Numerical Experiments with Toeplitz Matrix Approximation Methods

Suliman Al-Homidan*

January 16, 2005

Abstract

Positive semidefinite Toeplitz matrix constraints arise naturally in a variety of problems in engineering. This paper deals with the numerical of this problem. Our approach is based on (i) interior point primal-dual path-following method; (ii) a projection algorithm which converges globally but slowly; (iii) the filterSQP method which is faster. Hybrid methods that attempt to combine the best features of both methods are then considered. Comparative numerical results are reported.

Key words : Alternating projections, filterSQP method, non-smooth optimization, positive semidefinite matrix, primal-dual interior-point method, Toeplitz matrix.
AMS (MOS) subject classifications; 65F99, 99C25, 65F30.

1 Introduction

The problem we are interested in is the best approximation of a given matrix by a positive semidefinite symmetric Toeplitz matrix. Toeplitz matrices appear naturally in a variety of problems in engineering. Since positive semidefinite Toeplitz matrices can be viewed as shift-invariant autocorrelation matrices, considerable attention has been paid to them, especially in

*Department of Mathematical Sciences, King Fahd University of Petroleum and Minerals, Dhahran 31261, PO Box 119, Saudi Arabia. Email: homidan@kfupm.edu.sa.

the areas of stochastic filtering and digital signal processing applications [12] and [26]. Several problems in digital signal processing and control theory require the computation of a positive definite Toeplitz matrix that closely approximates a given matrix. For example, because of rounding or truncation errors incurred while evaluating a data matrix F, F does not satisfy one or all conditions. Another example in the power spectral estimation of a wide-sense stationary process from a finite number of data, the matrix F formed from the estimated autocorrelation coefficients, is often not a positive definite Toeplitz matrix [19]. In control theory, the Gramian assignment problem for discrete-time single input system requires the computation of a positive definite Toeplitz matrix which also satisfies certain inequality constraints [17]. Here we consider two problems in which they differ by the rank requirement, and this makes the second problem unconvex and harder to solve.

First problem is; Given a data matrix $F \in \mathbb{R}^{n\times n}$, find the nearest symmetric positive semidefinite Toeplitz matrix T to F that minimizes

$$\text{minimize} \quad \phi = \|F \ - \ T\|_F \tag{1.1}$$

where $\|.\|_F$ denotes the Frobenius norm.

Second problem is; Given a data matrix $F \in \mathbb{R}^{n\times n}$, and rank m find the nearest symmetric positive semidefinite Toeplitz matrix T to F and $\text{rank}(T) = m$ that minimizes

$$\text{minimize} \quad \phi = \|F \ - \ T\|_F. \tag{1.2}$$

Problem (1.2) was studied by Suffridge et. al. [18]. They solve the problem using the self-inversive polynomial $P(x)$. The roots of the derivative of $\frac{P(z)}{z^{n-1}}$ enable them to approximate the data matrix. They also solve (1.1) using the ideas of a modified alternating projection algorithm that was successfully used in solving similar approximation problems for distance matrices [3]. In [9], alternating convex projection techniques are used to solve problem B. Oh et. al. [14] use alternating projection onto fuzzy convex sets when three or more convex sets do not intersect. Toeplitz matrix approximations are also discussed in [4, 13, 16].

In [2], a similar problems is studied. One approach followed is a projection algorithm which converges globally but the rate of convergence is very slow. Another approach is the quasi-Newton method which is faster. Then a hybrid method to combine the best features of both is used. A similar problem which

requires the knowledge of the rank was studied in [1] and formulated as a nonlinear minimization problem and then solved using techniques related to filterSQP [8].

Recently, there has been much interest in the interior point methods applied to problems with semidefinite matrix constraints (e.g. the survey papers [20, 23, 27] and the references therein). Semidefinite programming optimizes a linear function subject to positive semidefinite matrix. It is a convex programming problem since the objective and constraints are convex. In this paper, we deal with a little different problem since the objective is quadratic; also an additional rank constraint is added in (1.2) which makes the problem unconvex and harder to solve. Howevere, in Section 2, we slove (1.1) using the interior point method, where we formulate the problem as first semidefinite programming (SDP) problem then as a mixed SDP and second-order cone problem. In Section 3, (1.1) is solved using the von Neumann algorithm. Section 4, describes briefly how (1.2), which requires rank(T) to be m, formulated as a nonlinear minimization problem and then solved using the filter Sequential Quadratic Programming (SQP) [8]. This method was studied in details in [1]. In Section 5, two new hybrid methods are described to solve (1.1) firstly, there is Algorithm 5.1, which starts with the projection method to determine the rank and continues with the filterSQP method; and secondly, Algorithm 5.2 is described which solves the problem by the filterSQP method and uses the projection method to update the rank. Numerical compressions are reported in Section 6.

1.1 Notation

Define $P_{\mathcal{O}}(W)$ to be the orthogonal projection of W onto the subspace of Toeplitz matrices $\mathcal{O}$. We also need the operator $\mathcal{T} : \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n}$

$$\mathcal{T}(x) = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_1 & \dots & x_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & \dots & x_1 \end{bmatrix}. \tag{1.3}$$

Also, define the isometry operator tvec $: \mathcal{O} \longrightarrow \mathbb{R}^n$ as

$$\hat{x} = \text{tvec}\,(\mathcal{T}(x)) = [\sqrt{n}x_1\ \sqrt{2n-2}x_2 \cdots\ \sqrt{2}x_n]^T \tag{1.4}$$

for any $x \in \mathbb{R}^n$. Similarly, define $\hat{y}$ and $\hat{s}$. tvec is a linear operator satisfying the following conditions: For any $x, y \in \mathbb{R}^n$

$$\mathcal{T}(x) \bullet \mathcal{T}(y) = \hat{x}^T \hat{y}, \qquad \|\mathcal{T}(x) - \mathcal{T}(y)\|_F^2 = (\hat{x} - \hat{y})^T (\hat{x} - \hat{y}). \tag{1.5}$$

2 Mixed-Cone Formulation

A direct approach for the nearest positive semidefinite Toeplitz matrix problem is obtained by formulating it as a first SDP problem then as a mixed SDP and second-order cone problem.

To take the advantage of the isometry operator tvec , we need F to be Toeplitz. If we project F onto $\mathcal{O}$, we get $P_{\mathcal{O}}(F)$. The following lemma shows that the nearest symmetric Toeplitz positive semidefinite matrix to F is exactly equal to the nearest symmetric Toeplitz positive semidefinite matrix to $P_{\mathcal{O}}(F)$.

Lemma 2.1 *Let $\mathcal{T}(x)$ be the nearest symmetric Toeplitz positive semidefinite matrix to $P_{\mathcal{O}}(F)$, then $\mathcal{T}(x)$ is so for F.*

Proof. If $P_{\mathcal{O}}(F)$ is positive semidefinite, then we are done. If not, then for any $\mathcal{T}(x) \in \mathcal{O}$, we have

$$(\mathcal{T}(x) - P_{\mathcal{O}}(F)) \bullet (P_{\mathcal{O}}(F) - F) = 0$$

since $P_{\mathcal{O}}(F)$ is the orthogonal projection of F. Thus,

$$\|\mathcal{T}(x) - F\|_F^2 = \|\mathcal{T}(x) - P_{\mathcal{O}}(F)\|_F^2 + \|P_{\mathcal{O}}(F) - F\|_F^2.$$

This complete the proof since the second part of the above equation is constant. ■

Therefore, an equivalent problem to (1.1) is

$$\begin{aligned} \mu^* = \quad & \text{minimize} \quad \frac{1}{2}||\mathcal{T}(x) - P_{\mathcal{O}}(F)||_F^2 \\ & \text{subject to} \quad \mathcal{T}(x) \succeq 0. \end{aligned} \tag{2.1}$$

Now, we have the following equivalences (for $0 \leq \alpha \in \mathbb{R}$):

$$\begin{aligned} & \|\mathcal{T}(x) - P_{\mathcal{O}}(F)\|_F^2 \leq \alpha \\ \Leftrightarrow \quad & (\hat{x} - \hat{f})^T (\hat{x} - \hat{f}) \leq \alpha \ \text{ by (1.5)} \qquad (2.2) \\ \Leftrightarrow \quad & \alpha - (\hat{x} - \hat{f})^T I (\hat{x} - \hat{f}) \geq 0 \\ \Leftrightarrow \quad & \begin{pmatrix} I & (\hat{x} - \hat{f}) \\ (\hat{x} - \hat{f})^T & \alpha \end{pmatrix} \succeq 0 \quad \text{by Schur Complement ,} \qquad (2.3) \end{aligned}$$

where $\hat{f} = \text{tvec}\,(P_{\mathcal{O}}(F))$. Hence, we have the following SDP problem:

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & \begin{pmatrix} \mathcal{T}(x) & 0 & 0 \\ 0 & I & (\hat{x}-\hat{f}) \\ 0 & (\hat{x}-\hat{f})^T & \alpha \end{pmatrix} \succeq 0. \end{array} \tag{2.4}$$

This SDP problem has dimensions $n+1$ and $2n+1$. Also, by formulating the problem as a mixed SDP and second-order (or Lorentz) cone problem: we have the following:

$$\|\mathcal{T}(x) - P_{\mathcal{O}}(F)\|_F^2 = \|\hat{x}-\hat{f}\|_2^2.$$

So we have the following equivalent problem:

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & \mathcal{T}(x) \succeq 0 \\ & \begin{pmatrix} \alpha \\ \hat{x}-\hat{f} \end{pmatrix} \geq_Q 0. \end{array} \tag{2.5}$$

The dimension of SOC is $n+1$. The constraint $\geq_Q$ is the second-order cone constraint.

Several public domain software packages can solve (2.4) and (2.5). Many of them can be accessed via NEOS [6], (see also [11]). The main work per iteration for solving this problem is to form and solve the usually normal equations for the Newton search direction. There are many complications when forming and solving this system, since it is usually ill-conditioned at the solution.

3 The projection Method

The method of successive cyclic projections onto closed subspaces C_i's was first proposed by von Neumann [24] and independently by Wiener [25]. They showed that if, for example, C_1 and C_2 are subspaces and d is a given point, then the nearest point to d in $C_1 \cap C_2$ could be obtained by:

Algorithm 3.1 *Alternating Projection Algorithm*
Let $x_1 = d$
  For $k = 1, 2, 3, \ldots$
    $x_{k+1} = P_1(P_2(x_k))$.

Then x_k converges to the near point d in $C_1 \cap C_2$, where P_1 and P_2 are the orthogonal projections on C_1 and C_2, respectively. Dykstra [5] modified von Neumann's algorithm to handle the situation when C_1 and C_2 are replaced by convex sets. Other proofs and connections to duality along with applications were given in Han [10]. Express (1.1) as

$$\begin{array}{ll} \text{minimize} & \|\mathcal{T}(x) - F\|_F \\ \text{subject to} & \mathcal{T}(x) \in \mathcal{P} \cap \mathcal{O}. \end{array} \tag{3.1}$$

The modified Neumann's algorithm when applied to the above yields: Given a data matrix F,

Algorithm 3.2 *Modified Alternating Projection Algorithm*
Let $F_1 = F$
  For $j = 1, 2, 3, \ldots$
    $F_{j+1} = F_j + [P_{\mathcal{P}}(P_{\mathcal{O}}(F_j)) - P_{\mathcal{O}}(F_j)]$.

Then $\{P_{\mathcal{O}}(F_j)\}$ and $\{P_{\mathcal{P}}(P_{\mathcal{O}}(F_j))\}$ converge in Frobenius norm to the solution. Here, $P_{\mathcal{P}}(F)$ is the projection of F onto the convex cone $\mathcal{P}$. It is simply setting the negative eigenvalues of the spectral decomposition of F to zero.

4 The SQP Algorithms

In the previous section, the alternating projection algorithm computes a unique solution for (1.1) since the sets $\mathcal{P}$ and $\mathcal{O}$ are convex. It is the loss of convexity that increases the difficulty of (1.2). The main idea is to replace (1.2) by a smooth nonlinear programming problem in order to use a second order convergent SQP method. In this section, we use techniques related to filterSQP [8] for solving nonlinear programming problems in order to develop an algorithm to solving (1.2).

It is difficult to deal with the matrix cone constraints in (3.1) since it is not easy to specify if the elements are feasible. Using partial LDL^T factorization of $\mathcal{T}(x)$, this difficulty can be overcome. Since m, the rank of the optimal $\mathcal{T}(x)$, is known, therefore for F sufficiently close to $\mathcal{T}(x)$, the partial factors $\mathcal{T}(x) = LDL^T$ can be calculated such that

$$L = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}, \quad \mathcal{T}(x) = \begin{bmatrix} T_{11} & T_{21}^T \\ T_{21} & T_{22} \end{bmatrix}, \tag{4.1}$$

where $L_{11},\ D_1$ and T_{11} are $m \times m$ matrices; $I,\ D_2$ and T_{22} are $n-m \times n-m$ matrices; L_{21} and T_{21} are $n-m \times m$ matrices; D_1 is diagonal and $D_1 \succ 0$, and D_2 has no particular structure other than the Toeplitz form. At the solution, $D_2 = 0$ and $\mathcal{T}(x)$ are symmetric positive semidefinite Toeplitz matrix. In general,

$$D_2(\mathcal{T}(x)) \ = \ T_{22} \ - \ T_{21}T_{11}^{-1}T_{21}^T \tag{4.2}$$

which enables the constraint $\mathcal{T}(x) \in \mathcal{P}$ to be written in the form

$$D_2(\mathcal{T}(x)) = 0. \tag{4.3}$$

The condition that F is close to $\mathcal{T}(x)$ is needed to ensure that $D_1(\mathcal{T}(x)) \succ 0$. Hence, (1.2) can now be expressed as

$$\begin{aligned} &\text{minimize} \quad \phi \\ &\text{subject to} \ \ D_2(\mathcal{T}(x)) \ = \ 0 = Z^T\mathcal{T}(x)Z, \end{aligned} \tag{4.4}$$

where $Z \ = \ \begin{bmatrix} -T_{11}^{-1}T_{21}^T \\ I \end{bmatrix}$ is the basis matrix for the null space of $\mathcal{T}(x)$ when $D_2 = 0$. This approach has been studied in a similar way by [7]. The structure of the Toeplitz matrix $\mathcal{T}(x)$ as given in (1.3), is

$$\phi = \sum_{i,j=1}^{n} (f_{ij} - x_{|i-j+1|})^2, \tag{4.5}$$

For the gradient and Hessian of ϕ see [1] and since both are available, therefore the filterSQP can be used to solve (4.4). Also for more details of the method and comments on the constraints $D_1 \succ 0$, see [1].

5 Hybrid Methods

A combination of Algorithm 3.2 and filterSQP are introduced to solve (1.1). Projection methods are globally convergent and hence potentially reliable, but often converge slowly, which can be very inefficient. SQP methods are reliable and have a second order rate of convergence, but require that the correct rank is known. We therefore consider hybrid methods in which the projection algorithm is used sparingly as a way of establishing the correct rank, whilst the filterSQP method is used to provide rapid convergence.

In order to ensure that each component method is used to best effect, it is important to transfer information from one method to the other. In particular, the result from one method is used to provide the initial data for the other, and vice versa. This mechanism has a fixed point property so that if one method finds a solution, then the other method is initialized with an iterate that also corresponds to the solution.

We will evaluate two different algorithms which differ in respect of how the rank $m^{(0)}$ is initialized. Algorithm 5.1 is expressed as follows: Given any data matrix $F \in \mathbb{R}^{n\times n}$, let s be some pre-selected positive integer number and ϵ some small number. Then the following algorithm solves (1.1)

Algorithm 5.1 *($F^{(0)} := F, s, \epsilon$):*
repeat projection method
until $m^{(l)} = m^{(l-j)} \quad j = 1, 2, \ldots, s.$
repeat
Apply one iteration of projection method;
$m^{(0)} := m^{(l)}$;
$x^{(0)} := x^{(l)}$; *($x^{(l)}$ from projection method)*
repeat filterSQP method;
until $\|D_2(x)\| \leq \epsilon$;
until $\|x^{(k)} - x^{(l)}\| \leq \epsilon$; *($x^{(k)}$ filterSQP method)*
return $(F^* := F^{(k)}, x^* := x^{(k)}, m^* := m^{(k)})$.

The choice of s is a compromise between two effects. If s is small then the rank may not be accurately estimated, but the number of (expensive) iterations taken in the projection method is small. On the other hand if s is large then a more accurate rank is obtained but the projection method needs more iterations.

In Algorithm 5.2, $m^{(0)}$ is supplied by the user. This approach avoids the initial sequence of projection iterations, but works well if the user is able to make a good estimate of the rank, which is often the case. Thus, we can express Algorithm 5.2 as follows: Given any data matrix $F \in \mathbb{R}^{n\times n}$, let ϵ be some small number; also choose $m^{(0)}$ as a small integer number. Then the following algorithm solves (1.1)

Algorithm 5.2 *($F^{(0)} := F, m^{(0)}, \epsilon$):*
repeat
repeat filterSQP method;
until $\|D_2(x)\| \leq \epsilon$;

$x^{(0)} := x^{(k)}$; *($x^{(k)}$ from filterSQP method)*
Apply one iteration of projection method;
$m^{(0)} := m^{(l)}$;
$x^{(0)} := x^{(l)}$; *($x^{(l)}$ from projection method)*
until $\|x^{(k)} - x^{(l)}\| \leq \epsilon$;
return $(F^* := F^{(k)}, x^* := x^{(k)}, m^* := m^{(k)})$.

6 Numerical Results

Two sets of randomly generated problems are solved. First, we solve (1.2) using filterSQP and the result are summarized in Table 6.1. Second, problem (1.1) solved using the four approaches mentions in Sections 2, 3 and 5 and the results summarized in Figure 6.1.

The algorithm has been tested on randomly generated matrices with values distributed between 10^{-3} and 10^3. A Fortran codes have been written to program solver for (4.4) using filterSQP. The termination criterion for algorithm is $\|F^{(k)} - F^{(k-1)}\| < 10^{-5}$. Table 6.1 summarizes the results for the filterSQP algorithm. The initial value $m^{(0)}$ is tabulated, and m is increased by one until the solution is found. The number of iterations is tabulated, and within this figure, it is found that fewer iterations are required as m increases. Also the initial value $m^{(0)}$ is rather arbitrary: a smaller value of $m^{(0)}$ would have given an even larger number of iterations. An asterisk indicates where the correct rank has been identified. In some cases, the final rank is $m^* + 1$ but the solution is within the required tolerance.

For (1.1), we solve set of problems using the four approaches: (i) Algorithm 3.1, (ii) the mixed-cone SDP formulation (2.5), (iii) Algorithm 5.1, and (iv) Algorithm 5.2. The SDPT3-3.0 code [21, 22] was used for approache (ii). The tests were done using MATLAB 6.5 on a Pentium IV PC with 512MB of RAM.

We solved problems with dimensions $n = 5, \ldots, 200$. The results were obtained by applying the methods as follows: A positive definite Hankel matrix A was formed randomly, then the matrix obtained was perturbed by adding random noise matrix S to A, where elements of S vary between -0.50 and 0.50. The problem is to recover the matrix before the noise was added. In all cases, we found the optimum to high accuracy, at least ten decimals. The results appear in Figures 6.1. We can see the correlation between the cputime and the size of the matrix A. For the projection algorithm, each iteration

n	m	nq	ϕ	n	m	nq	ϕ
5	1	10	52.0961	50	5	11	221.351
	2*	9	51.4198		6	9	213.765
10	3	11	72.9793		7	7	210.251
	4	8	69.4593		8*	8	209.179
	5*	7	68.2386		9	8	209.180
20	4	12	121.763	100	7	14	431.122
	5	8	115.923		83	10	423.231
	6	6	112.913		9	7	421.658
	7*	5	112.501		10	8	420.345
30	6	13	153.155		11	8	419.532
	7	8	148.103		12*	7	419.463
	8*	7	147.873				

Table 6.1: The filterSQP method

involves an eigensolution, which entails relatively expensive $O(n^3)$ calculations. Thus the projection algorithm is not competitive. For Algorithms 5.1 and 5.2, the housekeeping associated with each iteration is $O(n^2)$. Also, if care is taken, it is possible to calculate the gradient and Hessian in $O(n^2)$ operations. Thus each iteration is much less expensive than an iteration of the projection method. The total number of iterations required by the filterSQP in Table 6.1 is much greater than the total number of iterations required by the filterSQP in Algorithms 5.1 and 5.2.

Both hybrid algorithms are seen to be effective. As n increases, Algorithm 5.1 takes an increasing number of projection iterations before the rank settles down. We find it better to increase the value of s as the value of m increases. Once the projection iteration has settled down, the filter-SQP method finds the solution rapidly and no further projection steps are needed. Algorithm 5.2 requires a relatively large number of iterations in the first call of the filterSQP method, after which one projection step finds the correct rank, and the next call of filterSQP finds the solution in a few iterations. This is because of the good initial starting vector x given by the projection method. Because the projection steps in Algorithm 5.1 are relatively expensive, in some cases it is slower than SDPT3 algorithm, however in most cases the difference in computing time between these algorithms is not very significant. As is typical with interior-point methods, the number of iterations required by SDPT3 remains essentially constant, 14 to 20 iterations for SDPT3, independently of

Figure 6.1: Comparing all four approaches

the dimension of the problem. However the time consumed in each iterations is depend on the size of the problem and that what make a different.

Acknowledgements The author is very grateful to King Fahd University of Petroleum and Minerals for providing excellent research facilities.

References

[1] S. Al-Homidan. SQP algorithms for solving Toeplitz matrix approximation problem. *Numerical Linear Algebra with Applications*, 9:619–627, 2002.

[2] S. Al-Homidan. Toeplitz matrix approximation. *Mathematical Sciences Research Journal*, 6:104–111, 2002.

[3] S. Al-Homidan and R. Fletcher. Hybrid methods for finding the nearest Euclidean distance matrix. In D. Du, L. Qi, and R. Womersley, editors, *Recent*

Advances in Nonsmooth optimization, pages 1–7. World Scientific Publishing Co. Pte. Ltd., 1995.

[4] G. Cybenko. Moment problems and low rank toeplitz approximations. *Circuits Systems Signal processing*, 1:345–365, 1982.

[5] R. L. Dykstra. An algorithm for restricted least squares regression. *J. Amer. Stat.*, 78:839–842, 1983.

[6] M. C. Ferris, M. P. Mesnier, and J. J. Moré. NEOS and Condor: solving optimization problems over the Internet. *ACM Transactions on Mathematical Software*, 26(1):1–18, 2000. URL: http://www-neos.mcs.anl.gov.

[7] R. Fletcher. Semidefinite constraints in optimization. *SIAM J. Control Optim.*, 23:493–513, 1985.

[8] R. Fletcher and S. Leyffer. User manual for filtersqp. Technical Report NA/181, Univerity of Dundee Numerical Analysis Report, Dundee, Scotland, 1999.

[9] K. M. Grigoriadis, A. E. Frazho, and R. E. Skelton. Application of alternating convex projection methods for computing of positive Toeplitz matrices. *IEEE Trans. Signal Processing*, 42:1873–1875, 1994.

[10] S. P. Han. A successive projection method. *Math. Programming*, 40:1–14, 1988.

[11] C. Helmberg. URL: http://www.zib.de/helmberg/semidef.html.

[12] T. Kailath. A view of three decades of linear filtering theory. *IEEE Transactions on Information Theory*, IT-20:145–181, 1974.

[13] S. Y. Kung. Toeplitz approximation method and some applications. In *Internat, Sympos. On Mathematical Theory of Networks and Systems V. IV.* Western Periodicals Co., Morth Hollywood, CA, 1981, 1981.

[14] S. Oh and R. J.Marks II. Alternating projection onto fuzzy convex sets. In *Proc. IEEE*, volume 81, pages 148–155. 1993.

[15] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.

[16] S.Cabay and R. Meleshko. A weakly stable algorithm for pade' approximants and the inversion of hankel matrices. *SIAM J. Matrix Analysis and Appl.*, 14:735–765, 1992.

[17] R. Skelton. The jury test and covariance control. In *Proceedings of the Symposium on Fundamentals of Discrete Time Systems, publisher =.*

[18] Y. Suffridge and T. Hayden. Approximation by a hermitian positive semi–definite toeplitz matrix. *SIAM J. on Matrix Analysis and Applications*, 14:721–734, 1993.

[19] C. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1992.

[20] M. J. Todd. Semidefinite optimization. *Acta Numer.*, 10:515–560, 2001.

[21] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. *SIAM J. Optim.*, 8:769–796, 1998.

[22] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3, 2003. Computational semidefinite and second order cone programming: the state of the art.

[23] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[24] J. VonNeumman. *Functional Operators II, The geometry of orthogonal spaces*. Annals of Math. studies No.22, Princeton Univ. Press., 1950.

[25] N. Wiener. On factorization of matrices. *Comm. Math. Helv.*, 29:97–111, 1955.

[26] A. Willsky. *Digital Signal Processing and Control and Estimation Theory*. MIT Press: Cambridge: Mass,, 1979.

[27] H Wolkowicz, R. Saigal, and L Vandenberghe, editors. *Handbook of semidefinite programming, Theory, algorithms, and applications*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000.