{M}_0^+ & 0 & \dots & 0 & -\mathbf{M}_0^- \\ -\mathbf{M}_1^- & \mathbf{M}_1^0 & -\mathbf{M}_1^+ & 0 & \dots & 0 \\ \dots & & & & & \dots \\ 0 & \dots & 0 & -\mathbf{M}_{N-1}^- & \mathbf{M}_{N-1}^0 & -\mathbf{M}_{N-1}^+ \\ -\mathbf{M}_N^+ & 0 & \dots & 0 & -\mathbf{M}_N^- & \mathbf{M}_N^0 \end{bmatrix}. \quad (8)$$

We assume that the model (6) is well–posed, i.e. that it admits a unique solution. Being $\mathbf{\Sigma}_e = \mathbf{M}$, in particular we assume that the matrix $\mathbf{M}$ is positive definite. We are given the observations:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad 0 \le k \le N \quad (9)$$

where $\mathbf{v}_k$ is a white Gaussian noise uncorrelated with $\mathbf{e}_k$ with

$$\mathbb{E}\left[\mathbf{v}_k \mathbf{v}_l^\top\right] = \mathbf{\Lambda}_k\,\delta_{kl} . \quad (10)$$

We seek to compute the smoothed estimate

$$\hat{\mathbf{x}}_k = \mathbb{E}\left[\mathbf{x}_k \mid \mathbf{y}_0, \dots, \mathbf{y}_N\right]$$

We make the following definitions:

$$\mathbf{y} = \text{vec}\left(\{\mathbf{y}_k\}\right), \quad \mathbf{v} = \text{vec}\left(\{\mathbf{v}_k\}\right), \quad (11)$$

$$\mathbf{\Lambda} = \text{diag}\left(\{\mathbf{\Lambda}_k\}\right), \quad \mathbf{H} = \text{diag}\left(\{\mathbf{H}_k\}\right), \quad (12)$$

Equation (9) can be rewritten in matrix notation

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} . \quad (13)$$

With definitions (7), (8) and (11), (12), the maximum a posteriori (MAP) estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ can be written as the solution of the following optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \ \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2_{\mathbf{\Lambda}^{-1}} + \|\mathbf{M}\mathbf{x}\|^2_{\mathbf{M}^{-1}} \quad (14)$$

where $\|\mathbf{a}\|^2_{\mathbf{M}} = \mathbf{a}^\top \mathbf{M}\,\mathbf{a}$. By setting the first derivative to zero, one finds that $\hat{\mathbf{x}}$ must satisfy

$$\left(\mathbf{H}^\top \mathbf{\Lambda}^{-1} \mathbf{H} + \mathbf{M}\right) \mathbf{x} = \mathbf{H}^\top \mathbf{\Lambda}^{-1} \mathbf{y} \quad (15)$$

The linear system in (15) has a very special structure being the matrix $\left(\mathbf{H}^\top \mathbf{\Lambda}^{-1} \mathbf{H} + \mathbf{M}\right)$ the sum of a diagonal plus a cyclic block tridiagonal matrix. To observe it is positive definite, note that $\mathbf{M}$ is positive definite and $\mathbf{\Lambda}$ is positive definite by assumption. Let

$$\mathbf{H}^\top \mathbf{\Lambda}^{-1} \mathbf{H} + \mathbf{M} := \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 & 0 & \dots & \mathbf{C}_0 \\ \mathbf{B}_0^\top & \mathbf{A}_1 & \mathbf{B}_1 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ & & \mathbf{B}_{N-2}^\top & \mathbf{A}_{N-1} & \mathbf{B}_{N-1} \\ \mathbf{C}_0^\top & \dots & 0 & \mathbf{B}_{N-1}^\top & \mathbf{A}_N \end{bmatrix} \quad (16)$$

with $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_0 \in \mathbb{R}^{m\times m}$ given by

$$\mathbf{A}_k = \mathbf{M}_k^0 + \mathbf{H}_k^\top \mathbf{\Lambda}_k^{-1}\mathbf{H}_k \quad (17)$$

$$\mathbf{B}_k = -\mathbf{M}_k^+ \quad (18)$$

$$\mathbf{C}_0 = -\mathbf{M}_0^- \quad (19)$$

and

$$\mathbf{d} = \text{vec}\left(\{\mathbf{d}_k\}\right) \quad (20)$$

with

$$\mathbf{d}_k = \mathbf{H}_k^\top \mathbf{\Lambda}_k^{-1}\mathbf{y}_k\,. \quad (21)$$

III. A Smoothing Algorithm for Vector–Valued Gaussian HRMs

We propose the following algorithm for the solution of (15). Let $\mathbf{E}$ be the $(N-1)\times(N-1)$ block tridiagonal symmetric matrix

$$\mathbf{E} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 & 0 & \dots & 0 \\ \mathbf{B}_1^\top & \mathbf{A}_2 & \mathbf{B}_2 & \dots & \dots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{B}_{N-2} \\ 0 & \dots & 0 & \mathbf{B}_{N-2}^\top & \mathbf{A}_{N-1} \end{bmatrix}, \quad (22)$$

with matrices $\mathbf{A}_k$ and $\mathbf{B}_k$, $k = 1,\dots,N-1$ defined as in (17), (18), and let $\mathbf{x}_I$ and $\mathbf{d}_I$ be the $(N-1)$-dimensional column vectors

$$\mathbf{x}_I = \text{vec}\left(\{\mathbf{x}_1,\dots,\mathbf{x}_{N-1}\}\right), \quad (23)$$

$$\mathbf{d}_I = \text{vec}\left(\{\mathbf{d}_1,\dots,\mathbf{d}_{N-1}\}\right), \quad (24)$$

Moreover let $\mathbf{F}$ be the $(N-1)\times 2$ block matrix

$$\mathbf{F} = \begin{bmatrix} \mathbf{B}_0^\top & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & \mathbf{B}_{N-1} \end{bmatrix}. \quad (25)$$

System (15) can be decomposed as

$$\mathbf{E}\mathbf{x}_I + \mathbf{F}\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_N\end{bmatrix} = \mathbf{d}_I \quad (26)$$

$$\mathbf{F}^\top\mathbf{x}_I + \begin{bmatrix}\mathbf{A}_0 & \mathbf{C}_0\\ \mathbf{C}_0^\top & \mathbf{A}_{N-1}\end{bmatrix}\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_N\end{bmatrix} = \begin{bmatrix}\mathbf{d}_0\\ \mathbf{d}_N\end{bmatrix} \quad (27)$$

$\mathbf{E}$ is invertible, being $\left(\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H} + \mathbf{M}\right)$ in (16) positive definite. Equation (26) yields

$$\mathbf{x}_I = \mathbf{E}^{-1}\mathbf{d}_I - \mathbf{E}^{-1}\mathbf{F}\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_N\end{bmatrix} \quad (28)$$

If we substitute (28) into (27) we obtain

$$\left\{\begin{bmatrix}\mathbf{A}_0 & \mathbf{C}_0\\ \mathbf{C}_0^\top & \mathbf{A}_N\end{bmatrix} - \mathbf{F}^\top\mathbf{E}^{-1}\mathbf{F}\right\}\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_N\end{bmatrix} = \begin{bmatrix}\mathbf{d}_0\\ \mathbf{d}_N\end{bmatrix} - \mathbf{F}^\top\mathbf{E}^{-1}\mathbf{d}_I \quad (29)$$

In this way the original problem has been decomposed into two steps. The first steps amounts to solve (29) for $\mathbf{x}_0$, $\mathbf{x}_N$. Then $\mathbf{x}_I$ can be computed by means of (26). Note that the system (26) has a very special structure, being its coefficient matrix $\mathbf{E}$ a banded block tridiagonal matrix. This system can be efficiently solved via the Thomas algorithm [7], that particularized to our problem reads as follows. Let $\mathbf{q} = \text{vec}\left(\{\mathbf{q}_1,\dots,\mathbf{q}_{N-1}\}\right)$ be the block-vector $\mathbf{q} = \mathbf{d}_I - \mathbf{F}\begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_N\end{bmatrix}$. Then the $\hat{\mathbf{x}}_k$'s, $k = 1,\dots,N-1$ can be computed via the following iterative procedure:

1) Set $\mathbf{G}_1 = \mathbf{A}_1$ and $\mathbf{r}_1 = \mathbf{q}_1$
   For $k = 2$ to $N-1$

$$\text{Solve } \mathbf{P}_k\mathbf{G}_{k-1} = \mathbf{B}_k^\top \text{ for } \mathbf{P}_k \quad (30)$$

$$\text{Set } \mathbf{G}_k = \mathbf{A}_k - \mathbf{P}_k\mathbf{B}_k \quad (31)$$

$$\text{Set } \mathbf{r}_k = \mathbf{q}_k - \mathbf{P}_k\mathbf{r}_{k-1} \quad (32)$$

2) Solve $\mathbf{G}_{N-1}\mathbf{x}_{N-1} = \mathbf{r}_{N-1}$ for $\mathbf{x}_{N-1}$
   For $k = N-2$ to 1

$$\text{Solve } \mathbf{G}_k\mathbf{x}_k = \mathbf{r}_k - \mathbf{B}_k\mathbf{x}_{k+1} \text{ for } \mathbf{x}_k \quad (33)$$

Thomas algorithm for solving the banded tridiagonal system (26) can be seen as the composition of two steps: a factorization step, where the LU factors of $\mathbf{E}$, say $\mathbf{L}$ and $\mathbf{U}$, are computed

$$\mathbf{L} = \begin{bmatrix}\mathbf{I} & 0 & \dots & 0\\ \mathbf{P}_2 & \mathbf{I} & & \vdots\\ & \ddots & \ddots & 0\\ 0 & \dots & \mathbf{P}_{N-1} & \mathbf{I}\end{bmatrix} \quad \mathbf{U} = \begin{bmatrix}\mathbf{G}_1 & \mathbf{B}_1 & \dots & 0\\ 0 & \mathbf{G}_2 & & \vdots\\ & \ddots & \ddots & \mathbf{B}_{N-2}\\ 0 & \dots & 0 & \mathbf{G}_{N-1}\end{bmatrix} \quad (34)$$

with the $\mathbf{P}_k$'s and $\mathbf{G}_k$'s as in (30) and (31), plus a substitution step, that involves the solution of the two resulting block bidiagonal systems $\mathbf{U}\mathbf{x}_I = \mathbf{r}$ and $\mathbf{L}\mathbf{r} = \mathbf{q}$, with $\mathbf{r} = \text{vec}\left(\{\mathbf{r}_1,\dots\mathbf{r}_{N-1}\}\right)$. For positive definite matrices that are well conditioned [9], the Thomas algorithm is a stable procedure with computational complexity $O(Nm^3)$, that, for block size $m \ll N$ reduces approximately to N flops.

For what concerns the solution of (29), the products $\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{F}$ and $\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{d}_I$ need to be computed. Once again we can take advantage of the block tridiagonal structure of the positive definite matrix $\mathbf{E}$, since closed form formulas exist for the inverse of a block tridiagonal matrix [10]. Indeed, if we let $\mathbf{\Gamma} := \mathbf{E}^{-1}$ with block structure

$$\mathbf{\Gamma} = \begin{bmatrix}\mathbf{\Gamma}_{1,1} & \dots & \mathbf{\Gamma}_{1,N-1}\\ \vdots & & \vdots\\ \mathbf{\Gamma}_{1,N-1}^\top & & \mathbf{\Gamma}_{N-1,N-1}\end{bmatrix},$$

we get

$$\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{F} = \begin{bmatrix}\mathbf{B}_0\mathbf{\Gamma}_{1,1}\mathbf{B}_0^\top & \mathbf{B}_0\mathbf{\Gamma}_{1,N-1}\mathbf{B}_{N-1}^\top\\ \mathbf{B}_{N-1}^\top\mathbf{\Gamma}_{1,N-1}^\top\mathbf{B}_0^\top & \mathbf{B}_{N-1}^\top\mathbf{\Gamma}_{N-1,N-1}\mathbf{B}_{N-1}\end{bmatrix} \quad (35)$$

$$\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{d}_I = \begin{bmatrix}\mathbf{B}_0\sum_{i=1}^{N-1}\mathbf{\Gamma}_{1,i}\mathbf{d}_i\\ \mathbf{B}_{N-1}^\top\sum_{i=1}^{N-1}\mathbf{\Gamma}_{i,N-1}^\top\mathbf{d}_i\end{bmatrix} \quad (36)$$

so that only the first and last block-rows of $\mathbf{E}^{-1}$ are needed to compute $\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{F}$ and $\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{d}_I$. The overall procedure reads as follows.

Algorithm 1:

ISBN 978-90-827970-1-5

1) Compute the first row and last row of the matrix $\mathbf{E}^{-1}$ via Theorem 3.4 in [10].
2) Compute $\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{F}$ and $\mathbf{F}^\top\mathbf{E}^{-1}\mathbf{d}_I$ via (35) and (36).
3) Compute $[\hat{\mathbf{x}}_0^\top, \hat{\mathbf{x}}_N^\top]^\top$ by solving (29).
4) Compute the remaining $\hat{\mathbf{x}}_k$'s, $k = 1, \ldots, N-1$ solving the block tridiagonal system (26) via the iterative procedure (30)–(33).

The computational complexity of the overall algorithm for small block dimensions m is $O(N)$ since also the computational complexity of the inversion of a block tridiagonal matrix is linear in N [10].

IV. Stationary Scalar-Valued Case

Let $\{\mathbf{x}_k\}$, $k \in \mathcal{I}$ be a second order stationary scalar-valued reciprocal process. In this case (1) becomes a linear time-invariant system with constant scalar parameters m^0, $m^\pm$ and the matrix $\mathbf{M}$ in (8) becomes a circulant tridiagonal (and symmetric) matrix. Assuming a time invariant measurement model (9), the matrix in (16) is also circulant,

$$\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H}+\mathbf{M}=\begin{bmatrix} a & b & 0 & \ldots & 0 & b \\ b & a & b & 0 & \ldots & 0 \\ \ldots & & & & & \ldots \\ 0 & \ldots & 0 & b & a & b \\ b & 0 & \ldots & 0 & b & a \end{bmatrix}. \tag{37}$$

In this case, an alternative factorization procedure for the computation of the optimal smoother $\hat{\mathbf{x}}$ can be devised, that factorizes the circulant matrix $\left(\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H}+\mathbf{M}\right)$ into the product of a circulant lower bidiagonal matrix and a circulant upper bidiagonal matrix. This procedure was originally introduced in [8] in the context of numerical methods for the solution of certain partial differential equations with cyclic boundary conditions. In particular it can be shown that if $\left(\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H}+\mathbf{M}\right)$ is positive definite and diagonally dominant it admits a real factorization of the type

$$\left(\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H}+\mathbf{M}\right) = \alpha\mathbf{L}_C\mathbf{L}_C^\top \tag{38}$$

where $\mathbf{L}_C$ has the form

$$\mathbf{L}_C = \begin{bmatrix} 1 & 0 & 0 & \ldots & \ell \\ \ell & 1 & 0 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & 0 \\ 0 & \ldots & \ldots & \ell & 1 \end{bmatrix} \tag{39}$$

with $|\ell| < 1$. We call a matrix like $\mathbf{L}_c$ a circulant lower bidiagonal matrix. In particular one gets

$$\alpha = \frac{1}{2}\left[a + \sqrt{a^2 - 4b^2}\right] \tag{40}$$

$$\ell = b/\alpha \tag{41}$$

With the factorization (38)–(41) in hand, system $(\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H}+\mathbf{M})\mathbf{x} = \mathbf{d}$ can be solved by considering the two systems

$$\mathbf{L}_C\mathbf{h} = \alpha^{-1}\mathbf{d} \tag{42}$$

and

$$\mathbf{L}_C^\top\mathbf{x} = \mathbf{h} \tag{43}$$

whose coefficient matrix is a circulant lower (respectively, upper) bidiagonal matrix. By letting

$$\mathbf{L}_T = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ \ell & 1 & 0 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & 0 \\ 0 & \ldots & \ldots & \ell & 1 \end{bmatrix} \tag{44}$$

$\mathbf{L}_C$ can be decomposed as

$$\mathbf{L}_C = \mathbf{L}_T + \ell\,\mathbf{e}_1\mathbf{e}_N \tag{45}$$

where $\mathbf{e}_i$ denotes a vector whose components are all zeros except a one in position i. By using the Sherman–Morrison formula, systems (42) and (43) can be transformed into the following lower bidiagonal and upper bidiagonal Toeplitz systems

$$\mathbf{L}_T\mathbf{h} = \frac{1}{\alpha}\left\{\mathbf{d} - \left[\zeta\sum_{j=1}^{N} d_j(-\ell)^{N-j}\right]\mathbf{e}_1\right\} \tag{46}$$

$$\mathbf{L}_T^\top\mathbf{x} = \mathbf{h} - \left[\zeta\sum_{j=1}^{N} h_j(-\ell)^{j-1}\right]\mathbf{e}_N \tag{47}$$

where

$$\zeta = \frac{\ell}{1-(-\ell)^N} \tag{48}$$

and h_j and d_j denote the j-th component of the vector $\mathbf{h}$ and $\mathbf{d}$, respectively. The overall procedure reads as follows.

Algorithm 2:

1) Compute α and ℓ via (40), (41)
2) Compute ζ via (48)
3) Compute $\hat{\mathbf{x}}$ by solving (46) and (47).

The procedure is stable since $|\ell| < 1$. The algorithm solves equation (15) in $O(5N)$ flops.

V. Numerical Examples

The two proposed algorithms for the computation of the optimal smoother for a Gaussian reciprocal process have been implemented in Matlab. In this Section we compare Algorithm 1 and 2 on a scalar valued instance of the problem in the stationary case and then comment on the computational complexity/execution time of the two proposed algorithms. Let $\left(\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H}+\mathbf{M}\right)$ in (16) and $\mathbf{d}$ in (20), (21) be given by

$$\left(\mathbf{H}^\top\mathbf{\Lambda}^{-1}\mathbf{H}+\mathbf{M}\right) = \begin{bmatrix} 5 & 2 & 0 & 0 & 2 \\ 2 & 5 & 2 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 2 & 5 & 2 \\ 2 & 0 & 0 & 2 & 5 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 6 \\ 1 \\ 4 \\ 9 \\ 3 \end{bmatrix} \tag{49}$$

ISBN 978-90-827970-1-5 © EURASIP 2018

For what concerns Algorithm 1, the matrices $\mathbf{E}$ and $\mathbf{F}$ in (22), (25) are given by

$$\mathbf{E} = \begin{bmatrix} 5 & 2 & 0 \\ 2 & 5 & 2 \\ 0 & 2 & 5 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 2 \end{bmatrix}.$$

The terms in (35) and (36) are

$$\mathbf{F}^\top \mathbf{E}^{-1} \mathbf{F} = \begin{bmatrix} 0.99 & 0.19 \\ 0.19 & 0.99 \end{bmatrix}, \quad -\mathbf{F}^\top \mathbf{E}^{-1} \mathbf{d}_I = \begin{bmatrix} -0.4 \\ -3.6 \end{bmatrix}$$

To compute the optimal smoother $\hat{\mathbf{x}}$ one needs to solve the two-dimensional system

$$\begin{bmatrix} 4.0118 & 1.8118 \\ 1.8118 & 4.0118 \end{bmatrix} \begin{bmatrix} x_0 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5.6 \\ -0.6 \end{bmatrix} \tag{50}$$

for $[x_0, x_4]$, yielding $[\hat{x}_0, \hat{x}_4] = [1.8384, -0.9798]$. The remaining $\hat{x}_k$'s, $k = 1, 2, 3$ are obtained by solving the symmetric tridiagonal system (for large N, via the Thomas algorithm)

$$\begin{bmatrix} 5 & 2 & 0 \\ 2 & 5 & 2 \\ 0 & 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2.6768 \\ 4 \\ 10.9596 \end{bmatrix},$$

with associated $\mathbf{L}$ and $\mathbf{U}$ factors given by

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0.4 & 1 & 0 \\ 0 & 0.4762 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 5 & 2 & 0 \\ 0 & 4.2 & 2 \\ 0 & 0 & 4.0476 \end{bmatrix}.$$

yielding $\hat{\mathbf{x}}_I^\top = [-0.6162, 0.2020, 2.1111]$.

For what concerns Algorithm 2, the matrix $\left(\mathbf{H}^\top \mathbf{\Lambda}^{-1} \mathbf{H} + \mathbf{M}\right)$ in (49) is directly factorized into the product of a circulant lower bidiagonal matrix and a circulant upper bidiagonal matrix, $\left(\mathbf{H}^\top \mathbf{\Lambda}^{-1} \mathbf{H} + \mathbf{M}\right) = \alpha \mathbf{L}_C \mathbf{L}_C^\top$ with $\alpha = 4$ and off diagonal element $\ell = 0.5$. Via the Shermann–Morrison formula the resulting circulant lower (respectively, upper) bidiagonal systems are transformed into two lower (respectively, upper) non-circulant bidiagonal systems, that can be solved via forward (respectively, backward) substitution. In the case of the example, $\zeta = 0.48$, and (46) becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 & 0 \\ 0 & 0.5 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 1 & 0 \\ 0 & 0 & 0 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} 1.5303 \\ 0.25 \\ 1112.25 \\ 0.75 \end{bmatrix}$$

yielding $\mathbf{h}^\top = [1.5303, -0.5152, 1.2576, 1.6212, -0.0606]$. The optimal smooother can then be computed by solving equation (47)

$$\begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0.5 & 0 \\ 0 & 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1.5303 \\ -0.5151 \\ 1.2576 \\ 1.6212 \\ -0.9798 \end{bmatrix}$$

yielding to the same $\hat{\mathbf{x}}$ derived via Algorithm 1.

The computational times required by the Matlab implementation of the two algorithms have also been compared. What we observe is that, besides both the algorithms having a computational complexity that scales linearly with the length of the observed interval N, Algorithm 2 is generally faster than Algorithm 1. For $N = 500$, the computed average execution time observed for Algorithm 1 was $\bar{t}_1 = 0.059121$ s while the average execution time for Algorithm 2 was $\bar{t}_2 = 0.027039$ s. This is probably due to the fact that Algorithm 2 implementation is free from for cycles, that cannot be avoided in the implementation of Algorithm 1. On the other hand, Algorithm 1 enjoys a wider range of applicability, being suitable for generic cyclic block tridiagonal systems (non stationary, vector valued reciprocal processes).

VI. Conclusions

This paper deals with the smoothing problem for Gaussian hidden reciprocal processes. The problem can be reduced to the solution of a linear system, whose coefficient matrix has a cyclic block tridiagonal structure. Two factorization approaches have been proposed. The first method applies to general vector valued non-stationary RPs and recast the computation of the optimal smoother as the problem of solving a block tridiagonal system for which efficient algorithms based on the factorization of block tridiagonal matrices have been devised in the literature. The second method applies to scalar valued stationary RPs. In this case the coefficient matrix of the system that needs to be solved for the computation of the optimal smoother becomes circulant tridiagonal, and can be directly factorized into the product of circulant bidiagonal matrices. Both algorithms have computational complexity that scales linearly with the length of the observation interval.

References

[1] S. Bernstein, "Sur les liaisons entre les grandeurs aléatoires," *Verh. Internat. Math.-Kongr., Zurich*, pp. 288–309, 1932.

[2] B. Jamison, "Reciprocal processes," *Probability Theory and Related Fields*, vol. 30, no. 1, pp. 65–86, 1974.

[3] B. Levy, R. Frezza, and A. Krener, "Modeling and estimation of discrete-time gaussian reciprocal processes," *IEEE Transactions on Automatic Control*, vol. 35, no. 9, pp. 1013–1023, 1990.

[4] F. Carravetta and L. B. White, "Modelling and estimation for finite state reciprocal processes," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2190–2202, 2012.

[5] F. P. Carli, A. Ferrante, M. Pavon, and G. Picci, "An efficient algorithm for maximum entropy extension of block-circulant covariance matrices," *Linear Algebra and its Applications*, vol. 439, no. 8, pp. 2309–2329, 2013.

[6] F. P. Carli, "On the geometry of message passing algorithms for Gaussian reciprocal processes," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 4572–4577.

[7] L. Thomas, "Elliptic problems in linear difference equations over a network," Watson Sci. Comput. Lab. Rept., Columbia University, New York, Tech. Rep., 2010.

[8] M. Chen, "On the solution of circulant linear systems," *SIAM Journal on Numerical Analysis*, vol. 24, no. 3, pp. 668–683, 1987.

[9] N. J. Higham, *Accuracy and stability of numerical algorithms*. SIAM Publications, Philadelphia, PA, 1996.

[10] G. Meurant, "A review on the inverse of symmetric tridiagonal and block tridiagonal matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 3, pp. 707–728, 1992.

ISBN 978-90-827970-1-5 © EURASIP 2018