A Fast Solver for the Circulant Rational Covariance Extension Problem

Axel Ringh and Johan Karlsson

***Abstract*— The rational covariance extension problem is to parametrize the family of rational spectra of bounded degree that matches a given set of covariances. This article treats a circulant version of this problem, where the underlying process is periodic and we seek a spectrum that also matches a set of given cepstral coefficients. The interest in the circulant problem stems partly from the fact that this problem is a natural approximation of the non-periodic problem, but is also a tool in itself for analysing periodic processes. We develop a fast Newton algorithm for computing the solution utilizing the structure of the Hessian. This is done by extending a current algorithm for Toeplitz-plus-Hankel systems to the block-Toeplitz-plus-block-Hankel case. We use this algorithm to reduce the computational complexity of the Newton search from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$, where n corresponds to the number of covariances and cepstral coefficients.**

I. Introduction

The spectrum of a stochastic process describes the energy content of the signal across frequencies. Spectral estimation is to estimate a spectrum based on samples of the underlying process, and is an important tool in many areas such as medicine, economy, astronomy and seismology (see, e.g., [30], [34]). For a zero-mean, second-order stationary process $y(t)$, $t \in \mathbb{Z}$, the spectrum is given by the discrete-time Fourier transform of the covariance lags of the process:

$$\Phi(e^{i\theta}) = \sum_{-\infty}^{\infty} c_k e^{-ik\theta}$$

where $c_k = \mathbb{E}\{y(t)\bar{y}(t+k)\}$ and $c_{-k} = \bar{c}_k$. These covariances can be estimated from samples of the process, $y(0), \dots, y(M)$, via the biased estimate (see, e.g., [34, p. 24]):

$$c_l = \frac{1}{M} \sum_{k=0}^{M-l} y(k)\bar{y}(k+l), \text{ for } l \ll M.$$

In 1981, Kalman posted the so called *Rational Covariance Extension Problem* (RCEP) [23], which is to find a parametrization of all rational spectra of degree at most n that matches the first $n+1$ covariances. That is, given a sequence of covariances $\{c_k\}_{k=0}^n$ parametrize all rational spectra $\Phi(e^{i\theta}) = P(e^{i\theta})/Q(e^{i\theta})$ of degree at most n, where P and Q are positive trigonometric polynomials of the form

$$P(e^{i\theta}) = \sum_{k=-n}^{n} p_k e^{-ik\theta}, \quad Q(e^{i\theta}) = \sum_{k=-n}^{n} q_k e^{-ik\theta} \tag{1}$$

This work was supported by the Swedish Research Council, the Swedish Foundation for Strategic Research and the The Center for Industrial and Applied Mathematics, KTH.

A. Ringh and J. Karlsson are with the Division of Optimization and Systems Theory, Department of Mathematics, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden. aringh@kth.se, johan.karlsson@math.kth.se.

where $p_{-k} = \bar{p}_k$ and $q_{-k} = \bar{q}_k$, that matches the given covariance sequence:

$$\int_{-\pi}^{\pi} e^{ik\theta}\Phi(e^{i\theta})\frac{d\theta}{2\pi} = c_k, \text{ for } k = 0, \dots, n. \tag{2}$$

This is a moment problem, and the problem without the degree constraint is a classical problem referred to as the truncated trigonometric moment problem, and which is closely related to the Carathédory extension problem [25].

In the past decades the scalar valued version of the RCEP has been solved [4]–[8], [10], [12]–[14], [27], [29], [33], as well as several versions of the multivariate counterpart, where $y(t)$ is vector-valued [1], [15], [26], [31]. It turns out that there is a unique solution to the RCEP, where Q is completely determined by the covariance sequence c and the spectral zeros P. Moreover Q can be computed as the solution to a finite-dimensional, convex optimization problem. In order to apply this to obtain a spectrum Φ, the spectral zeros need to be selected. The most common choice is to take $P \equiv 1$, which gives the Burg Maximum Entropy solution [3]. However by utilizing the choice of P one can obtain spectral estimates with larger dynamic range and which approximates the true spectrum better. Several such approaches have been proposed, for example based on an inverse problem approach (see, e.g., [11], [24]), or based on simultaneous matching of covariances and cepstral coefficients [5], [10], [28].

In this work we consider a discrete approximation of the RCEP with cepstral matching derived in [27], for real signals. We develop a fast Newton solver for the corresponding optimization problem, and by utilizing the structure of the Hessian matrix the search direction in each iteration is computed with complexity $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$. In order to do this we generalize theory regarding Toeplitz-plus-Hankel matrices [17]–[19] to matrices with the same block structure. The outline of the article is as follows: Section II provides background on the problem of covariance and cepstral matching. Here we also introduce the notation used. In Section III we derive the general Newton solver for the problem, and in Section IV we investigate the structure of the Hessian. Section V presents the theory for solving systems with block-Toeplitz-plus-block-Hankel matrices, and in Section VI we present numerical results.

II. Background

Let $\mathfrak{P}$ be the set of all trigonometric polynomials $P(e^{i\theta})$ of the form (1) with real coefficients, and define the positive convex cone

$$\mathfrak{P}_+ := \{P \in \mathfrak{P} \mid P(e^{i\theta}) > 0 \text{ for } \theta \in [-\pi, \pi]\}.$$

By $\bar{\mathfrak{P}}_+$ we will denote its closure, i.e., all polynomials P where $P(e^{i\theta}) \geq 0$. The notation P will be reserved for the polynomial, while $p \in \mathbb{R}^{2n+1}$ will be used to denote the vector of polynomial coefficients. To ease notation we will therefore sometimes write "$p \in \mathfrak{P}_+$", instead of "for p such that $P \in \mathfrak{P}_+$".

Next consider the real sequences $x := \{x_k\}_{k=-n}^{n}$, with $x_{-k} = x_k$, and $y := \{y_k\}_{k=-n}^{n}$, with $y_{-k} = y_k$, and define the inner product as

$$\langle x, y \rangle = \sum_{k=-n}^{n} x_k y_{-k}.$$

Now note that both the coefficient vector $p \in \bar{\mathfrak{P}}_+$, and a real covariance sequence c have the same symmetry property. Hence using this inner product we can define the interior of the dual cone of $\bar{\mathfrak{P}}_+$ as the convex cone

$$\mathfrak{C}_+ = \{c \mid \langle c, p \rangle > 0,\ \forall p \in \bar{\mathfrak{P}}_+ \setminus \{0\}\}.$$

The existence of a unique solution to the RCEP can then be expressed as follows.

Theorem 1 ([8]): Given $c \in \mathfrak{C}_+$, for every $P \in \mathfrak{P}_+$ there exist a unique $Q \in \mathfrak{P}_+$ such that $\Phi = P/Q$ satisfies (2).

In the recent paper [27], Lindquist and Picci studies the *Circulant RCEP* (CRCEP), that is, the case where the stochastic process $y(t)$, $t \in \mathbb{Z}$, is periodic of period $2N$. In this case the spectrum is the discrete Fourier transform (DFT) of the covariances

$$\Phi(\zeta_k) = \sum_{j=0}^{2N-1} c_j \zeta_k^{-j}, \quad c_k = \frac{1}{2N} \sum_{j=0}^{2N-1} \zeta_j^k \Phi(\zeta_j), \tag{3}$$

where $\zeta_k = e^{ik\pi/N}$. Introducing the discrete measure

$$d\nu(\theta) = \sum_{j=0}^{2N-1} \delta(e^{i\theta} - \zeta_j) \frac{d\theta}{2N}$$

we can write (3) as

$$c_k = \int_{-\pi}^{\pi} e^{ik\theta} \Phi(e^{i\theta}) d\nu. \tag{4}$$

We thus see that this is also a moment problem, but where the mass-distribution is discrete instead of continuous, and the problem formulation is similar to that of the RCEP.

We introduce the corresponding convex cone $\mathfrak{P}_+(N)$, namely

$$\mathfrak{P}_+(N) := \{P \in \mathfrak{P} \mid P(\zeta_k) > 0 \text{ for } k = 0, \dots 2N-1\} \tag{5}$$

and by $\mathfrak{C}_+(N)$ we denote the inner of the corresponding dual cone. In [27] it was shown that given a $c \in \mathfrak{C}_+(N)$, for each $P \in \mathfrak{P}_+(N)$ there exist a unique $Q \in \mathfrak{P}_+(N)$ such that $\Phi = P/Q$ satisfies (4) for $k = 0, \dots, n$. This result is analogues to that of Theorem 1, and highlights some of the similarities of the problems. The CRCEP is interesting in its own right, however it is also a natural way to approximate the solution of the RCEP.

Theorem 2 ([27]): Fix a $P \in \mathfrak{P}_+ \subset \mathfrak{P}_+(N)$. For every $c \in \mathfrak{C}_+$ there exists $\tilde{N}$ big enough so that $c \in \mathfrak{C}_+(N)$ for all $N \geq \tilde{N}$. Let $\hat{Q}_N$ and $\hat{Q}$ the optimal solutions to the corresponding CRCEP and RCEP respectively. Then

$$\lim_{N \to \infty} \hat{Q}_N = \hat{Q}.$$

Since the computations involved in the CRCEP are based on DFT, they are computable by the Fast Fourier Transform (FFT). The above result ensures that such solution is a good approximation of the original problem.

A. Covariance and cepstral matching

The (real) cepstrum is defined as the (real) logarithm of the spectrum, $\log(\Phi)$, and the cepstral coefficients, m_k, are the Fourier coefficients of the cepstrum:

$$m_k = \int_{-\pi}^{\pi} e^{ik\theta} \log\left(\Phi(e^{i\theta})\right) \frac{d\theta}{2\pi}.$$

The cepstral coefficients can be estimated from the sampled process $\{y(k)\}_{k=0}^{M}$ [21], and by combining covariance and cepstral matching we can estimate both spectral zeros P and spectral poles Q [5], [10], [28].

The cepstrum can also be considered for periodic signals [27], and the cepstral coefficients correspond to the inverse discrete Fourier transform of the cepstrum

$$m_k = \frac{1}{2N} \sum_{j=0}^{2N-1} \zeta_j^k \log\left(\Phi(\zeta_j)\right) = \int_{-\pi}^{\pi} e^{ik\theta} \log\left(\Phi(e^{i\theta})\right) d\nu.$$

It is then possible to formulate a corresponding Enqvist-regularized problem [10] for both covariance and cepstral matching, as follows.

Theorem 3 ([27]): For any $\lambda > 0$, given $c \in \mathfrak{C}_+(N)$ and sequence $\{m_k\}_{k=0}^{n}$ there exists a unique pair $P(z)$ and $Q(z)$, with $p_0 = 1$, such that $\Phi(z) = P(z)/Q(z)$ and

$$\begin{aligned} &\int_{-\pi}^{\pi} e^{ik\theta} \Phi(e^{i\theta}) d\nu = c_k, \text{ for } k = 0, \dots, n \\ &\int_{-\pi}^{\pi} e^{ik\theta} \log\left(\Phi(e^{i\theta})\right) d\nu = m_k + \lambda \int_{-\pi}^{\pi} e^{ik\theta} \frac{1}{P(e^{i\theta})} d\nu, \\ &\text{for } k = 1, \dots, n. \end{aligned}$$

Moreover, $P(z)$ and $Q(z)$ can be obtained as the unique solution to the convex optimization problem

$$\begin{aligned} \min_{p,\, q} \quad & \mathbb{J}(P, Q) \\ \text{subject to} \quad & P(e^{i\theta}), Q(e^{i\theta}) \in \mathfrak{P}_+(N) \\ & p_0 = 1, \end{aligned} \tag{6}$$

where

$$\begin{aligned} \mathbb{J}(P, Q) = \quad & \langle c, q \rangle - \langle m, p \rangle + \int_{-\pi}^{\pi} P \log\left(\frac{P}{Q}\right) d\nu \\ & - \lambda \int_{-\pi}^{\pi} \log(P) d\nu. \end{aligned} \tag{7}$$

III. Unconstrained Newton-solver

We will now consider the optimization problem (6) in detail.[1] The problem can be reformulated into an explicitly unconstrained form, by formulating the problem in the polynomial coefficients of P and Q. Now since $p_0 = 1$ this gives that $p_1, \dots, p_n, q_0, \dots, q_n$ are the variables, giving a total of $2n + 1$ variables. Moreover, the only remaining constraint that needs to be handled is the positivity constraint. This can be handled implicitly in the algorithm, as explained further below, and the problem can thus be solved numerically by a unconstrained Newton-solver (c.f., [32]).

The functional to be minimized in the Enqvist-regularized problem is $J(P, Q)$, given in (7). The unconstrained Newton-method search for a minimum in an iterative way, and in each iteration the search direction δ is computed as the solution to $\nabla^2\mathbb{J}\delta = -\nabla\mathbb{J}$, where $\nabla\mathbb{J}$ and $\nabla^2\mathbb{J}$ are the gradient and Hessian of $\mathbb{J}$. The Newton algorithm is described in Algorithm 1, and we will now derive analytical expressions for the gradient and the Hessian.

Noting that p and q are real we have that $P(e^{i\theta}) = P(e^{-i\theta})$ and $Q(e^{i\theta}) = Q(e^{-i\theta})$, and using this when taking partial derivatives of (7) we get

$$\frac{\partial \mathbb{J}}{\partial q_k} = c_k - \int_{-\pi}^{\pi} \frac{P}{Q} e^{ik\theta} d\nu, \text{ for } k = 0, \dots, n \tag{8a}$$

$$\frac{\partial \mathbb{J}}{\partial p_k} = -m_k + \int_{-\pi}^{\pi} \log\left(\frac{P}{Q}\right) e^{ik\theta} d\nu - \lambda \int_{-\pi}^{\pi} \frac{1}{P} e^{ik\theta} d\nu, \text{ for } k = 1, \dots, n. \tag{8b}$$

Note that the derivatives with respect to q are given by the mismatch between the desired covariances and the covariances of the current estimate $\Phi = P/Q$. The derivatives with respect to $\tilde{p} := \begin{bmatrix} p_1 & \dots & p_n \end{bmatrix}^T$ are given by the mismatch plus the regularization term in the cepstral coefficients (c.f., [4], [27]).

Next we consider the second partial derivatives, which gives that the elements in the Hessian, $\nabla^2\mathbb{J}$, are given by

$$\frac{\partial^2 \mathbb{J}}{\partial q_k \partial q_\ell} = \int_{-\pi}^{\pi} \frac{P}{2Q^2} (e^{i(k+\ell)\theta} + e^{i(k-\ell)\theta}) d\nu, \tag{9a}$$

for $k, \ell = 0, \dots, n$,

$$\frac{\partial^2 \mathbb{J}}{\partial p_k \partial p_\ell} = \int_{-\pi}^{\pi} \frac{1}{2P} (e^{i(k+l)\theta} + e^{i(k-\ell)\theta}) d\nu + \lambda \int_{-\pi}^{\pi} \frac{1}{2P^2} (e^{i(k+\ell)\theta} + e^{i(k-\ell)\theta}) d\nu, \tag{9b}$$

for $k, \ell = 1, \dots, n$, and

$$\frac{\partial^2 \mathbb{J}}{\partial q_k \partial p_\ell} = -\int_{-\pi}^{\pi} \frac{1}{2Q} (e^{i(k+\ell)\theta} + e^{i(k-\ell)\theta}) d\nu, \tag{9c}$$

for $k = 0, \dots, n$, and $\ell = 1, \dots, n$.

Remark 1: Note that the sums in (8) and (9) are all computable using FFT. To see this, note that the trigonometric polynomials P and Q can be evaluated in all points $\{\zeta_k\}_{k=0}^{2N-1}$ simultaneously by FFT, namely $P(\zeta_k)$ is given by the k:th coefficient in the vector

$$FFT\{\begin{bmatrix} p_0, \frac{1}{2}p_1, \dots, \frac{1}{2}p_n, 0, \dots, 0, \frac{1}{2}p_n, \dots, \frac{1}{2}p_1 \end{bmatrix}\}.$$

Next note that an integral of the form $\int_{-\pi}^{\pi} \diamondsuit e^{ik\theta} d\nu$ is the k:th coefficient of the inverse FFT of $\diamondsuit$. Hence instead of computing each sum independently, which would require $\mathcal{O}(Nn)$ operations, we can use the inverse FFT to compute the 2N sums in $\mathcal{O}(N \log(N))$.

Remark 2: A second remark is regarding the computations of the maximum step length, $\rho_{\max}$, and the line search used to compute the actual step length. Let $\delta_{\tilde{p}}$ and δ_q be the parts of the search direction δ that corresponds to $\tilde{p}$ and q respectively. Then a step of length ρ in this direction will affect the values of the polynomials P and Q as

$$\delta_P = \rho \cdot FFT\{\begin{bmatrix} 0, \frac{1}{2}\delta_{\tilde{p}_1}, \dots, \frac{1}{2}\delta_{\tilde{p}_n} 0, \dots, 0, \frac{1}{2}\delta_{\tilde{p}_n}, \dots, \frac{1}{2}\delta_{\tilde{p}_1} \end{bmatrix}\},$$
$$\delta_Q = \rho \cdot FFT\{\begin{bmatrix} \delta_{q_0}, \frac{1}{2}\delta_{q_1}, \dots, \frac{1}{2}\delta_{q_n} 0, \dots, 0, \frac{1}{2}\delta_{q_n}, \dots, \frac{1}{2}\delta_{q_1} \end{bmatrix}\}.$$

Only components k where $\delta_P(k)$ or $\delta_Q(k)$ are negative can make the respective polynomial negative. Therefore the maximum step lengths allowed in order to keep the respective polynomial positive are given by

$$_{\tilde{p}}\rho_{\max} = \min_{k \text{ s.t } \delta_P(k)<0} \left\{ \left| \frac{\delta_P(k)}{P(k)} \right| \right\},$$
$$_{q}\rho_{\max} = \min_{k \text{ s.t } \delta_Q(k)<0} \left\{ \left| \frac{\delta_Q(k)}{Q(k)} \right| \right\}.$$

From this, the maximum step length is computed as

$$\rho_{\max} = \min\{1, \gamma \cdot {}_{\tilde{p}}\rho_{\max}, \gamma \cdot {}_{q}\rho_{\max}\}$$

where $0 < \gamma < 1$ is a parameter that will ensure strict positivity. Then an ordinary back-tracking line-search, initialized at $\rho_{\max}$, is used (see, e.g., [2, p. 464]).

Algorithm 1 Newton-solver for the problem (6)

Input: $c,\ m,\ \lambda$

$j = 0$
$q^0 = [1, 0 \dots, 0],\ \tilde{p}^0 = [0, 0 \dots, 0]$
Compute P^0 and Q^0
Compute $\nabla\mathbb{J}$ and $\nabla^2\mathbb{J}$
while $||\nabla\mathbb{J}|| > \epsilon$ **do**
  Solve $\nabla^2\mathbb{J}\,\delta = \nabla\mathbb{J}$
  Find maximum step length $\rho_{\max} \leq 1$ such that $P, Q \in \mathfrak{P}_+(N)$ for all $\rho \leq \rho_{\max}$
  Perform line-search to find step length $\rho \leq \rho_{\max}$
  $\tilde{p}^{j+1} \leftarrow \tilde{p}^j + \rho\delta_{\tilde{p}}$
  $q^{j+1} \leftarrow q^j + \rho\delta_q$
  Compute P^{j+1} and Q^{j+1}
  Compute $\nabla\mathbb{J}$ and $\nabla^2\mathbb{J}$
end while

Output: $p,\ q$

[1] Since the sequences c and m are real, the optimal solution will have real coefficients.

IV. Structure of the Hessian

In this section we will consider the structure of the Hessian matrix. By ordering the variables as

$$\begin{bmatrix} q_0 & q_1 & p_1 & \dots & q_n & p_n \end{bmatrix} \tag{10}$$

the Hessian contains a sub-matrix with block-Toeplitz-plus-block-Hankel structure. This allows us to solve step 6 of Algorithm 1 in $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$.

With the ordering of the variables according to (10), the Hessian matrix, with elements given in (9), gets the structure

$$\begin{bmatrix} d_0 & \begin{bmatrix} 2d_1 & \cdots & 2d_n \end{bmatrix} \\ \begin{bmatrix} 2d_1 \\ \vdots \\ 2d_n \end{bmatrix} & \begin{bmatrix} D_0 + D_2 & \dots & D_{-n+1} + D_{n+1} \\ \vdots & \ddots & \vdots \\ D_{n-1} + D_{n+1} & \cdots & D_0 + D_{2n} \end{bmatrix} \end{bmatrix} \tag{11}$$

where

$$\begin{aligned} d_0 &= r_0, \\ d_k &= \begin{bmatrix} r_k & s_k \end{bmatrix}, \text{ for } k = 1, \dots, n, \text{ and} \\ D_k &= \begin{bmatrix} r_k & s_k \\ s_k & t_k \end{bmatrix}, \text{ for } k = 0, \dots, 2n, \end{aligned} \tag{12}$$

and where

$$\begin{aligned} r_k &= \int_{-\pi}^{\pi} \frac{P}{2Q^2} e^{ik\theta} d\nu \\ s_k &= -\int_{-\pi}^{\pi} \frac{1}{2Q} e^{ik\theta} d\nu \\ t_k &= \int_{-\pi}^{\pi} \frac{1}{2P} e^{ik\theta} d\nu + \lambda \int_{-\pi}^{\pi} \frac{1}{2P^2} e^{ik\theta} d\nu. \end{aligned}$$

We see that the lower right block of (11) has a block-Toeplitz-plus-block-Hankel structure, where the individual blocks are 2×2-matrices. Therefore we split the matrix as

$$\nabla^2 \mathbb{J} = \begin{bmatrix} h_{11} & h_{12}^T \\ h_{12} & H_{22} \end{bmatrix}, \tag{13}$$

where $h_{11} \in \mathbb{R}^{1\times 1}$, $h_{12} \in \mathbb{R}^{2n\times 1}$ and $H_{22} \in \mathbb{R}^{2n\times 2n}$. In each Newton step we thus want to solve a system of equations of the form

$$\begin{bmatrix} h_{11} & h_{12}^T \\ h_{12} & H_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \tag{14}$$

where $x_1, y_1 \in \mathbb{R}$ and $x_2, y_2 \in \mathbb{R}^{2n}$. This can be solved first for x_1 and then for x_2 by using the Schur complement, leading to the system of equations

$$x_1 = (h_{11} - h_{12}^T H_{22}^{-1} h_{12})^{-1} (y_1 - h_{12}^T H_{22}^{-1} y_2) \tag{15a}$$

$$x_2 = H_{22}^{-1} y_2 - H_{22}^{-1} h_{12} x_1. \tag{15b}$$

Now $H_{22}^{-1} y_2$ and $H_{22}^{-1} h_{12}$ can be computed with complexity $\mathcal{O}(n^2)$, which will be explained in the next section. Thus the system (14) can be solved fast using (15a) and (15b).

V. Fast algorithm for solving block-Toeplitz-plus-block-Hankel-system

The computational bottleneck in (15) is to compute $H_{22}^{-1} y_2$ and $H_{22}^{-1} h_{12}$. In this section we will compute this in $\mathcal{O}(n^2)$, using the fact that H_{22} is a block-Toeplitz-plus-block-Hankel matrix, and hence has a structure with low displacement rank. The displacement of a matrix A is defined as $\Delta(A) := AU - UA$ for a suitable matrix U, where U is selected depending on a given structure. If $\Delta(A)$ is of low rank, then the displacement rank of A^{-1} is low as well, a fact that in many cases be used for explicit formulas for A^{-1} as well as for efficiently computing the solution to the corresponding linear system [16]. For a more complete overview of displacement structure, see, e.g., [22].

For the case of strongly non-singular,[2] scalar-valued Toeplitz-plus-Hankel matrices, the displacement structure and the structure of the inverse were derived in [17]–[19]. The inverse of A may be represented as a sum of products of triangular Toeplitz matrices, using the so called fundamental solution, and allows for solving the system of equations using FFT.[3] The fundamental solution can be computed in $\mathcal{O}(n^2)$, using a recursive scheme similar to the Levinson-Durbin algorithm for Toeplitz matrices. Note that this approach also allows for solving matrix systems without explicitly forming the matrices and hence uses considerably less memory. In this section we will generalize the results from [17]–[19] to the block-case.

A. Fundamental solution

We analyse the problem of solving Toeplitz-plus-Hankel systems using the approach in [17], but for the case when the elements are matrices of size $\ell \times \ell$. For $k\ell \times m\ell$ matrices, partitioned into blocks of size $\ell \times \ell$, we also define the block-transpose $(\cdot)^{T_B}$ as

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{km} \end{bmatrix} \quad \Rightarrow \quad A^{T_B} = \begin{bmatrix} a_{11} & \cdots & a_{k1} \\ \vdots & & \vdots \\ a_{1m} & \cdots & a_{km} \end{bmatrix}$$

where the internal structures of the blocks $a_{11}, \dots, a_{km}$ are preserved. By $I_{\ell\times\ell}$ and $0_{\ell\times\ell}$ we denote the identity and the zero matrix of size $\ell \times \ell$, and by e_k^m we denote the kth block-unit vector

$$e_k^m = \begin{bmatrix} 0_{\ell\times\ell} & \dots & I_{\ell\times\ell} & \dots & 0_{\ell\times\ell} \end{bmatrix}^{T_B} \in \mathbb{R}^{m\ell\times\ell}.$$

We also define $U = S + S^{T_B}$, where S is the block-shift-down matrix, i.e., zero expect for the elements on sub-block-diagonal which are $I_{\ell\times\ell}$.

For a block-Toeplitz-plus-block-Hankel matrix A,

$$A = \begin{bmatrix} t_0 + h_0 & t_{-1} + h_1 & \cdots & t_{-n} + h_n \\ t_1 + h_1 & t_0 + h_2 & \cdots & t_{-n+1} + h_{n+1} \\ \vdots & & \ddots & \vdots \\ t_n + h_n & t_{n-1} + h_{n+1} & \cdots & t_0 + h_{2n} \end{bmatrix}, \tag{16}$$

[2] A strongly non-singular matrix is a matrix who's principle minors all are non-singular. Positive definite matrices are one example of such matrices.

[3] The number of FFTs needed can further be reduced using the results in [20]. However this implementation is considerably more complex and does not address the bottleneck.

we can due to its displacement structure write

$$AU - UA = \sum_{j=1}^{4} g_j f_j^{T_B}. \tag{17}$$

Here g_j, f_j, for $j = 1, \ldots, 4$, are the block-vectors

$$\begin{aligned} g_1 &= e_1^{n+1} \\ g_2 &= e_{n+1}^{n+1} \\ g_3 &= -\begin{bmatrix} t_1 + h_{-1} & \cdots & t_{n+1} + h_{n-1} \end{bmatrix}^{T_B} \\ g_4 &= -\begin{bmatrix} t_{-n-1} + h_{n+1} & \cdots & t_{-1} + h_{2n+1} \end{bmatrix}^{T_B} \end{aligned} \tag{18}$$

$$\begin{aligned} f_1 &= \begin{bmatrix} t_{-1} + h_{-1} & \cdots & t_{-n-1} + h_{n-1} \end{bmatrix}^{T_B} \\ f_2 &= \begin{bmatrix} t_{n+1} + h_{n+1} & \cdots & t_1 + h_{2n+1} \end{bmatrix}^{T_B} \\ f_3 &= g_1 \\ f_4 &= g_2, \end{aligned} \tag{19}$$

where $t_{n+1}, t_{-n-1}, h_{-1}$ and h_{2n+1} are arbitrary $\ell \times \ell$ matrices.[4] By multiplying (17) to the left and right with A^{-1} we obtain the representation for the inverse as

$$UA^{-1} - A^{-1}U = \sum_{j=1}^{4} x_j y_j^{T_B}$$

where the vectors $x_1, \ldots, y_4$ is the so called fundamental solution, and solves the system of equations

$$Ax_j = g_j, \quad \text{for } j = 1, \ldots, 4, \tag{20a}$$

$$y_j^{T_B} A = f_j^{T_B}, \quad \text{for } j = 1, \ldots, 4. \tag{20b}$$

The system of equations (20) can be solved recursively by considering the consecutive systems

$$\begin{aligned} A^m x_j^m &= g_j^m, \qquad \text{for } j = 1, \ldots, 4 \\ (y_j^m)^{T_B} A^m &= (f_j^m)^{T_B}, \ \text{for } j = 1, \ldots, 4 \end{aligned}$$

and studying how the solutions for m and $m+1$ are related. This allows for solving (20) in $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$. Here $A^m, g_1^m, \ldots, f_4^m$ are the matrix and vectors obtained by taking $n = m$ in (16), (18) and (19).

The update equations for $x_1, \ldots, x_4$ are similar to the ones in [17], and can be written as

$$\begin{aligned} x_2^{m+1} &= \begin{bmatrix} x_4^m \\ I_{\ell\times\ell} \end{bmatrix} \alpha_m^{-1} \\ x_j^{m+1} &= \begin{bmatrix} x_j^m \\ 0_{\ell\times\ell} \end{bmatrix} - x_2^{m+1}\gamma_{jm}, \quad \text{for } j = 1, 3 \\ x_4^{m+1} &= U_{m+1} \begin{bmatrix} x_4^m \\ I_{\ell\times\ell} \end{bmatrix} - \begin{bmatrix} x_4^{m-1} \\ I_{\ell\times\ell} \\ 0_{\ell\times\ell} \end{bmatrix} \alpha_{m-1}^{-1}\alpha_m \\ &\quad + \begin{bmatrix} x_4^m \\ I_{\ell\times\ell} \end{bmatrix} \alpha_m^{-1} \left(-\lambda_{m+1} + \lambda_m \alpha_{m-1}^{-1}\alpha_m\right) \\ &\quad - \sum_{j=1,3} x_j^{m+1}\beta_{jm}. \end{aligned}$$

[4]Note that $t_{n+1}, t_{-n-1}, h_{-1}$ and h_{2n+1} are not specified by the problem, but introduced to ease the notation above. They will cancel in the actual computations.

However one difference is that in this formulation the constants $\alpha_m, \lambda_m, \gamma_{jm}$ and β_{jm} are matrices of size $\ell \times \ell$. These are updated as[5]

$$\begin{aligned} \alpha_m &= (f_2^m)^{T_B} x_4^m + a_0 + b_{2m}, \\ \gamma_{jm} &= (f_2^m)^{T_B} x_j^m - (e_m^{m+1})^{T_B} g_j^{m+1}, \quad \text{for } j = 1, 3, \\ \lambda_m &= (f_2^m)^{T_B} \begin{bmatrix} x_4^{m-1} \\ I_{l\times l} \end{bmatrix}, \\ \beta_{jm} &= (f_i^{m+1})^{T_B} \begin{bmatrix} x_4^m \\ I_{l\times l} \end{bmatrix}, \qquad \text{for } j = 1, 3. \end{aligned}$$

Note that in these recursions x_4^{m+1} depends on x_1^{m+1} and x_3^{m+1}, which in turn depend on x_2^{m+1}.

To obtain $y_1, \ldots, y_4$, we now want to solve (20b) recursively. However transposing the equation gives the equivalent system of equations

$$A^T y_j^{T_B T} = f_j^{T_B T}, \ j = 1, \ldots, 4.$$

Note that the operation $(\cdot)^{T_B T}$ will result in only a transpose of the individual block-elements. Now, let $\hat{g}_j, \hat{f}_j$ for $j = 1, \ldots, 4$ be the corresponding vectors associated with A^T. Considering these vectors we get the relations $\hat{g}_1 = f_3^{T_B T} = f_3$, $\hat{g}_2 = f_4^{T_B T} = f_4$, $\hat{g}_3 = -f_1^{T_B T}$, $\hat{g}_4 = -f_2^{T_B T}$, and hence

$$y_1 = -\hat{x}_3^{T_B T}, y_2 = -\hat{x}_4^{T_B T}, y_3 = \hat{x}_1^{T_B T}, y_4 = \hat{x}_2^{T_B T}$$

where $\hat{x}_j$ is the solution to

$$A^T \hat{x}_j = \hat{g}_j, \ j = 1, \ldots, 4.$$

The case of interest here is H_{22} in (13), which has symmetric block structure. Moreover, the block-elements in (12) are also symmetric, making H_{22} a symmetric matrix. Therefore $A^T = A$ and $f_j^{T_B T} = f_j$ also for $j = 1$ and 2, which gives that the solutions $y_1, \ldots, y_4$ are obtained directly as $y_1 = -x_3^{T_B T}$, $y_2 = -x_4^{T_B T}$, $y_3 = x_1^{T_B T}$, and $y_4 = x_2^{T_B T}$.

Remark 3: Note that the computation time can be reduced by treating $f_1, \ldots, f_4$ as row block-vectors, since all of the computations involve their block-transpose.

B. Triangular representation

In [19] a representation is derived for the inverse of a Toeplitz-plus-Hankel matrix as sums of products of triangular Toeplitz matrices. These triangular Toeplitz matrices are completely determined by the fundamental solution, described above. It can be verified that this also holds in the block-Toeplitz-plus-block-Hankel case.

Following the notation in [19], let

$$z = \begin{bmatrix} z_0 & z_1 & \cdots & z_{n+2} \end{bmatrix}^{T_B} \in \mathbb{R}^{(n+3)\ell\times\ell}$$

[5]These equations contain a correction of a typo in [17]. The equation for β_{2m}, after equation (5.8) in [17], should be $\beta_{2m} = \frac{\lambda_{m+1}}{\alpha_m} - \frac{\lambda_m}{\alpha_{m-1}}$. However the constant β_{2m} is not used in the block-case, since some of the simplifications yielding the constant are not possible to do when α_m and λ_m are matrix-valued.

be a block-vector and let $T_+(z)$ denote the block-lower-triangular block-Toeplitz matrix of the first $n+1$ block-elements, i.e.,

$$T_+(z) = \begin{bmatrix} z_0 & 0 & \cdots & 0 \\ z_1 & z_0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ z_n & z_{n-1} & \cdots & z_0 \end{bmatrix}.$$

Further, let $T_-(z)$ be its block-transpose, $T_-(z) = (T_+(z))^{T_B}$, and let J_k denote the block-flip-matrix with identity blocks $I_{\ell\times\ell}$ on the anti-diagonal:

$$J_k = \begin{bmatrix} 0 & \cdots & 0 & I_{\ell\times\ell} \\ 0 & \cdots & I_{\ell\times\ell} & 0 \\ \vdots & \iddots & & \vdots \\ I_{\ell\times\ell} & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{k\ell\times k\ell}.$$

Moreover, by $\hat{z} \in \mathbb{R}^{(n+3)\ell\times\ell}$ we denote the vector $J_{n+3}z$, i.e., the block-vector z block-flipped upside down, and $z' \in \mathbb{R}^{(n+1)\ell\times\ell}$ denotes the block-vector $z' := [z_k]_{k=1}^{n+1}$. Using this notation we can state the result as follows.

Proposition 1: Let A be a block-Toeplitz-plus-block-Hankel matrix of size $\ell(n+1)\times\ell(n+1)$, as given in (16), and let x_j, y_j for $j = 1,\ldots,4$ be the fundamental solution to this matrix. Moreover, let

$$a_1 = \begin{bmatrix} 0_{l\times l} & x_1^{T_B} & 0_{l\times l} \end{bmatrix}^{T_B}, b_1 = \begin{bmatrix} -I_{l\times l} & y_1^{T_B} & 0_{l\times l} \end{bmatrix}^{T_B}$$
$$a_2 = \begin{bmatrix} 0_{l\times l} & x_2^{T_B} & 0_{l\times l} \end{bmatrix}^{T_B}, b_2 = \begin{bmatrix} 0_{l\times l} & y_2^{T_B} & -I_{l\times l} \end{bmatrix}^{T_B}$$
$$a_3 = \begin{bmatrix} I_{l\times l} & x_3^{T_B} & 0_{l\times l} \end{bmatrix}^{T_B}, b_3 = \begin{bmatrix} 0_{l\times l} & y_3^{T_B} & 0_{l\times l} \end{bmatrix}^{T_B}$$
$$a_4 = \begin{bmatrix} 0_{l\times l} & x_4^{T_B} & I_{l\times l} \end{bmatrix}^{T_B}, b_4 = \begin{bmatrix} 0_{l\times l} & y_4^{T_B} & 0_{l\times l} \end{bmatrix}^{T_B}$$

and index the block-elements of these block-vectors from 0 to $n+2$, in accordance with z above. Then

$$A^{-1} = \sum_{j=1}^{4} T_+(a_j)(R(J_{n+1}\,T_+(\hat{b}_j) + T_-(b_j)) + u\,b_j'^{T_B})$$

where $u = [I_{\ell\times\ell}, 0_{\ell\times\ell}, I_{\ell\times\ell}, 0_{\ell\times\ell}, \ldots]^{T_B} \in \mathbb{R}^{(n+1)\ell\times\ell}$ and $R = T_+([0_{\ell\times\ell}, I_{\ell\times\ell}, 0_{\ell\times\ell}, I_{\ell\times\ell}, \ldots]^{T_B}) \in \mathbb{R}^{(n+1)\ell\times(n+1)\ell}$.

Hence by obtaining the fundamental solution to the system matrix recursively, and then applying the above proposition, we can solve step 6 in Algorithm 1 in $\mathcal{O}(n^2)$ and without explicitly forming the Hessian or its inverse.

VI. Numerical examples

The algorithm was implemented in `Matlab`, and in this section we present numerical experiments performed. [6] In Fig. 1 we show a numerical experiment to verify the scaling of the algorithm for solving the system of equations. We generate random positive definite block-Toeplitz-plus-block-Hankel matrices of size $2n\times 2n$, with n symmetric block-elements of size 2×2, and random vectors of size $2n\times 1$, and solve the corresponding systems. The mean and standard deviation of the solution times over 100 random problems, for $n = 800, 1600, 3200$ and 6400, for both the equation solver developed in this paper and for `Matlab`'s built-in solver, are shown in the log-log plot. The lines in the plot are the linear least squares solutions fitted to the four data points $(\log(t_n), \log(n))$, for each of the two data sets, where t_n is the average solution time over the 100 test problems for fixed n. Thus the slop of each line is an estimate of the scaling of the corresponding algorithm. The numerical values for these slopes are 1.99 and 3.05 for the equation solver developed here and `Matlab`'s built-in solver respectively, which agrees with the theoretical values 2 and 3.

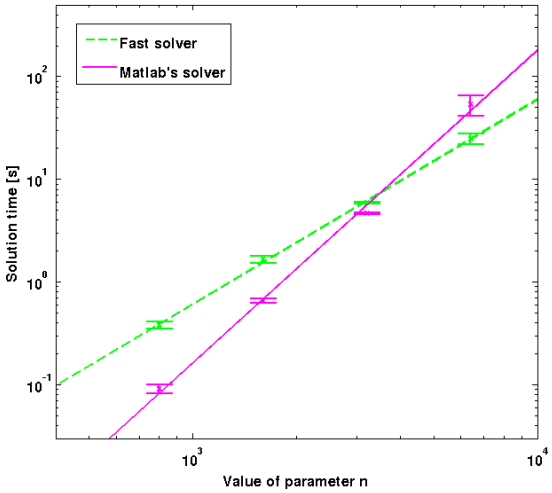

Fig. 1. Log-log plot of the average solution time for block-Toeplitz-plus-block-Hankel matrices of block-size 2, for different values of n.

In Table I solution times for the Newton-solver are presented for varying values of n. For each value of n, 100 test examples were generated and solved. In each run two positive function P and Q were generated as the sum of low degree polynomials and a function with χ^2 distributed values, from which the covariances and cepstral coefficients were computed. The corresponding solution to the optimization problem (6) were then computed. Times are reported both for when the search direction is computed with the equation solver developed above, and using `Matlab`'s solver.

The block-Toeplitz-plus-block-Hankel solver developed in this paper is faster than the `Matlab` solver when $n \gtrsim 4000$. Numerical tests indicate that when the Hessian matrix gets a condition number around 10^6, the numerical errors is of the order 10^{-3} smaller than the magnitude of the search direction. This seems to be the limit for when the Newton-solver still converges. Therefore only relatively well-conditioned problems can be solved with this approach, which exclude problem instances where the roots of Q are close to the unit circle [9].

[6] All simulations were run on an ordinary desktop computer running Ubuntu, with a 2.9GHz dual core processor and 3.8GiB RAM. However please note that the times reported here are not a completely fair comparison. The built-in functionality in `Matlab` is optimized to take advantage of multi-core processors, where as the equation solver developed here and implemented as a `Matlab` script is not always capable of using such possibilities. A comparison between optimized implementations would therefore likely give a lower threshold for where the proposed algorithm is beneficial.

TABLE I

SOLUTION TIMES FOR THE NEWTON-SOLVER, COMPARING THE FAST ALGORITHM AND MATLAB'S BUILT-IN SOLVER. AVERAGE SOLUTION TIME AND STANDARD DEVIATION OVER 100 RANDOM TEST PROBLEMS, FOR DIFFERENT VALUES OF n. $\lambda = 10^{-2}$ AND $N = 5\,000$.

	Fast solver		Matlab's \	
n	Avg. $[s]$	Std. $[s]$	Avg. $[s]$	Std. $[s]$
100	0.3727	0.008187	0.06845	0.005767
400	1.872	0.07348	0.4448	0.01679
1000	8.319	1.465	4.380	1.277
4000	124.3	24.75	153.1	28.89

VII. CONCLUSIONS

In this article we have developed a fast Newton-solver for the circulant rational covariance extension problem. This has been done by extending results for solving Toeplitz-plus-Hankel systems, to systems with the same block-structure. In this way we can compute the search direction more efficiently in each iteration, which allows the algorithm to run in $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n^3)$.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank Anders Lindquist for discussions and insights on the subject. We also want to thank the referees for valuable comments and suggestions.

REFERENCES

[1] A. Blomqvist, A. Lindquist, and R. Nagamune. Matrix-valued Nevanlinna-Pick interpolation with complexity constraint: an optimization approach. *IEEE Transactions on Automatic Control*, 48(12):2172–2190, 2003.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, 2004.

[3] J.P. Burg. *Maximum Entropy Spectral Analysis*. PhD thesis, 1975. Department of Geophysics, Stanford University.

[4] C. Byrnes, P. Enqvist, and A. Lindquist. Cepstral coefficients, covariance lags, and pole-zero models for finite data strings. *IEEE Transactions on Signal Processing*, 49(4):677–693, 2001.

[5] C. Byrnes, P. Enqvist, and A. Lindquist. Identifiability and well-posedness of shaping-filter parameterizations: A global analysis approach. *SIAM Journal of Control and Optimization*, 41(1):23–59, 2002.

[6] C. Byrnes, T.T. Georgiou, and A. Lindquist. A generalized entropy criterion for Nevanlinna-Pick interpolation with degree constraint. *IEEE Transactions on Automatic Control*, 46(6):822–839, 2001.

[7] C. Byrnes, S.V. Gusev, and A. Lindquist. A convex optimization approach to the rational covariance extension problem. *SIAM Journal on Control and Optimization*, 37(1):211–229, 1998.

[8] C. Byrnes, A. Lindquist, S.V. Gusev, and A.S. Matveev. A complete parameterization of all positive rational extensions of a covariance sequence. *IEEE Transactions on Automatic Control*, 40(11):1841–1857, 1995.

[9] P. Enqvist. A homotopy approach to rational covariance extension with degree constraint. *International journal of mathematics and computer science*, 11(5):1173–1201, 2001.

[10] P. Enqvist. A convex optimization approach to ARMA(n,m) model design from covariance and cepstral data. *SIAM Journal on Control and Optimization*, 43(3):1011–1036, 2004.

[11] G. Fanizza. *Modeling and Model Reduction by Analytic Interpolation and Optimization*. PhD thesis, 2008. Optimization and systems theory, Department of mathematics, KTH Royal Institue of Technology.

[12] T.T. Georgiou. *Partial Realization of Covariance Sequences*. PhD thesis, 1983. Center for mathematical systems theory, Univeristy of Florida.

[13] T.T. Georgiou. Realization of power spectra from partial covariance sequences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(4):438–449, 1987.

[14] T.T. Georgiou. The interpolation problem with a degree constraint. *IEEE Transactions on Automatic Control*, 44(3):631–635, 1999.

[15] T.T. Georgiou. Relative entropy and the multivariable multidimensional moment problem. *IEEE Transactions on Information Theory*, 52(3):1052–1066, 2006.

[16] G. Heinig and F. Hellinger. Displacement structure of pseudoinverses. *Linear Algebra and its Applications*, 197/198:623–649, 1994.

[17] G. Heinig, P. Jankowski, and K. Rost. Fast inversion algorithms of Toeplitz-plus-Hankel matrices. *Numerische Mathematik*, 52(6):665–682, 1988.

[18] G. Heinig and K. Rost. On the inverses of Toeplitz-plus-Hankel matrices. *Linear Algebra and its Applications*, 106(0):39 – 52, 1988.

[19] G. Heinig and K. Rost. Matrix representations of Toeplitz-plus-Hankel matrix inverses. *Linear Algebra and its Applications*, 113(0):65 – 78, 1989.

[20] G. Heinig and K. Rost. DFT representations of Toeplitz-plus-Hankel Bezoutians with application to fast matrix-vector multiplication. *Linear Algebra and its Applications*, 284(13):157 – 175, 1998.

[21] IEEE Acoustics, Speech, and Signal Processing Society. *Programs for digital signal processing*. IEEE press, New York, 1979.

[22] T. Kailath and A.H. Sayed. Displacement structure: Theory and applications. *SIAM Review*, 37(3):pp. 297–386, 1995.

[23] R. Kalman. Realization of covariance sequences. In *Toeplitz memorial conference*, 1981. Tel Aviv, Israel.

[24] J. Karlsson, T.T. Georgiou, and A. Lindquist. The inverse problem of analytic interpolation with degree constraint and weight selection for control synthesis. *IEEE Transactions on Automatic Control*, 55(2):405–418, 2010.

[25] M.G. Krein and A.A. Nudelman. *The Markov moment problem and exextremal problems*, volume Fifty of *Translations of Mathematical monographs*. American Mathematical Society, Providence, R.I., 1977.

[26] A. Lindquist, C. Masiero, and G. Picci. On the multivariate circulant rational covariance extension problem. In *IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 7155–7161, 2013.

[27] A. Lindquist and G. Picci. The circulant rational covariance extension problem: The complete solution. *IEEE Transactions on Automatic Control*, 58(11):2848–2861, 2013.

[28] B.R. Musicus and A.M. Kabel. Maximum entropy pole-zero estimation. Technical Report 510, Research Laboratory of Electronics, Massachusetts Institute of Technology, August 1985.

[29] H.I. Nurdin. New results on the rational covariance extension problem with degree constraint. *Systems & Control Letters*, 55(7):530 – 537, 2006.

[30] A.V. Oppenheim and R.W. Schafer. *Digital signal processing*. Prentice-Hall, New Jerseys, 1975.

[31] F. Ramponi, A. Ferrante, and M. Pavon. A globally convergent matricial algorithm for multivariate spectral estimation. *IEEE Transactions on Automatic Control*, 54(10):2376–2388, Oct 2009.

[32] A. Ringh. The Circulant Rational Covariance Extension Problem for a Skew Periodic Stochastic Process. Master's thesis, 2014. Optimization and systems theory, Department of mathematics, KTH Royal Institue of Technology.

[33] A. Ringh and A. Lindquist. Spectral estimation of periodic and skew periodic random signals and approximation of spectral densities. In *33rd Chinese Control Conference (CCC)*, pages 5322–5327, 2014.

[34] P. Stoica and R. Moses. *Introduction to Spectral Analysis*. Prentice-Hall, Upper Saddle River, N.J., 1997.