

# Sampling method for semidefinite programs with nonnegative Popov function constraints\*

Anders Hansson<sup>†</sup>

Lieven Vandenberghe<sup>‡</sup>

## Abstract

An important class of optimization problems in control and signal processing involves the constraint that a Popov function is nonnegative on the unit circle or the imaginary axis. Such a constraint is convex in the coefficients of the Popov function. It can be converted to a finite-dimensional linear matrix inequality via the Kalman-Yakubovich-Popov lemma. However, the linear matrix inequality reformulation requires an auxiliary matrix variable and often results in a very large semidefinite programming problem. Several recently published methods exploit problem structure in these semidefinite programs to alleviate the computational cost associated with the large matrix variable. These algorithms are capable of solving much larger problems than general-purpose semidefinite programming packages. In this paper we address the same problem by presenting an alternative to the linear matrix inequality formulation of the nonnegative Popov function constraint. We sample the constraint to obtain an equivalent set of inequalities of low dimension, thus avoiding the large matrix variable in the linear matrix inequality formulation. Moreover, the resulting semidefinite program has constraints with low-rank structure, which allows the problems to be solved efficiently by existing semidefinite programming packages. The sampling formulation is obtained by first expressing the Popov function inequality as a sum-of-squares condition imposed on a polynomial matrix and then converting the constraint into an equivalent finite set of interpolation constraints. A complexity analysis and numerical examples are provided to demonstrate the performance improvement over existing techniques.

## 1 Introduction

A Popov function is a rational matrix function of the form

$$\mathcal{F}_M(z) = \begin{bmatrix} ((1/\bar{z})E - A)^{-1}B \\ I \end{bmatrix}^H M \begin{bmatrix} (zE - A)^{-1}B \\ I \end{bmatrix}, \quad (1)$$

defined for all complex  $z$  with  $\det(zE - A) \neq 0$  and  $\det(\bar{z}^{-1}E - A) \neq 0$ . The symmetric matrix  $M \in \mathbf{R}^{(n+m) \times (n+m)}$  is sometimes called the *center matrix* of the Popov function. The other parameters in the definition are the matrices  $A, E \in \mathbf{R}^{n \times n}$  and  $B \in \mathbf{R}^{n \times m}$ .

A variety of important system properties can be expressed in the form

$$\mathcal{F}_M(z) \succeq 0 \quad \forall z \in \mathbf{C} : |z| = 1, \quad (2)$$

---

\*This work was carried out while the first author was a Visiting Professor at University of California, Los Angeles. The research of the second author is supported in part by the National Science Foundation under Grant No. 1128817.

<sup>†</sup>A. Hansson is with the Division of Automatic Control, Linköping University, Sweden, e-mail: hansson@isy.liu.se.

<sup>‡</sup>L. Vandenberghe is with the Electrical Engineering Department, University of California, Los Angeles, USA, e-mail: lieven.vandenberghe@ucla.edu.

where the inequality  $\mathcal{F}_M(z) \succeq 0$  means that  $\mathcal{F}_M(z)$  is Hermitian positive semidefinite. We will call this type of constraint a *Popov function constraint*. In applications one is often interested in optimization problems of the form

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && \mathcal{F}_{M(x)}(z) \succeq 0 \quad \forall z \in \mathbf{C} : |z| = 1, \end{aligned} \tag{3}$$

in which a linear cost function is minimized, subject to a Popov function constraint with a center matrix  $M(x)$  that depends affinely on the optimization variable  $x \in \mathbf{R}^p$ , and possibly subject to additional convex constraints on  $x$ ; see [BDG<sup>+</sup>93, BEFB94, Pag95, Bal95, Jön96, MR97a, MKJR98, HAPS01, RP02, HSK03, WBV98, Dum07]. As we explain in Section 2, this problem is readily transformed to an equivalent problem

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && W(1/\bar{z})^H M(x) W(z) \succeq 0 \quad \forall z \in \mathbf{C} : |z| = 1, \end{aligned} \tag{4}$$

where  $W(z)$  is matrix polynomial satisfying

$$[-zE + A \quad B] W(z) = 0 \quad \forall z.$$

The main point of the paper is that such an optimization problem can be exactly reformulated as a semidefinite program (SDP)

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && W_i(1/\bar{\zeta})^H (M(x) - X) W_j(\zeta) = 0, \quad \zeta \in \mathcal{Z}_{ij}, \quad 1 \leq i \leq j \leq m \\ & && X \succeq 0 \end{aligned} \tag{5}$$

with variables  $x$  and  $X \in \mathbf{S}^{m+n}$ , where  $W_i$  denotes the  $i$ th column of  $W$ , and  $\mathcal{Z}_{ij}$  are finite sets of points in  $\mathbf{C} \cup \{\infty\}$ . We will refer to the SDP (5) as the sampled problem or sampled formulation. Notice the difference between (5) and a straightforward discretization of (4),

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && W(1/\bar{\zeta})^H M(x) W(\zeta) \succeq 0, \quad \zeta \in \mathcal{Z}, \end{aligned} \tag{6}$$

where  $\mathcal{Z}$  is a finite set of sample points on the unit circle. This is only an approximation of (4), while the sampled form (5) is exactly equivalent.

In control theory and signal processing, Popov function constraints are usually transformed to finite-dimensional inequalities by invoking the discrete-time Kalman-Yakubovich-Popov (KYP) lemma, which states that the Popov frequency domain inequality holds if and only if

$$\begin{bmatrix} A & B \\ E & 0 \end{bmatrix}^T \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \begin{bmatrix} A & B \\ E & 0 \end{bmatrix} + M \succeq 0 \tag{7}$$

for some symmetric matrix variable  $P$ . (There is also a continuous-time version, and more generally, the lemma holds for many other “stability” regions. In this paper we will focus on the discrete-time case. All the other cases can be transformed to this case, see [WHJ09].) By applying the KYP lemma the optimization problem in (5) is reformulated as an optimization problem of the form

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && \begin{bmatrix} A & B \\ E & 0 \end{bmatrix}^T \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \begin{bmatrix} A & B \\ E & 0 \end{bmatrix} + M(x) \succeq 0. \end{aligned} \tag{8}$$

This is an SDP problem in inequality form, with variables  $P$  and  $x$ , and is sometimes referred to as a KYP-SDP. In this paper we argue that in many cases the sampled formulation (5) is easier to solve than the equivalent KYP-SDP (8).

Before developing the details of the sampling formulation we review previous work on exploiting structure in KYP-SDPs.

**Algorithms** This work is motivated by the limits of general-purpose SDP solvers [Stu99, TTT03, BY05, YFK03, Bor99] when applied to (8). KYP-SDPs are usually quite dense, and include matrix variables, so the number of optimization variables can easily run into several ten thousand, even when the dimensions (number of states, inputs, and outputs) of the underlying control problem is not particularly large. The SDP (8) has in the order of  $p + n^2$  real variables and is therefore very expensive to solve via standard semidefinite programming software.

For this reason, several types of dedicated algorithms for KYP-SDPs have been proposed in the literature. These include cutting-plane methods that avoid the introduction of the matrix variable  $P$  by working directly with the frequency-domain inequality (2) [Par99, KMJ00, KM01, KMJ03, WKH08], interior-point methods based on new barrier functions for the convex set defined by the frequency domain inequalities [KM01, KM03, KM07], and customized implementations of standard interior-point methods for semidefinite programming [HV00, AV02, GHNV03, Hac03, GH03, VBW<sup>+</sup>05, LP04, RV06, HWH06, LV07].

With improvements in general-purpose SDP software packages, their capabilities of handling problem structure have advanced. It is therefore of interest to explore reformulations of Popov function constraints that result in problems that can be solved by general-purpose solvers at a cost that is comparable to the dedicated solvers listed above. (An example of such a preprocessor, based on a reformulation of the dual problem, is discussed in [WHJ09].) In particular, several general-purpose packages now exploit low-rank structure in the coefficient matrices of SDPs, in addition to sparsity. For example, SDPT3 [TTT10] offers this capability [FLH11]. Some of the fast algorithms for SDPs involving nonnegative scalar polynomials and Popov functions [LP04, RV06, RDV07, LV07] are based on exploiting low-rank structure and are therefore easily implemented with a general-purpose solver that exploits such structure. The main contribution of the paper is to extend the low-rank approach to Popov function constraints for *multi-input* systems. We present a reformulation that allows the problem to be solved at a cost in the order of  $n^3$  floating point operations per iteration (for fixed  $m$  and  $p$ ) using existing packages that exploit low-rank structure.

**Organization** The remaining part of this paper is organized as follows. Next we will define some notation and make some standing assumptions that will hold throughout the paper. In Section 2 we revisit the KYP lemma and discuss Popov functions in more detail. Here we will derive the sampling formulation and discuss its computational complexity. In Section 3 we recapitulate some common methods for modeling based on orthogonal basis functions. They are especially suitable when considering sampling, since they often result in well-conditioned problem formulations. In Section 4 we present two applications of Popov function inequalities: the identification of frequency domain spectra and stability analysis using integral quadratic constraints. We carry out numerical experiments for the two examples and we demonstrate that for larger problems the sampling method in many cases outperforms other solution methods. Finally, in Section 5 we make some concluding remarks and discuss directions for future research.

**Notation and assumptions**  $\mathbf{R}$  and  $\mathbf{C}$  denote the sets of real and complex numbers, respectively, and  $\mathbf{S}^n$  denotes the set of real symmetric matrices of order  $n$ . The symbols  $\succeq$  and  $\succ$  are used for matrix inequalities.  $\bar{A}$  is the complex conjugate of the matrix  $A$ ;  $A^H = \bar{A}^T$  is the complex conjugate transpose.  $\text{tr } X$  is the trace of the matrix  $X$ . We use  $Y = \mathbf{diag}(X_1, X_2, \dots, X_p)$  to denote the block-diagonal matrix with the matrices  $X_i$  as its diagonal blocks. The matrices  $X_i$  are not necessarily square. If  $T(z)$  is a polynomial matrix with  $r$  columns, then we define  $T_*(z)$  as the polynomial matrix

$$T_*(z) = \mathbf{diag}(z^{n_1}, z^{n_2}, \dots, z^{n_r})T(1/\bar{z})^H$$

where  $n_i$  is the column degree of column  $i$  of  $T(z)$ . In other words, if  $T_i(z) = a_0 + a_1z + \dots + a_{n_i}z^{n_i}$  is column  $i$  of  $T(z)$ , with  $a_{n_i} \neq 0$ , then  $T_*(z)$  has as its  $i$ th row the polynomial row vector

$$a_0^H z^{n_i} + a_1^H z^{n_i-1} + \dots + a_{n_i-1}^H z + a_{n_i}^H.$$

We assume that the  $n \times (n+m)$  pencil  $[-zE + A \quad B]$ , defined by the matrices  $A, B, E$  in the Popov function (1), has full row rank for all  $z$  and that  $E$  is invertible. Hence the pencil describes a controllable descriptor system without any algebraic constraints. It is also assumed that  $zE - A$  is invertible on the unit circle. This can always be achieved with a preliminary state-feedback. Notice that Popov function constraints are invariant under state-feedback.

## 2 The KYP lemma

The discrete-time KYP lemma states that the Popov function  $\mathcal{F}_M(z)$  defined in (1) is positive semidefinite on the unit circle if and only if there exists a matrix  $P \in \mathbf{S}^n$  that satisfies the LMI (7). In this section we develop an equivalent form of the KYP lemma that is better suited for semidefinite programming algorithms. The formulation is based on the interpretation of the KYP lemma as a sum-of-squares characterization of nonnegative Popov functions [Nes00, GHN+02].

### 2.1 Reformulated KYP lemma

Let  $\mathcal{A} : \mathbf{S}^n \rightarrow \mathbf{S}^{m+n}$  be the linear mapping

$$\mathcal{A}(P) = \begin{bmatrix} A & B \\ E & 0 \end{bmatrix}^T \begin{bmatrix} P & 0 \\ 0 & -P \end{bmatrix} \begin{bmatrix} A & B \\ E & 0 \end{bmatrix}$$

and let  $\mathcal{L} = \{\mathcal{A}(P) \mid P \in \mathbf{S}^n\}$  be its range. It is shown in [HSK99, p.340] that  $\mathcal{L}$  is the space of the center matrices of identically zero Popov functions:

$$\mathcal{L} = \{Q \in \mathbf{S}^{n+m} \mid \mathcal{F}_Q(z) = 0 \forall z \in \mathbf{C}\}. \quad (9)$$

(See also [IOW99].) Using this result we can rephrase the KYP lemma as follows. The standard version of the KYP lemma states that  $\mathcal{F}_M(z) \succeq 0$  on the unit circle if and only if  $\mathcal{A}(P) + M \succeq 0$  for some  $P$ . Equivalently, the matrix  $X = \mathcal{A}(P) + M$  satisfies

$$M - X \in \mathcal{L}, \quad X \succeq 0. \quad (10)$$

We conclude that the Popov function  $\mathcal{F}_M(z)$  is positive semidefinite on the unit circle if and only if there exists an  $X$  that satisfies (10).

We can use the geometric form of the KYP lemma to write problem (3) as

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && M(x) - X \in \mathcal{L} \\ & && X \succeq 0. \end{aligned} \tag{11}$$

This is an SDP in the variables  $X$  and  $x$ , with the equality constraint expressed geometrically, without choosing a specific parametrization of  $\mathcal{L}$ . It can be shown that if  $(A, B)$  is controllable, then  $\dim \mathcal{L} = n(n+1)/2$ , *i.e.*, the nullspace of  $\mathcal{A}$  has dimension zero [BV03].

Various equivalent forms of this SDP can now be derived by choosing different parameterizations of  $\mathcal{L}$ . If we use the range space parametrization  $\mathcal{L} = \{\mathcal{A}(P) \mid P \in \mathbf{S}^n\}$ , we obtain the standard KYP-SDP (8). In the next section we develop an alternative minimal ‘nullspace’ parametrization  $\mathcal{L} = \{Q \mid \mathcal{B}(Q) = 0\}$  where  $\mathcal{B} : \mathbf{S}^{m+n} \mapsto \mathbf{R}^r$  is a linear mapping

$$\mathcal{B}(X) = (\text{tr}(A_1 X), \dots, \text{tr}(A_r X)).$$

A simple dimension argument shows that this should be possible with

$$r = \frac{(n+m)(n+m+1)}{2} - \dim \mathcal{L} = mn + \frac{m(m+1)}{2}. \tag{12}$$

linear equalities. Using the nullspace parametrization of  $\mathcal{L}$ , the SDP (11) and its dual can be written as

$$\begin{aligned} & \text{minimize} && c^T x && \text{maximize} && b^T y \\ & \text{subject to} && Gx + \mathcal{B}(X) = b && \text{subject to} && \mathcal{B}^{\text{adj}}(y) \preceq 0 \\ & && X \succeq 0, && && G^T y = c, \end{aligned} \tag{13}$$

where  $\mathcal{B}^{\text{adj}}(y) = \sum_{i=1}^r y_i A_i$  is the adjoint of  $\mathcal{B}$ . If  $M(x)$  is given as  $M(x) = M_0 + \sum_{i=1}^p x_i M_i$  then  $G$  and  $b$  in (13) are defined as

$$G_{ik} = -\text{tr}(A_i M_k), \quad b_i = \text{tr}(A_i M_0), \quad i = 1, \dots, r, \quad k = 1, \dots, p. \tag{14}$$

In the semidefinite programming literature, the primal problem in (13) is called an SDP in *standard form with free variables* (the variables  $x$  are ‘free’, *i.e.*, do not have to be nonnegative).

One can also eliminate the equality constraint in the dual problem in (13) by expressing the dual variable as  $y = Fu - h$ , where  $F$  has  $r - p$  columns that span the nullspace of  $G^T$ . This gives

$$\begin{aligned} & \text{minimize} && \text{tr}(CX) && \text{maximize} && d^T u \\ & \text{subject to} && \mathcal{D}(X) = d && \text{subject to} && \mathcal{D}^{\text{adj}}(u) \preceq C \\ & && X \succeq 0, && && \end{aligned} \tag{15}$$

with  $C = \mathcal{B}^{\text{adj}}(h)$ ,  $\mathcal{D}(X) = F^T \mathcal{B}(X)$ , and  $d = F^T b$ . The primal problem is a standard form SDP with  $r - p$  equality constraints; the dual problem is an inequality form SDP with a vector variable  $u$  of dimension  $r - p$ .

A nullspace representation of  $\mathcal{L}$  can be derived by eliminating  $P$  from the range space parametrization. However, in the following sections we will see that a more direct derivation provides a nullspace representation with useful low-rank structure in the coefficients  $A_i$ .

## 2.2 Polynomial Popov functions

Consider a right co-prime Matrix Fraction Description (MFD) of  $(zE - A)^{-1}B$ :

$$(zE - A)^{-1}B = N(z)D(z)^{-1}$$

where  $N(z)$  and  $D(z)$  are  $n \times m$  and  $m \times m$  polynomial matrices with real coefficients and  $D(z)$  is column reduced. The column degrees  $n_i$ ,  $i = 1, 2, \dots, m$ , of  $D(z)$  are the controllability indexes (hence they satisfy  $n_1 + \dots + n_m = n$ ) and the column degrees of  $N(z)$  are equal to  $n_i - 1$ ; see [Kai80, chapter 6]. The columns of the polynomial matrix

$$W(z) = \begin{bmatrix} N(z) \\ D(z) \end{bmatrix}$$

have column degree  $n_i$  and form a minimal basis of the nullspace of the controllability pencil  $\begin{bmatrix} -zE + A & B \end{bmatrix}$ . Several numerically reliable algorithms for computing the basis matrix  $W(z)$  have been developed (see, *e.g.*, [Pat81, BV87]) and this can be done in cubic complexity in  $n$ . Substituting the MFD in the definition of  $\mathcal{F}_Q$  we obtain

$$\mathcal{F}_Q(z) = \begin{bmatrix} N(1/\bar{z})D(1/\bar{z})^{-1} \\ I \end{bmatrix}^H Q \begin{bmatrix} N(z)D(z)^{-1} \\ I \end{bmatrix}.$$

Now define

$$\begin{aligned} \mathcal{P}_Q(z) &= \mathbf{diag}(z^{n_1}, \dots, z^{n_m})D(1/\bar{z})^H \mathcal{F}_Q(z)D(z) \\ &= \mathbf{diag}(z^{n_1}, \dots, z^{n_m})W(1/\bar{z})^H QW(z) \\ &= W_*(z)QW(z). \end{aligned} \tag{16}$$

This is a polynomial matrix with  $i, j$  element of degree  $n_i + n_j$  or less. We will refer to  $\mathcal{P}_Q(z)$  as the *polynomial Popov function* with central matrix  $Q$ . Clearly,  $\mathcal{F}_Q(z) = 0 \forall z \in \mathbf{C}$  if and only if

$$\mathcal{P}_Q(z) = 0 \quad \forall z \in \mathbf{C}, \tag{17}$$

and therefore  $\mathcal{L} = \{Q \mid \mathcal{P}_Q(z) = 0 \forall z \in \mathbf{C}\}$ .

## 2.3 Sampled constraint

We now reduce the equality (17) in  $Q$  to a finite set of equalities

$$(\mathcal{P}_Q(\zeta))_{ij} = 0, \quad \zeta \in \mathcal{Z}_{ij}, \quad 1 \leq i, j \leq m, \tag{18}$$

where each  $\mathcal{Z}_{ij}$  is a finite set of points in  $\mathbf{C} \cup \{\infty\}$ . If  $\zeta = \infty$  the equality  $(\mathcal{P}_Q(\zeta))_{ij} = 0$  means that the highest order coefficient of  $(\mathcal{P}_Q(\zeta))_{ij}$  is zero. We refer to (18) as the *sampled form* of the constraint  $\mathcal{P}_Q(z) = 0$ .

Before deriving the conditions on the sets of sample points  $\mathcal{Z}_{ij}$  we note an important symmetry property. Define

$$p_{ij}(z) = z^{-n_i}(\mathcal{P}_Q(z))_{ij} = W_i(1/\bar{z})^H QW_j(z) \tag{19}$$

where  $W_i(z)$  denotes the  $i$ th column of  $W(z)$ . The functions  $p_{ij}(z)$  are Laurent polynomials of the form  $p_{ij}(z) = \sum_{k=-n_i}^{n_j} a_k z^k$ . The definition (19) implies that

$$p_{ji}(z) = \overline{p_{ij}(1/\bar{z})} \quad (20)$$

for all  $z$ . Hence,  $p_{ji}(z)$  is identically zero if and only if  $p_{ij}(z)$  is identically zero. It is therefore sufficient to impose constraints that guarantee that the upper triangular part of  $\mathcal{P}_Q(z)$  is zero. In other words,  $Q \in \mathcal{L}$  if and only if

$$p_{ij}(z) = 0 \quad \forall z, \quad 1 \leq i \leq j \leq m.$$

Each of these constraints can be converted as follows to a finite set of constraints

$$p_{ij}(\zeta) = 0, \quad \zeta \in \mathcal{Z}_{ij}. \quad (21)$$

(If  $\zeta = \infty$ , the equality means that the coefficient of  $z^{n_j}$  in  $p_{ij}(z)$  is zero; if  $\zeta = 0$ , it means that the coefficient of  $z^{-n_i}$  is zero). We consider the off-diagonal and the diagonal polynomials separately.

- $i < j$ . The polynomial  $z^{n_i} p_{ij}(z)$  has degree  $n_i + n_j$  or less, and hence at most  $n_i + n_j$  distinct zeros. Therefore  $z^{n_i} p_{ij}(z)$  is identically zero if it is zero at a set  $\mathcal{Z}_{ij}$  of  $n_i + n_j + 1$  or more distinct but otherwise arbitrary points in  $\mathbf{C} \cup \{\infty\}$ . Here we give the same interpretation as above to a zero at infinity, *i.e.*, the coefficient of  $z^{n_i+n_j}$  in  $z^{n_i} p_{ij}(z)$  is zero. Thus, if  $\mathcal{Z}_{ij}$  contains  $n_i + n_j + 1$  distinct elements, then the sampled constraint (21) is equivalent to  $p_{ij}(z) = 0 \forall z$ .
- $i = j$ . The polynomial  $z^{n_i} p_{ii}(z)$  has degree  $2n_i$  or less. By the same argument as above,  $z^{n_i} p_{ii}(z)$  is identically zero if it is zero at  $2n_i + 1$  distinct points in  $\mathbf{C} \cup \{\infty\}$ . However, the symmetry relation (20) implies that if  $p_{ii}(\zeta) = 0$ , then  $p_{ii}(1/\bar{\zeta}) = 0$ . Hence, any sampling point with  $|\zeta| \neq 1$  counts for two. We conclude that for  $i = j$  the sampled constraint (21) is equivalent to  $p_{ii}(z) = 0 \forall z$  if the points in  $\mathcal{Z}_{ii}$  are distinct,  $\zeta \neq 1/\bar{\gamma}$  for any two  $\zeta, \gamma \in \mathcal{Z}_{ii}$ , and

$$\sum_{\zeta \in \mathcal{Z}_{ii}} \delta(\zeta) = 2n_i + 1 \quad (22)$$

where  $\delta(\zeta) = 2$  if  $|\zeta| \neq 1$  and  $\delta(\zeta) = 1$  otherwise.

When the sets  $\mathcal{Z}_{ij}$  are chosen following these rules, the constraint (17) is equivalent to

$$W_i(1/\bar{\zeta})^H Q W_j(\zeta) = 0, \quad \zeta \in \mathcal{Z}_{ij}, \quad 1 \leq i \leq j \leq m, \quad (23)$$

and therefore the SDP (11) can be expressed as

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && W_i(1/\bar{\zeta})^H (X - M(x)) W_j(\zeta) = 0, \quad \zeta \in \mathcal{Z}_{ij}, \quad 1 \leq i \leq j \leq m \\ & && X \succeq 0. \end{aligned} \quad (24)$$

## 2.4 Nullspace representation of $\mathcal{L}$

The preceding discussion is independent of whether the problem data ( $Q$  and  $W$ ) are real. From now on we assume that  $Q$  and  $W$  are real and verify that the sampled constraint results in a number of linearly independent constraints equal to (12).

The function  $p_{ij}(z) = W_i(1/\bar{z})^H Q W_j(z)$  is complex-valued. We use the notation

$$p_{ij}(z) = \mathbf{tr}(F_{ij}(z)Q) + i \mathbf{tr}(G_{ij}(z)Q)$$

where  $i = \sqrt{-1}$  and  $F_{ij}(z)$  and  $G_{ij}(z)$  are real symmetric matrices defined as

$$F_{ij}(z) = \frac{1}{2} (U_j(z)U_i(1/\bar{z})^T + U_i(1/\bar{z})U_j(z)^T) \quad (25)$$

$$G_{ij}(z) = \frac{1}{2} (U_j(z)JU_i(1/\bar{z})^T - U_i(1/\bar{z})JU_j(z)^T) \quad (26)$$

with

$$U_i(z) = [ \Re W_i(z) \quad \Im W_i(z) ], \quad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Every sampling constraint in (23) corresponds to two equations

$$\mathbf{tr}(F_{ij}(\zeta)Q) = 0, \quad \mathbf{tr}(G_{ij}(\zeta)Q) = 0. \quad (27)$$

However, not all of these constraints are linearly independent because of symmetries in the definition of  $F_{ij}(z)$  and  $G_{ij}(z)$ . We have already noted the symmetry property (20). For  $i = j$  this implies that  $p_{ii}(z)$  is real ( $G_{ii}(z) = 0$ ) for  $z$  on the unit circle. This can also be seen from (26) which gives  $G_{ii}(z) = 0$  for  $z = 1/\bar{z}$  and  $i = j$ . From the fact that the coefficients of  $p_{ij}(z)$  are real, we also note that  $p_{ij}(\bar{z}) = \overline{p_{ij}(z)}$  for all  $z$ . In particular,  $p_{ij}(z)$  is real ( $G_{ij}(z) = 0$ ) for  $z$  on the real axis.

We now count the number of independent equations obtained by replacing each constraint in (23) by the linear equations (27) to get a nullspace representation  $\mathcal{L} = \{Q \mid \mathbf{tr}(A_k Q) = 0, k = 1, \dots, r\}$  with real symmetric  $A_k$ . We again distinguish the off-diagonal and the diagonal elements.

- $i < j$ . Recall that  $\mathcal{Z}_{ij}$  is a set of  $n_i + n_j + 1$  distinct points in  $\mathbf{C} \cup \{\infty\}$ .

For each real  $\zeta \in \mathcal{Z}_{ij}$  (including  $\zeta = \infty$ ), we obtain only one nontrivial constraint  $\mathbf{tr}(F_{ij}(\zeta)Q) = 0$  since  $p_{ij}(z)$  is real ( $G_{ij}(\zeta) = 0$ ) on the real axis. For each complex  $\zeta \in \mathcal{Z}_{ij}$  we get two constraints. However, without loss of generality we can choose the complex elements in  $\mathcal{Z}_{ij}$  in complex conjugate pairs. Since  $p_{ij}(\zeta)$  and  $p_{ij}(\bar{\zeta})$  are complex conjugates we obtain only two independent equalities (27) for each pair  $\zeta, \bar{\zeta}$ . The interpolation constraints (23) therefore reduce to a set of

$$n_i + n_j + 1$$

equality constraints with real coefficients.

- $i = j$ . Recall that  $\mathcal{Z}_{ii}$  is a set of distinct points in  $\mathbf{C} \cup \{\infty\}$  that satisfies (22). We again assume that the complex points are chosen in complex conjugate pairs.

We can partition  $\mathcal{Z}_{ii}$  in four sets, depending on whether  $\zeta$  is on the unit circle or not, and whether it is real or not. Define

$$\begin{aligned} k_r &= \#\{\zeta \in \mathcal{Z}_{ii} \mid |\zeta| = 1, \Im\zeta = 0\} \\ k_c &= \#\{\zeta \in \mathcal{Z}_{ii} \mid |\zeta| = 1, \Im\zeta > 0\} \\ l_r &= \#\{\zeta \in \mathcal{Z}_{ii} \mid |\zeta| \neq 1, \Im\zeta = 0\} \\ l_c &= \#\{\zeta \in \mathcal{Z}_{ii} \mid |\zeta| \neq 1, \Im\zeta > 0\}. \end{aligned}$$

If  $\infty \in \mathcal{Z}_{ii}$  it is counted as a real point (*i.e.*, included in  $l_r$ ). Since complex points in  $\mathcal{Z}_{ii}$  come in conjugate pairs, the condition (22) gives

$$k_r + 2k_c + 2l_r + 4l_c = 2n_i + 1.$$

This means that  $k_r = 1$ , *i.e.*, either  $1 \in \mathcal{Z}_{ii}$  or  $-1 \in \mathcal{Z}_{ii}$ , but not both. Therefore  $k_c + l_r + 2l_c = n_i$ . We now count the number of equality constraints.

For each real point  $\zeta$ , we obtain one equality constraint in (27). For each pair  $\zeta, \bar{\zeta}$  with  $|\zeta| = 1$ , we also obtain one equality constraint since  $p_{ii}(z)$  is real on the unit circle and equal to  $p_{ii}(\bar{z})$ . For each pair  $\zeta, \bar{\zeta}$  with  $|\zeta| \neq 1$ , we obtain two equality constraints. The interpolation constraint (23) for  $i = j$  therefore reduces to

$$k_r + k_c + l_r + 2l_c = n_i + 1$$

equality constraints with real symmetric coefficients.

Adding the constraints for  $1 \leq i \leq j \leq m$  gives a total of

$$\sum_{i=1}^m \sum_{j=1}^{i-1} (n_i + n_j + 1) + \sum_{i=1}^m (n_i + 1) = mn + \frac{m(m+1)}{2} \quad (28)$$

equality constraints. This number coincides with (12) and therefore the entire set of equalities is linearly independent.

## 2.5 Computational complexity

We now discuss the reduction in complexity obtained from the sampled formulation. To provide some perspective, we first note the per-iteration-complexity of a general-purpose SDP solver applied to the KYP-SDP in its common form (8). If the solver does not exploit any structure in the problem data, then the total count of floating-point operations per iteration includes terms

$$\left(p + \frac{n(n+1)}{2}\right)(m+n)^3, \quad \left(p + \frac{n(n+1)}{2}\right)^2(m+n)^2, \quad \left(p + \frac{n(n+1)}{2}\right)^3.$$

We are interested in applications where the  $n^6$  term dominates. In other applications, with  $n$  small compared to  $m$  or  $p$ , other terms can be dominant and the sampling method may not provide any advantage over a general-purpose method directly applied to (8). We will therefore assume that  $p \ll n^2$  and  $m \ll n$ .

We now compare the complexity of solving the SDPs (13) and (15) that result from the sampled formulation via a general-purpose SDP solver that does not exploit any structure in the coefficients

$A_k$  or via a solver that exploits low-rank structure. We exclude from the analysis the cost of computing and sampling the polynomial matrix  $W(z)$  (for algorithms to do this, see [Pat81, BV87]), and we do not explicitly take into account that the sampling points are chosen in complex conjugated pairs. These simplifications do not change the overall order of the complexity estimates.

The size of the SDPs (13) depends on the dimensions  $p$ ,  $n$ ,  $m$  of the original KYP-SDP (3). The vector variable  $x$  has dimension  $p$ , the matrix variable  $X$  has size  $m + n$ , and there are  $r = mn + m(m + 1)/2$  equality constraints. To make the interpretation of the analysis easier we will assume  $p = O(n)$  and  $r = O(mn)$  and express the flop counts in terms of  $m$  and  $n$ .

We focus on the SDP formulation (13). Alternatively, one can consider the standard form SDP (15) which has  $q = r - p$  equality constraints and no free variables. Under our assumptions  $q = O(mn)$ , so the number of equalities of the SDP is of the same order as (13) and the complexity estimates are roughly the same.

**General-purpose solver** We first estimate the complexity of a general-purpose solver. Each matrix  $A_i$  is of the form  $F_{ij}(\zeta)$  or  $G_{ij}(\zeta)$  as defined in (25)–(26), and therefore can be computed in order  $(m + n)^2$  operations. The total cost to compute the  $r$  coefficients  $A_i$  is order  $r(m + n)^2$ . The cost of forming the matrix  $G$  and the vector  $b$  in (14) is order  $rp(m + n)^2$  in general (for dense  $M_k$ ). Therefore the total cost of computing the parameters  $A_i$ ,  $G$ ,  $b$  in the standard form SDP is of order  $r(m + n)^2 + rp(m + n)^2$ . However, if the matrices  $M_k$  have low rank and appear in factored form, which is very common in applications, the cost of computing the coefficients  $G$ ,  $b$  is only  $rp(m + n)$ , and the total cost of setting up the problem is reduced to order  $r(m + n)^2 + rp(m + n)$ , where the second term can be neglected since we assume that  $p = O(n)$ . If the coefficients are very sparse, the cost of computing of  $G$  and  $b$  is even less.

The dominating cost in solving SDPs using interior-point (IP) methods is the cost of forming and solving the equations for the search directions in each iteration (typically for about 20–50 iterations). When a primal or dual algorithm is used, or a primal-dual algorithm based on the Nesterov-Todd scaling, these equations take the general form

$$\begin{aligned} -S^{-1}\Delta X S^{-1} + \mathcal{B}^{\text{adj}}(\Delta y) &= R_1 \\ G^T \Delta y &= r_2 \\ \mathcal{B}(\Delta X) + G\Delta x &= r_3 \end{aligned}$$

for some right-hand sides  $R_1$ ,  $r_2$ ,  $r_3$ , and a positive definite  $S$  (see [VBW<sup>+</sup>05, appendix A]). The equations are usually solved by eliminating  $\Delta X$  from the first equation and solving a reduced system

$$\begin{bmatrix} H & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \end{bmatrix} = \begin{bmatrix} r_3 + \mathcal{B}(SR_1S) \\ r_2 \end{bmatrix} \quad (29)$$

where  $H$  is defined by  $H\Delta y = \mathcal{B}(S\mathcal{B}^{\text{adj}}(\Delta y)S)$ , *i.e.*,

$$H_{kl} = \mathbf{tr}(A_k S A_l S), \quad k, l = 1, \dots, r$$

The cost of one iteration is dominated by computing the matrix  $H$  and solving the system (29).

The assembly of  $H$  can be carried out in several different ways with approximately the same complexity. For example, one can first compute the  $r$  matrix products  $A_k S$  at a cost of  $r(m + n)^3$

flops, and then make the inner products of the scaled matrices at a total cost of  $r^2(m+n)^2$  flops. Assuming  $r = O(mn)$  this results in a flop count in the order of

$$m^2n^2(m+n)^2 \quad (30)$$

for constructing  $H$ . The cost of factoring the coefficient matrix of (29) is order  $(r+p)^3$ , *i.e.*,  $m^3n^3$  if we assume  $p = O(n)$  and  $r = O(mn)$ . The cost (30) of assembling  $H$  therefore outweighs the cost of factoring and solving the reduced system (29).

**Solver that exploits low-rank structure** A solver that exploits low-rank structure can take advantage of the fact that the coefficient matrices  $A_k$  can be expressed as

$$A_k = \tilde{U}_k \tilde{V}_k^T + \tilde{V}_k \tilde{U}_k^T$$

with  $\tilde{U}_k$  and  $\tilde{V}_k$  column vectors or two-column matrices. This can be seen from (25)–(26). If we keep  $A_k$  in this factored form the calculation of  $H$  can be performed more efficiently because

$$\begin{aligned} H_{kl} &= \mathbf{tr} \left( (\tilde{U}_k \tilde{V}_k^T + \tilde{V}_k \tilde{U}_k^T) S (\tilde{U}_l \tilde{V}_l^T + \tilde{V}_l \tilde{U}_l^T) S \right) \\ &= 2 \mathbf{tr} \left( (\tilde{V}_k^T S \tilde{U}_l) (\tilde{V}_l^T S \tilde{U}_k) \right) + 2 \mathbf{tr} \left( \tilde{V}_k^T S \tilde{V}_l (\tilde{U}_l^T S \tilde{U}_k) \right). \end{aligned}$$

Hence  $H$  can be computed by first forming  $2r$  products  $S\tilde{U}_k$  and  $S\tilde{V}_k$  at at cost of  $r(m+n)^2$ , then computing the  $r(r+1)/2$  matrix products  $\tilde{V}_k^T(S\tilde{U}_l)$ ,  $\tilde{V}_k^T(S\tilde{V}_l)$  and  $\tilde{U}_l^T(S\tilde{U}_k)$ , at a cost of  $r^2(m+n)$ , and then computing the coefficients of  $H$  via order  $r^2$  inner products of  $2 \times 2$  matrices. Using this method, the cost of assembling  $H$  is of order

$$m^2n^2(m+n), \quad (31)$$

again assuming  $r = O(mn)$ . Hence, compared with a general-purpose implementation (30) the complexity as a function of  $m$  and  $n$  is reduced by one order. The cost of factoring  $H$  is identical to the cost for a general-purpose implementations, *i.e.*,  $O(m^3n^3)$ . Hence, at this point the cost of solving the reduced system dominates the cost of assembling  $H$  and the overall complexity per iteration is order

$$m^3n^3. \quad (32)$$

If  $m = 1$ , we obtain the same  $O(n^3)$  complexity per iteration as the algorithms for scalar nonnegative polynomials published in [AV02, LP04, RV06]. For the special case of a multivariable finite impulse response system we can also compare the cost of the sampling method with the fast barrier method for optimization over nonnegative trigonometric matrix polynomials discussed by Genin *et al.* [GHN03]. Consider the optimization problem

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && \sum_{k=-N}^N C_k(x) z^k \succeq 0 \quad \forall z \in \mathbf{C} : |z| = 1, \end{aligned} \quad (33)$$

with  $m \times m$  coefficients  $C_{-k}(x) = C_k(x)^T$  that depend linearly on the optimization variable  $x \in \mathbf{R}^p$ . The function on left-hand side of the constraint is a Popov function with central matrix

$$M(x) = \begin{bmatrix} 0 & \cdots & 0 & C_N(x) \\ 0 & \cdots & 0 & C_{N-1}(x) \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & C_1(x) \\ C_{-N}(x) & \cdots & C_{-1}(x) & C_0(x) \end{bmatrix} \in \mathbf{S}^{(N+1)m}$$

and system matrices

$$A = \begin{bmatrix} 0 & I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & I & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & I & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & I \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \in \mathbf{R}^{mN \times mN}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ I \end{bmatrix} \in \mathbf{R}^{mN \times m}.$$

The optimization problem (33) can be expressed as a standard SDP (15) with  $\mathcal{D}$  a linear mapping from  $\mathbf{S}^{m(N+1)}$  to  $\mathbf{R}^q$  and  $q = m(m+1)/2 + m^2N - p$ . If  $\mathcal{D}$  is chosen so that  $\mathcal{D}^{\text{adj}}(u)$  is block-Toeplitz [GHN03], displacement rank techniques can be used to speed up the calculations in a dual barrier method and, in particular, the cost of evaluating the Hessian of the barrier function. The complexity of the fast algorithm for assembling the Hessian includes a term  $O(q^2m^2N) = O(m^6N^3)$  [GHN03, page 75]. Substituting  $n = mN$  in (31) and (32) shows that the complexity of the sampling method is  $O(m^5N^3)$  for forming  $H$  and  $O(m^6N^3)$  for solving the system. The overall cost per iteration is therefore similar to the method in [GHN03].

### 3 Orthogonal basis models

In this section we revisit some orthogonal basis models which are suitable when modeling linear systems. This is motivated by the fact that for these types of basis functions a straightforward uniform sampling on the unit circle, possibly after a suitable transformation, results in a well-conditioned system of sampled constraints.

#### 3.1 FIR models

Finite Impulse Response (FIR) models are simple but often give good intuition. A multi-input-multi-output (MIMO) FIR model transfer function can be expressed as

$$G(z) = [C \quad D] \begin{bmatrix} (zI - A)^{-1}B \\ I \end{bmatrix}$$

where  $C$  and  $D$  are general matrices and

$$A = \mathbf{diag}(A_1, \dots, A_m), \quad B = \mathbf{diag}(B_1, \dots, B_m)$$

with

$$A_i = \begin{bmatrix} 0_{1 \times (n_i-1)} & 0 \\ I_{n_i-1} & 0_{(n_i-1) \times 1} \end{bmatrix}, \quad B_i = \begin{bmatrix} 1 \\ 0_{(n_i-1) \times 1} \end{bmatrix}, \quad i = 1, \dots, m.$$

We realize that  $(zI - A)^{-1}B = \Psi(z) \mathbf{diag}(z^{-n_1}, \dots, z^{-n_m})$ , where

$$\Psi(z) = \mathbf{diag} \left( [z^{n_1-1} \ \dots \ z \ 1]^T, [z^{n_2-1} \ \dots \ z \ 1]^T, \dots, [z^{n_m-1} \ \dots \ z \ 1]^T \right)$$

and hence the MFD description of the FIR model has a very simple structure with controllability indexes  $n_i$ . The polynomial Popov function associated with an FIR model has the form

$$\begin{bmatrix} \Psi(z) \\ \mathbf{diag}(z^{n_1}, \dots, z^{n_m}) \end{bmatrix}_* M \begin{bmatrix} \Psi(z) \\ \mathbf{diag}(z^{n_1}, \dots, z^{n_m}) \end{bmatrix}.$$

After a reordering of columns and rows in  $M$  the function may be written

$$\tilde{\Psi}_*(z) M \tilde{\Psi}(z)$$

where  $\tilde{\Psi}(z)$  is defined similarly as  $\Psi(z)$  but with diagonal blocks  $[z^{n_i} \ \dots \ 1]^T$ . From this we realize that for the FIR case the interpolation constraints only contain sub-blocks of the  $M$ -matrix. This means that the interpolation constraints will result in a linear system of equations in the entries of the  $M$ -matrix which is sparse and which can be set up in such a way that the associated coefficient matrix  $G$  in (13) is block-diagonal. We now turn our focus to an extension of FIR models which admits more accurate modeling with fewer terms in the series expansion.

### 3.2 Laguerre models

A Laguerre transfer function can be written as  $G(z) = CF(z)$  where  $C$  is a matrix of coefficients and  $F(z)$  is a basis vector of Laguerre functions  $F_k(z)$ ,

$$F(z) = \begin{bmatrix} F_1(z) \\ F_2(z) \\ \vdots \\ F_n(z) \end{bmatrix}, \quad F_k(z) = V_1(z)G_b(z)^{k-1},$$

with

$$V_1(z) = \frac{\sqrt{1-a^2}}{z-a}, \quad G_b(z) = \left( \frac{1-az}{z-a} \right)$$

and  $|a| < 1$ . The Laguerre functions are just like the FIR basis functions orthogonal with respect to the inner product on the unit circle. Hence they are well-suited for approximating a discrete time spectrum. A Popov function associated with the Laguerre model is

$$\mathcal{F}(z) = F_*(z)MF(z)$$

for a symmetric matrix  $M$ . To see this is of the form (1) let  $(A, B, C)$  be a realization of  $\mathcal{F}(z)$  and write  $\mathcal{F}(z) = [C \ 0] \begin{bmatrix} (zI - A)^{-1}B \\ I \end{bmatrix}$ . In case we are interested in matrix-valued Popov functions

we define  $F(z)$  to be block-diagonal with as its diagonal blocks basis vectors of Laguerre functions (possibly of different lengths and with different parameters  $a$ ).

Let  $\lambda$  and  $z$  be related via  $\lambda^{-1} = G_b(z)$ . This defines a bilinear transformation such that the unit circle is mapped to the unit circle [HdHW05]. We are interested in investigating when  $\mathcal{F}(z) \succeq 0$  for all  $|z| = 1$ . This is equivalent to

$$\frac{F_*(z)}{V_{1*}(z)} M \frac{F(z)}{V_1(z)} \succeq 0 \quad \forall z \in \mathbf{C} : |z| = 1.$$

Now we use the relation between  $\lambda$  and  $z$  to conclude that an equivalent condition is that

$$\check{\mathcal{F}}(z) = \check{F}_*(\lambda) M \check{F}(\lambda) \succeq 0 \quad \forall \lambda \in \mathbf{C} : |\lambda| = 1,$$

where

$$\check{F}(\lambda) = \begin{bmatrix} 1 \\ \lambda^{-1} \\ \lambda^{-2} \\ \vdots \\ \lambda^{-n+1} \end{bmatrix}$$

We see that we have transformed the Laguerre Popov function to an FIR Popov function. Notice that this is not the same transformation as in [HdHW05]. There they divide with  $\lambda$ , but we prefer to not do that, so that we get a direct term as in the previous subsection. It should be mentioned that there are other types of orthogonal basis models for which similar transformations to the FIR case can be done; see [HdHW05].

## 4 Applications

In this section we discuss different applications that can be formulated as problems of the form (11). For each application we present results of numerical experiments in which the sampling method is compared with other approaches. While some solvers exploit low-rank structure, YALMIP does not currently recognize it in the form we need. Hence in the numerical experiments we do not exploit the low-rank structure explained in Section 2.5, but we will see that good improvements in computational complexity are still achieved for the sampling method as compared to other methods.

### 4.1 Identification of frequency domain power spectra

In the first application we are interested in the identification of MIMO linear systems from measured power spectra; see e.g. [OMDS97]. For a given transfer function  $G(z)$  of dimension  $m \times m$  the power spectrum is defined as

$$\mathbf{S}(z) = G(z)G_*(z).$$

Hence the power spectrum must satisfy

$$\mathbf{S}(z) \succeq 0 \quad \forall z \in \mathbf{C} : |z| = 1, \tag{34}$$

a property often referred to as positive-realness. If we assume that  $G_*(z) = C(zE - A)^{-1}B + D$ , then the power spectrum can be written as a Popov function

$$\mathbf{S}(z) = \begin{bmatrix} (zE - A)^{-1}B \\ I \end{bmatrix}_* M \begin{bmatrix} (zE - A)^{-1}B \\ I \end{bmatrix}$$

with central matrix

$$M = [C \ D]^T [C \ D].$$

From this we see that the positive-real condition is a Popov function constraint and therefore convex in  $M$  for given  $A$ ,  $B$ , and  $E$ . In the method described in [OMDS97], the matrices  $A$ ,  $B$  and  $E$  are obtained from a subspace identification algorithm, and hence it is a valid assumption that these matrices are known. We will later on look at another example where they also are known.

Assume that we are given measurements of the power spectrum  $S_k$ ,  $k = 1, 2, \dots, N$ , and that we are interested in determining a transfer function  $G(z)$  such that

$$\mathbf{S}(\zeta_k) = S_k, \quad k = 1, 2, \dots, N.$$

This is a convex condition in the center matrix  $M$ . If we are given so many measurements of the power spectrum that we do not expect to find a perfect fit, we may instead minimize an objective like

$$\max_k \|\mathbf{S}(\zeta_k) - S_k\|_F^2,$$

which is a convex function of  $M$ . For practical purposes it makes sense to normalize by dividing with  $\|S_k\|_F^2$ , and this is the objective function, modulo an equivalent reformulation, that we minimize in the numerical experiments later on. Other frequency weightings are also possible.

An optimal  $M$  which approximates the measured power spectrum and satisfies the positive real condition can hence be found using the formulation in (11). After this has been done it remains to compute  $C$  and  $D$ . To this end let  $P$  and  $L$  solve

$$\mathcal{A}(P) + M = \begin{bmatrix} I & 0 \\ L & I \end{bmatrix}^H \begin{bmatrix} 0 & 0 \\ 0 & B^T P B + M_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ L & I \end{bmatrix} \quad (35)$$

where  $M = \begin{bmatrix} M_1 & M_{12} \\ M_{12}^T & M_2 \end{bmatrix}$ , which is an algebraic Riccati equation if  $L$  is eliminated. Let  $U(z) = (zE - A)^{-1}B$  and  $V(z) = \begin{bmatrix} U(z) \\ I \end{bmatrix}$ . By multiplying (35) with  $V(z)$  from the right and  $V_*(z)$  from the left and using the fact that

$$\begin{bmatrix} A & B \\ E & 0 \end{bmatrix} V(z) = \begin{bmatrix} zEU(z) \\ EU(z) \end{bmatrix}$$

it follows that the following spectral factorization holds:

$$\mathcal{F}(z) = S_*(z)(B^T P B + M_2)S(z)$$

where  $S(z) = I + LU(z)$ . Hence we may take  $G_*(z) = (B^T P B + M_2)^{1/2}S(z)$  which defines  $C$  and  $D$ .

In the experiments we will use a Laguerre basis for the transfer function. This fits into the above formulation by taking

$$M = [C_F \ 0]^T M_F [C_F \ 0]$$

and  $A = A_F$ ,  $B = B_F$ ,  $E = I$  and considering  $M_F \in \mathbf{S}^n$  as the optimization variable. Here  $(A_F, B_F, C_F)$  is a realization of the Laguerre basis vector or basis matrix  $F(z)$ . The constraints related to the objective may be defined in the  $z$ -domain whereas it is a good idea to transform the positive-real condition to the  $\lambda$ -domain as has already been described.

	$m = 1$		$m = 2$		$m = 3$		$m = 4$		$m = 5$	
	tot	sol								
$n = 2$	9.33	3.19	7.29	1.11	9.03	1.71	11.0	2.20	17.1	4.34
$n = 5$	10.1	1.29	14.5	2.28	27.1	7.01	55.8	14.3	121	37.4
$n = 10$	23.4	2.13	71.9	14.7	194	49.8	552	192	1170	355

Table 1: Results with the sampling method for the example of Section 4.1. The columns labeled ‘tot’ give the total time in seconds including YALMIP time, solver time, and setup time. The columns labeled ‘sol’ give the solver time only.

	$m = 1$		$m = 2$		$m = 3$		$m = 4$		$m = 5$	
	per iter	iter								
$n = 2$	0.11	28	0.04	28	0.05	32	0.08	27	0.14	32
$n = 5$	0.04	35	0.08	29	0.21	34	0.45	32	0.87	43
$n = 10$	0.07	30	0.44	33	1.42	35	3.83	50	8.66	41

Table 2: Time per iteration in seconds and number of iterations in the sampling method for the example in Section 4.1.

We now present some computational results. All implementations have been carried out in MATLAB R2010b using YALMIP [Löf04] and the solver SDPT3 [TTT10] with default settings unless otherwise specified. The computations have been run on an Intel Core i5 CPU M 250 4 GHz with 4 GB of RAM.

We generate random  $m \times m$  transfer functions  $G$  using the MATLAB function `drss`. However, in case the transfer function has poles with absolute values greater than 0.999, we move the poles to the nearest point with modulus 0.999. From these transfer functions we compute spectra  $\mathbf{S}$ , and we take the number of samples as  $N = 300$ . We consider values of  $n \in \{2, 5, 10\}$ , which is both the order of the random transfer function and the order of the Laguerre model, and we consider dimensions  $m$  in the range of 1 to 5. The value of the parameter  $a$  in the Laguerre basis has been taken as 0.3. We compare the sampling method as outlined in Section 4.1 with four other methods. For the sampling method we have made use of an add-on to YALMIP, [Löf09], which makes sure that the problem is parsed to SDPT3 in its standard primal form. The other four methods are all based on solving the KYP-SDP. This has been done with and without performing the transformation to the  $\lambda$ -domain. For both of these two cases the solver SDPT3 has been used both as it stands and in conjunction with a pre-processing step with the software STRUL, [FLH11]. This is a software which utilizes the fact that the  $P$ -variable stemming from the KYP lemma is a symmetric matrix variable with a low-rank and sparse basis. This is the same type of structure which is utilized in LMI Lab of the Robust Control Toolbox.

The resulting optimization problems become quite large, mainly because of the objective function, which is reformulated using an epigraph formulation resulting in  $N$  second order cone inequalities and associated variables. We report computational times for the sampling method in Table 1. We do not include the time for performing the spectral factorization, but only the time for setting up the optimization problem and solving it. We also separately report the solver time. All approaches except the sampling method had difficulties to converge for all problems except the ones of smallest dimension, and we therefore do not report the computational times for the other methods. In Table 2 we report the number of iterations and the time per iteration in the solver.

The YALMIP-time was between 1 and 6 seconds in all cases. The maximum number of iterations was exceeded for  $n = 10$  and  $m = 4$ . It is concluded that only the sampling method is viable for large scale problems.

## 4.2 Integral quadratic constraints

Consider a feedback connection given by

$$v = Gw + e, \quad w = \Delta(v)$$

where  $G$  is a causal and bounded linear time-invariant transfer function, and where  $\Delta$  is a bounded and causal operator on an appropriate function space  $\mathcal{H}$ , e.g.  $L_2^m[0, \infty)$  or  $\mathbb{I}_2^m[0, \infty)$ . We assume that the feedback connection is well-posed. Let  $\Pi$  be a bounded self-adjoint operator. We say that  $\Delta$  satisfies the Integral Quadratic Constraint (IQC) defined by  $\Pi$  if

$$\sigma_{\Pi}(v, \Delta(v)) = \left\langle \begin{bmatrix} v \\ \Delta(v) \end{bmatrix}, \Pi \begin{bmatrix} v \\ \Delta(v) \end{bmatrix} \right\rangle \geq 0, \quad \forall v \in \mathcal{H} \quad (36)$$

where  $\langle \cdot, \cdot \rangle$  is the inner product for the function space  $\mathcal{H}$ . For the case of  $\mathcal{H} = \mathbb{I}_2^m[0, \infty)$  we may take  $\Pi$  as a transfer function satisfying  $\Pi(z) = \Pi_*(z)$  on the unit circle.

The main result in IQC analysis says that if  $\Delta$  satisfies an IQC defined by  $\Pi$ , if a certain well-posedness assumption is fulfilled and if

$$\begin{bmatrix} G \\ I \end{bmatrix}_* \Pi \begin{bmatrix} G \\ I \end{bmatrix} \preceq -\epsilon I \quad (37)$$

for some  $\epsilon > 0$ , then the feedback connection is stable; see [MR97b] for the continuous time version and [Kao12] for the discrete-time case. The feasibility problem defined by (37) and (36), modulo an equivalent reformulation, is the feasibility problem that will be solved in the numerical experiments later on. Here we have also made the assumption that

$$\Pi = \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ (\Pi_{12})_* & \Pi_{22} \end{bmatrix}$$

is such that  $\Pi_{11} \succeq 0$  and  $\Pi_{22} \preceq 0$ , which is often the case in applications.

As an example we might consider the case when

$$\Delta = \mathbf{diag}(\delta_1 I_{m_1}, \dots, \delta_l I_{m_l})$$

(This is an abuse of notation in the sense that we mean that  $\Delta(v) = \Delta v$ .) We assume that  $\Delta$  is a constant such that  $|\delta_i| \leq 1$ ,  $i = 1, \dots, l$ . It is straightforward to verify that an IQC for this operator, i.e. that (36) holds, is given by

$$\Pi = \begin{bmatrix} X & Y \\ Y_* & -X \end{bmatrix}$$

where  $X = X_* \succeq 0$  and  $Y = -Y_*$ , and where  $X$  and  $Y$  should commute with  $\Delta$ . Because of this  $X$  and  $Y$  are block diagonal, i.e.

$$X = \mathbf{diag}(X_1, \dots, X_l), \quad Y = \mathbf{diag}(Y_1, \dots, Y_l)$$

where  $X_i$  and  $Y_i$  are  $m_i \times m_i$  matrices. Remember that  $\Pi$ , and then also  $X$  and  $Y$ , are transfer functions. In order to prove closed loop stability we will have to search for a  $\Pi$  such that the LMI condition is satisfied for all  $z$  on the unit circle, and there are many to search among. There are two obvious avenues to pursue. Either we sample the unit circle and consider a fine grid as an approximation to the original problem and hope for the best, or we parameterize  $\Pi$  and use the KYP lemma or exact sampling of the Popov function constraint. This parametrization can be done in many ways. One simple parametrization is to take

$$X_i = \phi_{i*} Q_i \phi_i, \quad Y_i = \phi_{i*} R_i \phi_i$$

where  $Q_i = Q_i^T \succeq 0$ , and  $R_i = -R_i^T$  are real-valued matrices, and where

$$\phi_i(z) = \left[ I_{m_i} \quad \frac{1-az}{z-a} I_{m_i} \quad \left( \frac{1-az}{z-a} \right)^2 I_{m_i} \quad \dots \quad \left( \frac{1-az}{z-a} \right)^{n_\phi} I_{m_i} \right]^T$$

is a matrix of Laguerre shifts. This is very close to the ideas in [MKJR98] for the continuous time case, where continuous time Laguerre shifts are used. Here  $a$  is a positive constant. The choice of this constant is not obvious, and with the proper choice less terms in the series expansion above are needed. We may then write

$$\Pi = \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix}_* \begin{bmatrix} Q & R \\ R^T & -Q \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix}$$

where  $Q = \mathbf{diag}(Q_1, \dots, Q_l)$ ,  $R = \mathbf{diag}(R_1, \dots, R_l)$ ,  $\Phi = \mathbf{diag}(\phi_1, \dots, \phi_l)$ . We now see that the stability criterion (37) can be written as

$$\begin{bmatrix} \Phi G \\ \Phi \end{bmatrix}_* \begin{bmatrix} Q & R \\ R^T & -Q \end{bmatrix} \begin{bmatrix} \Phi G \\ \Phi \end{bmatrix} \preceq -\epsilon I.$$

In order to use the KYP lemma we need to transform the left-hand side. To do this we first need a state-space description  $(A, B, C, D)$  such that

$$C(zI - A)^{-1}B + D = \begin{bmatrix} \Phi G \\ \Phi \end{bmatrix}.$$

This is straightforward but tedious. Then we are able to define a Popov function  $\mathcal{F}(z)$  such that

$$\mathcal{F}(z) = - \begin{bmatrix} \Phi G \\ \Phi \end{bmatrix}_* \begin{bmatrix} Q & R \\ R^T & -Q \end{bmatrix} \begin{bmatrix} \Phi G \\ \Phi \end{bmatrix} = \begin{bmatrix} (zI - A)^{-1}B \\ I \end{bmatrix}_* M \begin{bmatrix} (zI - A)^{-1}B \\ I \end{bmatrix}$$

by letting the center matrix in the Popov function be

$$M = - [C \ D]^T \begin{bmatrix} Q & R \\ R^T & -Q \end{bmatrix} [C \ D].$$

In case we are instead given a right MFD of the transfer function  $G$ , i.e.  $G(z) = N_G(z)D_G(z)^{-1}$ , where  $D_G(z)$  and  $N_G(z)$  are polynomial matrices we may proceed in a different way. We will specifically look at the case when  $G(z)$  is expressed in terms of a Laguerre basis as

$$G(z) = ZF(z)$$

where

$$Z = [Z_1 \quad Z_2 \quad \cdots \quad Z_n] = \begin{bmatrix} Z_{11} & \cdots & Z_{1n} \\ \vdots & \ddots & \vdots \\ Z_{q1} & \cdots & Z_{qn} \end{bmatrix}, \quad F(z) = \begin{bmatrix} F_1(z) \\ \vdots \\ F_n(z) \end{bmatrix}$$

with  $F_k(z) = \frac{\sqrt{1-a^2}}{z-a} \left( \frac{1-az}{z-a} \right)^{k-1}$ . To simplify the notation we will introduce a new variable via the relation  $\lambda^{-1} = \frac{1-az}{z-a}$ . In Appendix A we show that we are able to write the Popov function for the IQC constraint as

$$\mathcal{F}(z) = \begin{bmatrix} \Psi(\lambda) \tilde{D}(\lambda)^{-1} \\ I \end{bmatrix} M \begin{bmatrix} \Psi(\lambda) \tilde{D}(\lambda)^{-1} \\ I \end{bmatrix} \quad (38)$$

where now

$$M = [C_0 T \quad D_0]^T \begin{bmatrix} -Q & -R \\ -R^T & Q \end{bmatrix} [C_0 T \quad D_0] \quad (39)$$

with suitable definitions of  $C_0$ ,  $D_0$ ,  $T$ , and where  $\tilde{D}(\lambda) = \lambda^{n+n_\phi} I_m$  with  $m = \sum_{i=1}^l m_i$ . Here  $\Psi(\lambda)$  is defined similarly as  $\Psi(z)$  in Section 3.1. Notice that this is in the form of an FIR Popov function in the variable  $\lambda$ . The constraints  $X_i \succeq 0$  can be expressed in a similar way. Hence we obtain several Popov function constraints coupled via the variables  $Q$  and  $R$ .

We now investigate some numerical examples, where we have considered four different methods for solving the problem, which in all cases have been formulated in the  $\lambda$ -domain. The first one is the sampling method based on a uniform sampling on the unit circle. The other three methods are based on the KYP lemma formulation. For the first one of these the problem is just solved in a straightforward way using the solver SDPT3. The second one is using the preprocessing of STRUL in conjunction with SDPT3, and the third one is using the software KYPD, [WHJ09], in conjunction with SDPT3.

In the first experiment we generate random Laguerre models of degrees  $n$  in the range 2 to 16 in steps of 2. This is done by taking all entries of  $Z \in \mathbf{R}^{m \times n}$  from a standard normal distribution normalized with a factor of  $10^{-5}$  so that all systems generated are robustly stable. The value of  $m$  has been fixed to 10 and the number of uncertainties  $l$  has been 4, with dimensions  $m_i$  equal to 1, 2, 3, and 4, respectively. The value of  $n_\phi$  has been fixed to 3. Hence the total number of states  $m(n + n_\phi)$  for the largest Popov function constraint has been in the range of 50 to 190 with steps of 20. Notice that there are 4 additional Popov function constraints with  $n_\phi = 3$  states each in order to impose the constraints  $X_i \succeq 0$ .

We report computational results in tables 3 and 4. For the two methods based on solving the problem using the plain KYP lemma formulation and in conjunction with STRUL the solver runs out of memory for values of  $n$  greater or equal to 6. However, these methods are the fastest ones for small problems. When the KYPD solver is used it runs out of memory for values of  $n$  greater than or equal to 8. For the sampling method the solver runs out of memory for values of  $n$  greater than 16. For the value of  $n = 16$  there are for the sampling method 2065 equality constraints, 5 SDP blocks of dimension 240 in total, and there are 480 free variables. For the case of  $n = 4$ , which is the largest value of  $n$  for which all methods produce solutions, we can compare the dimensions of the problems. All of them have 5 SDP blocks of dimension 120. However, SDPT3 and STRUL have 3115 equality constraints, whereas the sampling method and KYPD only have 865 constraints. The number of free variables are 480 for the sampling method, none for SDPT3, 262 for STRUL and 210 for KYPD. The sampling method is always faster than the KYPD method irrespective of the

$m(n + n_\phi)$	SAMPLING		SDPT3		STRUL		KYPD	
	tot	sol	tot	sol	tot	sol	tot	sol
50	30.9	24.6	7.24	3.47	17.5	9.25	41.3	24.2
70	57.5	48.4	21.0	11.5	48.3	30.2	85.2	45.8
90	89.6	75.7	–	–	–	–	164	76.7
110	142	127	–	–	–	–	–	–
130	226	209	–	–	–	–	–	–
150	321	298	–	–	–	–	–	–
170	425	386	–	–	–	–	–	–
190	563	517	–	–	–	–	–	–

Table 3: Results with four methods for the example of Section 4.2. Total time in seconds including YALMIP time, solver time, and setup time, and separately solver time.

$m(n + n_\phi)$	SAMPLING		SDPT3		STRUL		KYPD	
	per iter	iter						
50	1.29	19	0.50	7	1.32	7	1.15	21
70	2.42	20	1.43	8	3.77	8	2.08	22
90	3.98	19	–	–	–	–	3.34	23
110	6.32	20	–	–	–	–	–	–
130	10.4	20	–	–	–	–	–	–
150	14.9	20	–	–	–	–	–	–
170	19.3	20	–	–	–	–	–	–
190	25.8	20	–	–	–	–	–	–

Table 4: Time per iteration and number of iterations for the example in Section 4.2.

$m(n + n_\phi)$	SAMPLING		SDPT3		STRUL		KYPD	
	tot	sol	tot	sol	tot	sol	tot	sol
20	1.34	0.47	1.48	0.52	1.47	0.44	3.52	0.68
40	3.46	1.24	3.62	1.74	4.21	1.94	4.21	1.44
60	6.96	2.37	9.94	6.35	16.8	9.90	9.02	2.85
80	14.0	4.93	28.9	18.9	48.1	33.5	19.4	6.47
100	27.4	10.1	–	–	–	–	42.9	14.2
120	48.9	18.8	–	–	–	–	79.3	28.2
140	80.4	30.8	–	–	–	–	122	46.0
160	134	53.0	–	–	–	–	193	79.8
180	202	77.7	–	–	–	–	316	130
200	–	–	–	–	–	–	467	195

Table 5: Total time in seconds including YALMIP time, solver time and setup time, and separately solver time for the second experiment in the example in Section 4.2.

values of  $n$ . It is seen that the computational time per iteration for the sampling method grows very closely to linearly with  $m(n + n_\phi)$  for values above 100.

In the second experiment we generate random Laguerre models of degrees  $n$  in the range of 9 to 99 with steps of 10. This is done by taking all entries of  $Z \in \mathbf{R}^{m \times n}$  from a standard normal distribution normalized with a factor of  $10^{-5}$  so that all systems generated are robustly stable. The value of  $m$  has been fixed to 2 and the number of uncertainties  $\delta_i$  have been 2, with dimensions 1. The value of  $n_\phi$  has been fixed to 1. Hence the total number of states has been in the range of 20 to 200 with steps of 20 for the largest Popov function constraint.

We report computational results in tables 5 and 6. When no data are reported for STRUL or SDPT3 it is because the solvers run out of memory. When no data is reported for SAMPLING it is because YALMIP runs out of memory when performing dualize. For larger problem sizes than the ones considered KYPD also fails due to the solver running out of memory. For the case of  $n = 39$ , which is the largest value of  $n$  for which all methods produce solutions, we can compare the dimensions of the problems. All of them have 3 SDP blocks of dimension 86. However, SDPT3 and STRUL have 3250 equality constraints, whereas the sampling method and KYPD only have 167 equality constraints. The number of free variables is 8 for the sampling method, 6 for KYPD, and none for SDPT3 and STRUL. For the case of  $n = 89$ , which is the largest value of  $n$  for which both the sampling method and KYPD produce solutions, both methods have 3 SDP blocks of of dimension 186 and 367 equality constraints. The number of free variables is 8 for the sampling method and 6 for KYPD. It is seen that the sampling method is always the fastest method. The time per iteration for this method grows as  $m(n + n_\phi)$  to the power of 3.55 for values of  $m(n + n_\phi)$  above 100. For KYPD the corresponding value is 3.97, which is very close to the theoretical value of 4.

## 5 Conclusions

In this paper we have shown how to reformulate an infinite-dimensional frequency domain Popov function constraint as a finite SDP constraint with low-rank coefficient matrices. The low-rank structure can be exploited to improve the efficiency of interior-point SDP solvers. The key obser-

$m(n + n_\phi)$	SAMPLING		SDPT3		STRUL		KYPD	
	per iter	iter						
20	0.03	17	0.07	7	0.06	7	0.04	19
40	0.07	19	0.25	7	0.28	7	0.06	23
60	0.12	20	0.79	8	1.24	8	0.12	24
80	0.26	19	2.36	8	4.18	8	0.27	24
100	0.51	20	–	–	–	–	0.55	26
120	0.94	20	–	–	–	–	1.01	28
140	1.62	19	–	–	–	–	1.71	27
160	2.65	20	–	–	–	–	2.75	29
180	3.88	20	–	–	–	–	4.32	30
200	–	–	–	–	–	–	6.71	29

Table 6: Time per iteration and number of iterations for the second experiment in the example in Section 4.2.

vation is that rational Popov function constraints can be expressed as polynomial Popov function constraints using a right MFD description of the associated transfer function, and that sampling this polynomial constraint results in an equivalent set of constraints with low-rank structure. This formulation is especially attractive when the transfer functions are parameterized using orthogonal basis functions. The sampling formulation also requires less memory than other methods, such as the ones based on the KYP lemma reformulation. The potential of the method has not been fully utilized. It is possible to speed up the computations and reduce memory requirements by making use of the fact that the sampling constraints are of low-rank and in many cases also sparse. Another important topic for future research is to investigate sampling strategies for general multi-input transfer functions represented in a non-orthogonal basis.

## A Polynomial IQC Popov function

In this appendix we give the details of the expressions (38) and (39) for the Popov function in the IQC example of Section 4.2.

With the definitions made in Section 4.2 it holds that  $G(z) = W_0(\lambda)\tilde{G}(\lambda) = \tilde{F}_0 + \tilde{N}_G(\lambda)\tilde{D}_G(\lambda)^{-1}$ , where

$$W_0(\lambda) = \frac{1 + a\lambda}{\sqrt{1 - a^2}}, \quad \tilde{N}_G(\lambda) = \tilde{Z} \begin{bmatrix} \lambda^{n-1} I_m \\ \vdots \\ \lambda I_m \\ I_m \end{bmatrix}, \quad \tilde{D}_G(\lambda) = \lambda^n I_m$$

and

$$\tilde{Z} = \frac{1}{\sqrt{1 - a^2}} [Z_1 + aZ_2 \quad Z_2 + aZ_3 \quad \cdots \quad Z_{n-1} + aZ_n \quad Z_n], \quad \tilde{F}_0 = \frac{a}{\sqrt{1 - a^2}} Z_1.$$

With

$$\bar{D}_i = \begin{bmatrix} I_{m_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \tilde{N}_{\phi_i}(\lambda) = \begin{bmatrix} 0 \\ \lambda^{n_\phi-1} I_{m_i} \\ \vdots \\ \lambda I_{m_i} \\ I_{m_i} \end{bmatrix}, \quad \tilde{D}_{\phi_i}(\lambda) = \lambda^{n_\phi} I_{m_i}$$

it holds that  $\phi_i(z) = \tilde{\phi}_i(\lambda) = \bar{D}_i + \tilde{N}_{\phi_i}(\lambda)\tilde{D}_{\phi_i}(\lambda)^{-1}$ . Here  $(\tilde{N}_{\phi_i}(z), \tilde{D}_{\phi_i}(z))$  is a strictly proper right co-prime and column-reduced MFD. With

$$\bar{D} = \mathbf{diag}(\bar{D}_1, \dots, \bar{D}_m), \quad \tilde{N}_\phi = \mathbf{diag}(\tilde{N}_{\phi_1}, \dots, \tilde{N}_{\phi_m}), \quad \tilde{D}_\phi = \mathbf{diag}(\tilde{D}_{\phi_1}, \dots, \tilde{D}_{\phi_m})$$

we have  $\Phi(z) = \tilde{\Phi}(\lambda) = \bar{D} + \tilde{N}_\phi(\lambda)\tilde{D}_\phi(\lambda)^{-1}$ , where  $(\tilde{N}_\phi, \tilde{D}_\phi)$  is strictly proper, right co-prime and column-reduced. Let  $X$  be a matrix such that  $\tilde{N}_\phi \tilde{F}_0 = X \tilde{N}_\phi$ . Then with

$$D_0 = \begin{bmatrix} \bar{D} \tilde{F}_0 \\ \bar{D} \end{bmatrix}, \quad C_0 = \begin{bmatrix} \bar{D} & I & X \\ 0 & 0 & I \end{bmatrix}, \quad \tilde{N} = \begin{bmatrix} \tilde{N}_G \tilde{D}_\phi \\ \tilde{N}_\phi \tilde{N}_G \\ \tilde{N}_\phi \tilde{D}_G \end{bmatrix}, \quad \tilde{D} = \tilde{D}_\phi \tilde{D}_G$$

it holds that

$$\begin{bmatrix} \Phi(z)G(z) \\ \Phi(z) \end{bmatrix} = D_0 + C_0 \tilde{N}(\lambda) \tilde{D}(\lambda)^{-1}.$$

It is straightforward to verify that  $(\tilde{N}, \tilde{D})$  is a strictly proper, right co-prime and column-reduced MFD. It only remains to write  $\tilde{N}(\lambda) = T\Psi(\lambda)$  to be able to write

$$\mathcal{F}(z) = \begin{bmatrix} \Psi(\lambda) \tilde{D}(\lambda)^{-1} \\ I \end{bmatrix}_* M \begin{bmatrix} \Psi(\lambda) \tilde{D}(\lambda)^{-1} \\ I \end{bmatrix}$$

where now

$$M = [C_0 T \quad D_0]^T \begin{bmatrix} -Q & -R \\ -R^T & Q \end{bmatrix} [C_0 T \quad D_0].$$

Notice that  $T$  might not be square, and that it is important to rewrite the function  $F(z)$  in terms of an MFD for which the number of rows of the MFD are equal to the sum of the column degrees. Otherwise the KYP lemma cannot be applied.

We will look at each of the block rows in  $\tilde{N}$  separately. The first one reads

$$\tilde{N}_G \tilde{N}_\phi = \tilde{Z} \begin{bmatrix} \lambda^{n+n_\phi-1} I_m \\ \vdots \\ \lambda^{n_\phi} I_m \end{bmatrix}.$$

The second block row reads

$$\tilde{N}_\phi \tilde{N}_G = \begin{bmatrix} H_1 \\ \vdots \\ H_q \end{bmatrix} \begin{bmatrix} \lambda^{n_\phi+n-2} I_m \\ \vdots \\ \lambda I_m \\ I_m \end{bmatrix}$$

where

$$H_i = \begin{bmatrix} 0 \\ \tilde{H}_i \end{bmatrix}$$

where  $\tilde{H}_i$  is a block Hankel matrix with first block row equal to  $[\tilde{Z}_{i1} \ \cdots \ \tilde{Z}_{in} \ 0 \ \cdots \ 0]$ . Finally, the third block row of  $\tilde{N}$  is given by

$$\tilde{N}_\phi \tilde{D}_G = \mathbf{diag}_i \begin{bmatrix} 0 \\ \lambda^{n+n_\phi-1} I_{m_i} \\ \vdots \\ \lambda^n I_{m_i} \end{bmatrix}.$$

From this it is straightforward to write down  $T$  by reordering columns and padding with zeros. We see that this results in a FIR Popov function in the variable  $\lambda$ .

The conditions  $X_i \succeq 0$  are Popov function inequalities. We may also rewrite these Popov functions in terms of MFDs in the variable  $\lambda$ . It is immediate that

$$X_i = \begin{bmatrix} \Psi_{Q_i}(\lambda) D_{Q_i}(\lambda)^{-1} \\ I \end{bmatrix}_* M_i \begin{bmatrix} \Psi_{Q_i}(\lambda) D_{Q_i}(\lambda)^{-1} \\ I \end{bmatrix}$$

where

$$M_i = [I \ D_{Q_i}^0]^T Q_i [I \ D_{Q_i}^0]$$

and

$$\Psi_{Q_i}(\lambda) = \begin{bmatrix} 0 \\ \lambda^{n_\phi-1} I_{m_i} \\ \vdots \\ \lambda I_{m_i} \\ I_{m_i} \end{bmatrix}, \quad D_{Q_i}^0 = \begin{bmatrix} I_{m_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad D_{Q_i}(\lambda) = \lambda^{n_\phi} I_{m_i}.$$

From this it is straightforward to write down  $T_i$  by reordering columns such that  $\Psi_{Q_i} = T_i \mathbf{diag}_j \begin{bmatrix} \lambda^{n_\phi-1} \\ \vdots \\ \lambda \\ 1 \end{bmatrix}$ .

## References

- [AV02] B. Alkire and L. Vandenberghe. Convex optimization problems involving finite auto-correlation sequences. *Mathematical Programming Series A*, 93:331–359, 2002.
- [Bal95] V. Balakrishnan. Linear matrix inequalities in robustness analysis with multipliers. *Syst. Control Letters*, 25(4):265–272, 1995.
- [BDG<sup>+</sup>93] G.J. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith.  *$\mu$ -Analysis and Synthesis Toolbox*, 1993.
- [BEFB94] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.
- [Bor99] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

- [BV87] Th. G. J. Beelen and G. W. Veltkamp. Numerical computation of a coprime factorization of a transfer function matrix. *Systems & Control Letters*, 9:281–288, 1987.
- [BV03] V. Balakrishnan and L. Vandenberghe. Semidefinite programming duality and linear time-invariant systems. *IEEE Transactions on Automatic Control*, 48:30–41, 2003.
- [BY05] S. J. Benson and Y. Ye. DSDP5 user guide — software for semidefinite programming. Technical Report ANL/MCS-TM-277, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, September 2005.
- [Dum07] B. Dumitrescu. *Positive Trigonometric Polynomials and Signal Processing Applications*. Springer, 2007.
- [FLH11] R. Falkeborn, J. Löfberg, and A. Hansson. Lowrank exploitation in semidefinite programming for control. *International Journal of Control*, 84(12):1975–1982, 2011.
- [GH03] J. Gillberg and A. Hansson. Polynomial complexity for a Nesterov-Todd potential-reduction method with inexact search directions. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 3, pages 3824–3829, 2003.
- [GHN<sup>+</sup>02] Y. Genin, Y. Hachez, Y. Nesterov, R. Stefan, P. Van Dooren, and S. Xu. Positivity and linear matrix inequalities. *European Journal of Control*, 8(3):275–298, 2002.
- [GHN<sup>V</sup>03] Y. Genin, Y. Hachez, Yu. Nesterov, and P. Van Dooren. Optimization problems over positive pseudopolynomial matrices. *SIAM Journal on Matrix Analysis and Applications*, 25(1):57–79, 2003.
- [Hac03] Y. Hachez. *Convex Optimization over Non-Negative Polynomials: Structured Algorithms and Applications*. PhD thesis, Université catholique de Louvain, Belgium, 2003.
- [HAPS01] D. Henrion, D. Arzelier, D. Peaucelle, and M. Sebek. An LMI condition for robust stability of polynomial matrix polytopes. *Automatica*, 37(3):461–468, Mar. 2001.
- [HdHW05] P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlberg. *Modeling and Identification with Rational Orthogonal Basis Functions*. Springer Verlag, London, 2005.
- [HSK99] B. Hassibi, A. H. Sayed, and T. Kailath. *Indefinite-Quadratic Estimation and Control. A Unified Approach to  $H^2$  and  $H^\infty$  Theories*. Society for Industrial and Applied Mathematics, 1999.
- [HSK03] D. Henrion, M. Sebek, and V. Kucera. Positive polynomials and robust stabilization with fixed-order controllers. *IEEE Transactions on Automatic Control*, 48(7):1178–1186, Jul 2003.
- [HV00] A. Hansson and L. Vandenberghe. Efficient solution of linear matrix inequalities for integral quadratic constraints. In *Proc. IEEE Conf. on Decision and Control*, pages 5033–5034, 2000.
- [HWH06] J. Harju, R. Wallin, and A. Hansson. Utilizing low rank properties when solving KYP-SDPs. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5150–5155, 2006.

- [IOW99] V. Ionescu, C. Oară, and M. Weiss. *Generalized Riccati Theory and Robust Control—A Popov Function Approach*. John Wiley & Sons, Chichester, 1999.
- [Jön96] U. Jönsson. *Robustness Analysis of Uncertain and Nonlinear Systems*. PhD thesis, Lund Institute of Technology, Sweden, 1996.
- [Kai80] T. Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, N.J., 1980.
- [Kao12] C.-Y. Kao. On stability of discrete-time LTI systems with varying time delays. *IEEE Transactions on Automatic Control*, 57(5):1243–1248, 2012.
- [KM01] C.-Y. Kao and A. Megretski. Fast algorithms for solving IQC feasibility and optimization problems. In *Proceedings of the American Control Conference*, volume 4, pages 3019–3024, 2001.
- [KM03] C.-Y. Kao and A. Megretski. A new barrier function for IQC optimization problems. In *Proceedings of the American Control Conference*, volume 5, pages 4281–4286, 2003.
- [KM07] C.-Y. Kao and A. Megretski. On the new barrier function and specialized algorithms for a class of semidefinite programs. *SIAM Journal on Control and Optimization*, 46(2):468–495, 2007.
- [KMJ00] C.-Y. Kao, A. Megretski, and U. Jönsson. An algorithm for solving optimization problems involving special frequency dependent LMIs. In *Proceedings of the American Control Conference*, volume 1, pages 307–311, 2000.
- [KMJ03] C.-Y. Kao, A. Megretski, and U. Jönsson. Specialized fast algorithms for IQC feasibility and optimization problems. *Automatica*, 40(2):239–252, 2003.
- [Löf04] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [Löf09] J. Löfberg. Dualize it: software for automatic primal and dual conversions of conic programs. *Optimization Methods and Software*, 24:313 – 325, 2009.
- [LP04] J. Löfberg and P. A. Parrilo. From coefficients to samples: a new approach to SOS optimization. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 3154–3159, 2004.
- [LV07] Z. Liu and L. Vandenberghe. Low-rank structure in semidefinite programs derived from the KYP lemma. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5652–5659, 2007.
- [MKJR98] A. Megretski, C.-Y. Kao, U. Jönsson, and A. Rantzer. *A Guide To IQC $\beta$ : Software for Robustness Analysis*. Massachusetts Institute of Technology, 1998.
- [MR97a] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Trans. Aut. Control*, 42(6):819–830, June 1997.
- [MR97b] A. Megretski and A. Rantzer. Systems analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, 1997.

- [Nes00] Y. Nesterov. Squared functional systems and optimization problems. In J. Frenk, C. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization Techniques*, pages 405–440. Kluwer Academic Publishers, 2000.
- [OMDS97] P. Van Overschee, B. De Moor, W. Dehandschutter, and J. Swevers. A subspace algorithm for the identification of discrete time frequency domain power spectra. *Automatica*, 33(12):2147–2157, 1997.
- [Pag95] F. Paganini. Necessary and sufficient conditions for robust  $H_2$  performance. In *Proc. IEEE Conf. on Decision and Control*, pages 1970–1975, New Orleans, LA, December 1995.
- [Par99] P. A. Parrilo. On the numerical solution of LMIs derived from the KYP lemma. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 3, pages 2334–2338, 1999.
- [Pat81] R. V. Patel. Computation of matrix fraction descriptions of linear time-invariant systems. *IEEE Transactions on Automatic Control*, 26(1):148–161, 1981.
- [RDV07] T. Roh, B. Dumitrescu, and L. Vandenberghe. Multidimensional FIR filter design via trigonometric sum-of-squares optimization. *IEEE Journal of Selected Topics in Signal Processing*, 1:641–650, 2007.
- [RP02] D.C.W. Ramos and P.L.D. Peres. An LMI condition for the robust stability of uncertain continuous-time linear systems. *IEEE Transactions on Automatic Control*, 47(4):675–678, Apr. 2002.
- [RV06] T. Roh and L. Vandenberghe. Discrete transforms, semidefinite programming, and sum-of-squares representations of nonnegative polynomials. *SIAM Journal on Optimization*, 16(4):939–964, 2006.
- [Stu99] J. F. Sturm. Using SEDUMI 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.
- [TTT03] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming Series B*, 95:189–217, 2003.
- [TTT10] K.-C. Toh, M. J. Todd, and R. H. Tütüncü. On the implementation and usage of SDPT3—a MATLAB software package for semidefinite-quadratic-linear programming, version 4.0. Technical report, Department of Mathematics, National University of Singapore, 2010. draft.
- [VBW<sup>+</sup>05] L. Vandenberghe, V. R. Balakrishnan, R. Wallin, A. H. Hanson, and T. Roh. Interior point algorithms for semidefinite programming problems derived from the KYP lemma. In D. Henrion and A. Garulli, editors, *Positive Polynomials in Control*, volume 312 of *Lecture Notes on Control and Information Sciences*. Springer Verlag, 2005.
- [WBV98] S.-P. Wu, S. Boyd, and L. Vandenberghe. FIR filter design via spectral factorization and convex optimization. In B. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, pages 215–245. Birkhauser, 1998.

- [WHJ09] R. Wallin, A. Hansson, and J. H. Johansson. A structure exploiting preprocessor for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma. *IEEE Transactions on Automatic Control*, 54(4):697–704, 2009.
- [WKH08] R. Wallin, C.-Y. Kao, and A. Hansson. A cutting plane method for solving KYP-SDPs. *Automatica*, 44(2):418–429, 2008.
- [YFK03] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of SDPA 6.0 (Semidefinite Programming Algorithm 6.0). *Optimization Methods and Software*, 18(4):491–505, 2003.