Nu-Gap Metric A Sum-Of-Squares and Linear Matrix Inequality Approach

S. Taamallah
Executive summary

Nu-Gap Metric A Sum-Of-Squares and Linear Matrix Inequality Approach

Problem area
The nu-gap metric represents a good measure of the distance between systems in a closed-loop setting. The purpose of this paper is to present a novel method to compute the nu-gap using a Semi-Definite Programming (SDP) procedure.

Description of work
Our approach is formulated through a three-step modus operandi: (i) first an initial central transfer function is computed through Linear Matrix Inequality (LMI) relaxations of a non-convex problem, on the basis of matrix Sum-Of-Squares (SOS) decompositions, followed by (ii) a non-linear LMI-based refinement, and finally (iii) the actual computation of the nu-gap using the Kalman-Yakubovich-Popov (KYP) Lemma. We illustrate the practicality of the proposed method on numerical examples.

Results and conclusions
The main benefits of our approach are as follows. First, it is well known that the nu-gap metric does not account for any performance objectives of a closed-loop system. Hence, if the application at hand includes also robust performance specifications, and since our optimization problem is SDP based, one could easily add additional performance constraints to the nu-gap by ensuring that the appropriate sensitivity functions are all well behaved. Second, our approach could potentially be combined with other LMI based applications in the nu-gap metric, such as the recent and powerful results related to model order reduction.

Applicability
System modeling and design of (flight) control systems.
Nu-Gap Metric A Sum-Of-Squares and Linear Matrix Inequality Approach

Nationaal Lucht- en Ruimtevaartlaboratorium, National Aerospace Laboratory NLR

Anthony Fokkerweg 2, 1059 CM Amsterdam,
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
Telephone +31 88 511 31 13, Fax +31 88 511 32 10, Web site: www.nlr.nl
Nu-Gap Metric A Sum-Of-Squares and Linear Matrix Inequality Approach

S. Taamallah

This report is based on a presentation held at the International Conference on System Theory, Control and Computing (IEEE co-sponsor), October 17 – 19, 2014, Sinaia, Romania.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.
Nu-Gap Metric
A Sum-Of-Squares and Linear Matrix Inequality Approach

Skander Taamallah
National Aerospace Laboratory (NLR)
Anthony Fokkerweg 2, 1059 CM, Amsterdam, The Netherlands
email: staamall@nlr.nl

Abstract—The nu-gap metric represents a good measure of the distance between systems in a closed-loop setting. The purpose of this paper is to present a novel method to compute the nu-gap using a Semi-Definite Programming (SDP) procedure. Our approach is formulated through a three-step modus operandi: (i) first an initial central transfer function is computed through Linear Matrix Inequality (LMI) relaxations of a nonconvex problem, on the basis of matrix Sum-Of-Squares (SOS) decompositions, followed by (ii) a non-linear LMI-based refinement, and finally (iii) the actual computation of the nu-gap using the Kalman-Yakubovich-Popov (KYP) Lemma. We illustrate the practicality of the proposed method on numerical examples.

I. INTRODUCTION

Gap and graph metrics [1] have been known to provide a measure of the separation between open-loop systems, in terms of their closed-loop behavior. The first attempt to introduce such a metric, simply known as gap metric, was formulated in [2], [3], whereas an efficient method for computing it was presented in [4], with recent works from a fairly general perspective proposed in [5]. Other significant metrics have also been investigated, such as (i) the T-gap metric [6], (ii) the pointwise gap [7], and (iii) Vinnicombe’s popular nu-gap (i.e. v-gap) metric [8], [9]. Similar to its predecessor gap metrics, the v-gap provides also a means of quantifying feedback system stability and robustness, while being concurrently less conservative and simpler to compute. Time-varying and nonlinear extensions to both the gap metric [10], [11], [12], [13] and the v-gap metric [14], [15], [16] have also been researched, although analytical computations of these metrics, in this nonlinear setting, is generally difficult. Over the years the use of these metrics has received much attention. In particular, the v-gap was extensively studied in the realm of system identification metrics has received much attention. In particular, the setting, is generally difficult. Over the years the use of these analytical computations of these metrics, in this nonlinear setting, is generally difficult. Over the years the use of these metrics has received much attention. In particular, the v-gap was extensively studied in the realm of system identification metrics has received much attention. In particular, the setting, is generally difficult. Over the years the use of these analytical computations of these metrics, in this nonlinear setting, is generally difficult. Over the years the use of these metrics has received much attention. In particular, the v-gap was extensively studied in the realm of system identification metrics has received much attention. In particular, the setting, is generally difficult. Over the years the use of these analytical computations of these metrics, in this nonlinear setting, is generally difficult. Over the years the use of these metrics has received much attention. In particular, the v-gap was extensively studied in the realm of system identification metrics has received much attention. In particular, the setting, is generally difficult. Over the years the use of these analytical computations of these metrics, in this nonlinear setting, is generally difficult. Over the years the use of these metrics has received much attention. In particular, the v-gap was extensively studied in the realm of system identification.
Functions (TFs) in \( \mathcal{L}_\infty \) (resp. \( \mathcal{H}_\infty \)). Finally 1 and 0 will be used to denote the identity and null matrices respectively, assuming appropriate sizes.

II. PRELIMINARIES

This section introduces first the KYP Lemma [31] (see also [32] for a proof) and the Bounded Real Lemma (BRL) [33].

Lemma 1: Let complex matrices \( A, B \), and a symmetric matrix \( \Theta \), of appropriate sizes, be given. Suppose \( \lambda(A) \subset \mathbb{C}^- \cup \mathbb{C}^+ \), then the following two statements are equivalent.

(i) \[
\begin{bmatrix}
(j\omega - A)^{-1}B \\
I
\end{bmatrix} \Theta \begin{bmatrix}
(j\omega - A)^{-1}B \\
I
\end{bmatrix} < 0
\] (1)

(ii) There exists a matrix \( P = P^* \), for which the following linear matrix map \( L(P) \) holds.

\[
L(P) + \Theta < 0, \\
L(P) = \begin{bmatrix}
A & B \\
I & 0
\end{bmatrix} \begin{bmatrix}
P & 0 \\
0 & P
\end{bmatrix} \begin{bmatrix}
A & B \\
I & 0
\end{bmatrix} < 0
\] (2)

Proof: See [32].

Remark 1: We have dealt here with the strict version of the KYP lemma, i.e. strict inequalities, since no controllability/stabilizability assumptions become necessary, and with use of interior-point based solvers, existence of strictly feasible solutions will be guaranteed.

Remark 2: If matrices \( A, B, \) and \( \Theta \) are all real, the equivalence still holds when restricting \( P \) to be real [34].

Lemma 2: Let a transfer function \( G(s) := \frac{A}{C} \) be given, then the following statements are equivalent.

(i) \( \forall \gamma > 0, \lambda(A) \subset \mathbb{C}^- \cup \mathbb{C}^+ \), \( ||G||^2 < \gamma^2 \) (3)

(ii) \( \exists P | P = P^* \) and \( L(P) + \Theta < 0 \)

\[
L(P) = \begin{bmatrix}
A & B \\
I & 0
\end{bmatrix} \begin{bmatrix}
P & 0 \\
0 & P
\end{bmatrix} \begin{bmatrix}
A & B \\
I & 0
\end{bmatrix}
\]

\[
\Theta = \begin{bmatrix}
C & D \\
D & 0
\end{bmatrix} \begin{bmatrix}
P & 0 \\
0 & P
\end{bmatrix} \begin{bmatrix}
C & D \\
D & 0
\end{bmatrix}
\] (4)

Proof: Using Lemma 1. This result is also known as the BRL in LMI form [35]. Note that for the case where \( \lambda(A) \subset \mathbb{C}^- \) we need to add the stability constraint \( P > 0 \) in (4), and for the case where \( \lambda(A) \subset \mathbb{C}^+ \) it is standard practice to perturb \( A \) by \( -\epsilon I \), with \( 0 < \epsilon < \infty \).

Next we briefly recall here some SOS related results. A SOS is a convex, finite dimensional, optimization problem, with a linear cost and SOS constraints [36], [37]. A scalar multivariate polynomial \( p(x) \), with \( x \in \mathbb{R}^n \), may be written as \( p(x) := \sum_{i=1}^{N} c_i x_{\alpha_i} \), with \( c_i \in \mathbb{R} \) the coefficients, and \( \alpha_{i,j} \in \mathbb{N}_0 \) := \( \{0, \ldots, N\} \) the exponents. The set of all polynomials in the variable \( x \) is denoted by \( \mathbb{R}[x] \). Polynomials are often expressed in terms of a finite linear combination of their monomials (products of powers of \( x_j \), these latter denoted by \( k(x) = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n} \)). For \( a \in \mathbb{R}^n \) it is often standard practice to instead use the following notation \( k(x) = x^a \). Recent extensions to so-called matrix SOS decompositions have been derived and are restated hereafter.

Definition 1: A symmetric matrix-valued polynomial matrix \( S(x) : \mathbb{R}^n \rightarrow \mathcal{S}^r \) is said to be SOS, if there exists a (not necessarily square and typically tall) polynomial matrix \( T(x) \) such that \( S(x) = T(x)^T T(x) \) [38], [39], [40].

III. PROBLEM STATEMENT

There exists several equivalent definitions of the \( \nu \)-gap metric. The one chosen in this paper is most convenient for our purpose. The \( \nu \)-gap between two plants having transfer functions \( P_1 \) and \( P_2 \), with \( P_1, P_2 \in \mathcal{F} \), is defined as [8]

\[
\delta_{\nu}(P_1, P_2) := \begin{cases} 
\delta_{L_2}(P_1, P_2) & \text{if the WNC holds} \\
1 & \text{else}
\end{cases}
\] (5)

with

\[
\delta_{L_2}(P_1, P_2) := \|[I + P_2 P_1^{-1}]^{-1/2}(P_2 - P_1)(I + P_1 P_2^{-1})^{-1/2}\|_\infty
\] (6)

and WNC the so-called Winding Number Condition associated with the Nyquist diagram, for which an efficient computational method already exists [8], [9]. This paper focuses on the \( \delta_{L_2}(P_1, P_2) \) part of the \( \nu \)-gap metric, also expressed as

\[
\sup_{\omega \in \mathcal{R}} \left|\lambda\left([I + P_1 P_2^{-1}]^{-1/2}(P_2 - P_1)^*(I + P_1 P_2^{-1})^{-1/2}(P_2 - P_1)\right)\right|^{1/2}
\] (7)

Our goal consists now in computing (7) using a SDP approach, albeit through a sub-optimal approach (the sub-optimality aspects of our approach will further be discussed in Section V). Now (7) can easily be recast into LMI form, as the minimization of a maximum eigenvalue problem [27] (or \( L_2 \)-induced gain of a static operator).

Problem 1 Let transfer functions \( P_1, P_2 \in \mathcal{F} \) be given, then the solution to (7) is also given by

\[
\begin{align*}
\delta_{L_2}^2(P_1, P_2) := \inf_{\omega \in \mathcal{R}, \lambda \in \mathbb{R}^+} \lambda & \text{ subject to } \\
(P_2 - P_1)^*(I + P_2 P_1^{-1})^{-1/2}(P_2 - P_1) & < \lambda(I + P_1 P_2^{-1})
\end{align*}
\] (8)

From Problem 1 we obtain the following result.

Lemma 3: Let \( P_1, P_2 \) be given. Let a so-called central transfer function \( T \) be such that \( T^* T := I + P_2 P_2^{-1} \), and a transfer function \( L \) be such that \( L := T^{-1} \), with \( P_1, P_2, T, L \in \mathcal{F} \), then the solution \( \delta_{L_2}(P_1, P_2) \) to Problem 1 is given by

\[
\begin{align*}
\delta_{L_2}^2(P_1, P_2) := \inf_{\omega \in \mathcal{R}, \lambda \in \mathbb{R}^+} \lambda & \text{ subject to } \\
L(P_2 - P_1) & > \lambda, P_1
\end{align*}
\] (9)

(8)
Proof: Define $P_2 := \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}$ and $T := \begin{bmatrix} A_T & B_T \\ C_T & D_T \end{bmatrix}$.

Since by definition $I + P_2P_2^* > 0$, $\exists \mathbf{T}$ such that $T^T = I + P_2P_2^*$. Again since $D_T = I + D_2D_2^* > 0$, we infer $D_T^{-1}$ to be well-defined, thus we can find a realization for $L$, such that $L = T^{-1}$ [41]. Now by rewriting Problem 1 as partitioned matrices we get (8).

IV. APPLICATION OF THE KYP LEMMA

Now we express (8) in a form which is amenable to the KYP paradigm.

Lemma 4: Let $P_1 := \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}$, $P_2 := \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix}$ be given. Let the so-called central transfer function be defined as $T := \begin{bmatrix} A_T & B_T \\ C_T & D_T \end{bmatrix}$ such that $T^T = I + P_2P_2^*$, and define the following additional transfer functions as $L := \begin{bmatrix} A_L & B_L \\ C_L & D_L \end{bmatrix}$ such that $L := T^{-1}$.

$$R := \begin{bmatrix} A_R & B_R \\ C_R & D_R \end{bmatrix} \text{ such that } R := L(P_2 - P_1) = \begin{bmatrix} A_L & B_L \\ C_L & D_L \end{bmatrix} \begin{bmatrix} A_R & B_R \\ C_R & D_R \end{bmatrix}, \quad \text{and} \quad S := \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix}$$

with $P_1, P_2, T, L, R, S \in \mathcal{R}$, then the solution $\delta_L(P_1, P_2)$ to Problem 1 is given by

$$\delta_L^2(P_1, P_2) := \inf_{\omega \in \mathbb{R}, \lambda \in \mathbb{R}^+} \lambda \text{ subject to } ...$$

$$\Theta := \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -\lambda I \end{bmatrix} \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix}$$

Proof: The proof is straightforward. From (8), expand $P_1$ as $C_1(sI - A_1)^{-1}B_1 + D_1$, and similarly for $P_2$ and $L$, and regroup terms.

Now we give the following main result.

Lemma 5: Let transfer functions $P_1, P_2 \in \mathcal{R}$ be given. Let transfer function $S$ be defined as in Lemma 4, then the following three statements are equivalent:

(i) $\delta_L^2(P_1, P_2) := \|I + P_2P_2^*\|^{1/2}(P_2 - P_1)(I + P_1P_1^*)^{-1/2}\|_{\infty}^2$

(ii) $\delta_L^2(P_1, P_2) := \inf_{\omega \in \mathbb{R}, \lambda \in \mathbb{R}^+} \lambda$ subject to ...

(iii) $\delta_L^2(P_1, P_2) := \inf_{\omega \in \mathbb{R}, \lambda \in \mathbb{R}^+} \lambda$ subject to ...

$\Theta := \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -\lambda I \end{bmatrix} \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix}$

$\Phi := \begin{bmatrix} A_\Phi & B_\Phi \\ C_\Phi & D_\Phi \end{bmatrix}$ such that $H := (I + P_2P_2^*) - T^T = \begin{bmatrix} -A_2 & -C_2C_1 \\ 0 & A_2 \end{bmatrix}$

$\Theta := \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -\lambda I \end{bmatrix} \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix}$

Proof: From (9), it is a straightforward application of Lemma 1 and Lemma 2.

Note that (iii) in Lemma 5 gives the exact solution, in LMI form, to Problem 1 and hence to (7). So far we have assumed that the central transfer function $T$, such that $T^T := I + P_2P_2^*$, could exactly be computed. To the best of our knowledge, no method is known that could deliver this exact $T$, in the general case. Hence, we present in the sequel a method which provides an approximation to $T$. It is this approximation in $T$ that renders our computation of the $\nu$-gap metric sub-optimal.

V. COMPUTING THE CENTRAL TRANSFER FUNCTION $T$

Our goal is to find the central transfer function $T \in \mathcal{R}$ such that $T^T := I + P_2P_2^*$, with the ensuing realizations of $T$ and $P_2$ as defined in Lemma 3. Here too we will solve our problem in the $L_\infty$ norm paradigm.

Problem 2 Let $P_2 := \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} \in \mathcal{R}$ be given, with $A_2 \in \mathbb{R}^{n \times m}$, $B_2 \in \mathbb{R}^{n \times k}$, $C_2 \in \mathbb{R}^{r \times m}$, and $D_2 \in \mathbb{R}^{r \times k}$, find the optimal $T \in \mathcal{R}$ such that

$$\hat{T} := \arg \inf_{\Theta \in \mathcal{R}} \| (I + P_2P_2^*) - T^T \|_{\infty}$$

Next, we present a solution to Problem 2.

Lemma 6: Let $P_2 := \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} \in \mathcal{R}$ be given, and let transfer function $T$ be defined as $T := \begin{bmatrix} A_T & B_T \\ C_T & D_T \end{bmatrix}$, and transfer function $H$ be defined as $H := \begin{bmatrix} A_H & B_H \\ C_H & D_H \end{bmatrix}$ such that $H := (I + P_2P_2^*) - T^T = \begin{bmatrix} -A_2 & -C_2C_1 \\ 0 & A_2 \end{bmatrix}$

$\Theta := \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -\lambda I \end{bmatrix} \begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix}$

$\Phi := \begin{bmatrix} A_\Phi & B_\Phi \\ C_\Phi & D_\Phi \end{bmatrix}$ such that $H := (I + P_2P_2^*) - T^T = \begin{bmatrix} -A_2 & -C_2C_1 \\ 0 & A_2 \end{bmatrix}$

then the solution $\hat{T}$ to Problem 2 is given by

$$\hat{T} := \arg \inf_{\Theta \in \mathcal{R}} \| f(x) - f(x) \|_{\infty}$$

subject to $f(x) > 0$, $G(x) < 0$.

$\Theta := \begin{bmatrix} A_\Theta & K \end{bmatrix}$, $K H B_H C_H^* - C_H^* D_H T_B$,

$G(x) := \begin{bmatrix} A_H^* & K^* A_H \\ B_H^* & -\gamma I \end{bmatrix}$

(11)
Proof: Direct application of the BRL in LMI form to Problem 2.

Here we clearly see that the problem defined by (11) is non-convex, on the one hand due to such products as \( C_T^2 C_T \), \( C_T^2 D_T \), and on the other, due to the various cross-products between the Lyapunov matrix \( K \) and the state-space matrices of \( T \). Hence to solve Problem 2, we aim at constructing meaningful convex relaxations. Solving Problem 2 will be done in two stages: (i) first find an accurate initial value for \( \hat{\theta} \), and then (ii) through an iterative, nonlinear, yet SDP approach, refine this earlier guess. We present next the first stage of this method.

A. LMI relaxations on the basis of matrix SOS decompositions

In this section, we compute an initial value for \( \hat{\theta} \) in (11). The method we use here is based upon SOS decompositions, since it is well known that a plethora of relevant non-convex optimization problems may have natural formulations (or relaxations) as SOS programs. From our presentation of SOS problems in Section II, and from Definition 1, we recognize that (11) may be viewed as a polynomial SDP, having a linear objective \( f(x) \), with coefficients \( c_i \) as symmetric matrices, and variable \( x \) defined as \( x := (y, K, A_T, B_T, C_T, D_T) \). Hence, we are here concerned with a matrix SOS decompositions [38], [39], [40]. To convert our non-convex problem in (11) to an amenable SOS representation, we will heavily rely on the work done in [39], although other approaches also exist [38], [42], [43], [44].

Remark 3: Since we have \( I + D_T^2 D_T > 0 \), we can always find \( D_T \) such that \( D_T^2 D_T = I + D_T^2 D_2 \) by cholesky factorization. Hence, in our case \( D_T \) is not a free variable, which simplifies (11), by reducing \( x \) to \( x := (y, K, A_T, B_T, C_T) \).

Lemma 7: Let \( x \in \mathbb{R}^n \), \( f : \mathbb{R}^n \to \mathbb{R} \), and \( G : \mathbb{R}^n \to \mathbb{R}^r \) be general scalar, and symmetric-matrix-valued polynomials. Consider now the following problem

\[
\text{minimize } f(x) \text{ subject to } G(x) \leq 0
\]

then an LMI relaxation of (12), and hence a sub-optimal solution, based on matrix SOS decompositions, is given by the following optimization problem

\[
\text{minimize } d \text{ subject to } \ldots
S(x) \geq 0 \text{ and } s_0(x) := f(x) + (S(x), G(x)) - d
\]

\[
S(x) \text{ and } s_0(x) \text{ are SOS}
\]

with \( S(x) \) Lagrange multiplier matrices which are polynomial functions of \( x \).

Proof: From polynomial Lagrange duality with SOS, see p. 61 [39].

Next we recall the following important results, from [39], that will prove very useful in finding an expression for \( S(x) \).

Proposition 1: If we choose pairwise different monomials \( u_j(x), j = 1 \cdots \zeta \), and collect them into the vector \( u(x) = (u_1(x))^T \cdots (u_\zeta(x))^T \), then \( S(x) \) is said to be SOS with respect to the monomial basis \( u(x) \), if there exist real matrices \( T_j, j = 1 \cdots \zeta \), with \( r \) columns such that \( S(x) = T(x)^T T(x) \) with \( T(x) = \sum_{j=1}^\zeta T_j u_j(x) \). Further if \( w_j(x), j = 1 \cdots \eta \) denote the pairwise different monomials that appear in \( u(x) u(x)^T \), then we can determine the unique symmetric matrices \( Z_j \) s.t.

\[
u(x) u(x)^T = \sum_{j=1}^\zeta Z_j w_j(x), \text{ [39]}
\]

Lemma 8: The polynomial matrix \( S(x) \) of dimension \( r \) is SOS with respect to the monomial basis \( u(x) \) if and only if there exists symmetric \( S \) with \( S(x) = \sum_{j=1}^\eta S_j w_j(x), \) and the following linear system has a solution \( W \)

\[
\langle W, I \rangle \leq Z_j, \quad j = 1 \cdots \eta \quad W \geq 0
\]

with the matrix-valued generalization of the inner product defined by \( \langle A, B \rangle_r = \text{Trace}(A^T B) \) such that

\[
\text{Trace}(C) = \begin{bmatrix}
\text{Trace}(C_{11}) & \cdots & \text{Trace}(C_{1r})
\vdots & \ddots & \vdots
\text{Trace}(C_{r1}) & \cdots & \text{Trace}(C_{rr})
\end{bmatrix}
\]

Proof: Using Proposition 1, see pp. 66-67 in [39].

We are now in a position to present the LMI relaxations procedure, from [39], on the basis of matrix SOS decompositions, which will allow us to compute a solution to (13), hence an approximative solution to (11), and thus a solution to Problem 2.

Proposition 2: Let \( S(x) \) and \( s_0(x) := f(x) + \langle S(x), G(x) \rangle \) be given as in Lemma 7, then

(i) Select the monomial vector \( v(x) \in \mathbb{R}^\zeta \), and some real coefficient matrix \( B \in \mathbb{R}^{n \times \zeta} \), such that \( G(x) = B(I \otimes v(x)) \)

(ii) Choose monomial vectors \( u(x) \) and \( w_0(x) \) of length \( \zeta \) and \( \zeta_0 \) to parameterize \( S(x) \) and \( s_0(x) \) respectively

(iii) Find the pairwise different monomials \( w_j(x) \) such that:

\[
\langle w_0(x), 1 \rangle = 0, \quad \langle u(x) u(x)^T, v(x) \rangle = \sum_{j=0}^\zeta p_j w_j(x), \quad \langle u(x) u(x)^T, v(x) \rangle = \sum_{j=0}^\zeta a_j w_j(x)
\]

(iv) Find a solution to the following LMI

\[
W_0 \geq 0, \quad W \geq 0 \quad \text{and} \quad a_j + \langle W_0, P_j \rangle + \langle W(B \otimes I) \rangle (I \otimes P_j) - \delta_j d = 0, \quad \delta_j = 0, \quad j = 0, 1, \ldots, \eta
\]

with \( \delta_0 = 1 \) and \( \delta_j = 0 \) for \( j \geq 1 \)

Proof: Using Proposition 1, Lemma 8, and the derivation on pp. 71-72 [39].

Finally \( u_0(x) \) ought to be chosen so that its first \( n+1 \) elements are such that

\[
(I_{n+1} 0) u_0(x) = (1 \quad x^T)^T
\]

then the optimal solution to (13) can be reconstructed from the kernel of \( W_0 \). Ideally this kernel should have dimension one, so that only a single vector is recovered, i.e. the optimal solution. If this is not the case, then the optimal solution can be found by solving a polynomial semi-definite feasibility
problem on the basis vectors of this kernel (see [39] for further details).

Remark 4: The relaxations in Lemma 7, as stated by Theorem 4.9 p. 72 in [39], are "guaranteed to converge to the optimal value, if we choose (...) all monomials up to a certain degree, and if we let the degree bound grow to infinity". Hence, the non-linear SDP refinement of Section V-B is, from a conceptual viewpoint, not required. However, dimensionality problems exist with SOS formulations, since the number of decision variables increases exponentially with the number of variables and the degree of the polynomials. Hence, for system's order higher than say three or four, a minimal length for \( u_0(x) \) could be chosen as \( u_0(x) := [1 \ y \ vec(K)^T \ vec(A_T)^T \ vec(B_T)^T \ vec(C_T)^T] \), with \( vec(\cdot) \) and \( svec(\cdot) \) the vectorization of a matrix, and a symmetric matrix, respectively.

B. SDP-based nonlinear refinement

In Section V-A we computed an initial value for \( \tilde{T} \) in (11). Now, to keep the SOS decomposition computationally tractable, we had to limit the number, and degree, of monomials used in Proposition 2. To compensate for this loss of accuracy, we introduce next a computationally fast non-linear SDP based refinement. First we recall a result, that will be very helpful to solve LMIs with quadratic equality constraints, related to the solution of the Continuous-Time Algebraic Riccati Equation ( CARE) [31] as presented in [45].

Lemma 9: Let matrix \( Q > 0 \) be given, then the solution \( \hat{X} \), of appropriate size, to the equation \( Q - X^T X = 0 \) is given by

\[
\hat{X} := \arg \min_X \quad \text{trace}(X)
\]

subject to

\[
[Q \quad X^T \quad I] \geq 0
\]

(17)

Proof: See [45].

We will use LMI (17) to linearize the quadratic constraint \( C_T^T C_T \) in (11), by replacing \( C_T^T C_T \) by \( Q \), and then replacing \( X \) in (17) by \( X^T C_T \) (assuming matrix \( C_T \) square or fat, which is always the case in practice), and finally by adding LMI (17) to LMI (11). Now we discuss the algorithm for our recursive procedure. Our proposed approach is basically a simple two-step iterative LMI search, with a bisection on \( \gamma \), in spirit reminiscent of \( \mu \) DK-iteration synthesis [41]. The algorithm is structured as follows

(i) Compute \( D_T \) by cholesky factorization, see Remark 3
(ii) Compute an initial value for \( A_T, B_T, C_T \) from Section V-A
(iii) In LMI (11), fix \( A_T, B_T, C_T \), minimize \( \gamma \) wrt \( K \)
   (a) Set \( \hat{\gamma} = \gamma \), \( \gamma = 0 \), and \( \gamma = \frac{\gamma + (\hat{\gamma} - \gamma)}{2} \)
   (b) In LMI (11) replace \( C_T^T C_T \) by \( Q > 0 \), and in LMI (17) replace \( X \) by \( C_T \), minimize \( -\text{trace}(C_T) \) wrt \( A_T, B_T, C_T, Q \)
   (c) If optimization is feasible set \( \bar{\gamma} = \gamma \), otherwise \( \gamma = \gamma - \bar{\gamma} \)

(d) Repeat from (a) until convergence

(iv) Repeat from (iii) until convergence or maximum iteration reached

Note that this method is only a heuristic for which convergence towards a global optimum, or even a local optimum, is not guaranteed. This said, in practice convergence has been achieved within 40 to 60 iterations.

VI. Numerical Experiment

We provide first a brief summary of the complete method. Computation of the \( \delta \nu \) part of the nu-gap, through an SDP approach, is done as follows

(1) Compute first the central transfer function \( T \) by

(a) Computing \( D_T \) by cholesky factorization, see Remark 3
(b) Computing an initial value for \( A_T, B_T, C_T \) from Section V-A
(c) Computing refined values for \( A_T, B_T, C_T \) from Section V-B

(2) Then use the obtained \( A_T, B_T, C_T, D_T \) to solve the LMI

(iii) in Lemma 5 which gives the solution to Problem 1, and hence to (7)

We illustrate now our approach with numerical examples, in which all LMI problems are solved in a MATLAB\textsuperscript{®} environment using YALMIP [46] together with the SeDuMi

<table>
<thead>
<tr>
<th>( P_1 ) and ( P_2 )</th>
<th>( A )</th>
<th>( B )</th>
<th>( C )</th>
<th>( D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1 )</td>
<td>-1/2</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( P_2 ) (Example 1)</td>
<td>1.25</td>
<td>4</td>
<td>1/2</td>
<td>-0.75</td>
</tr>
<tr>
<td>( P_2 ) (Example 3)</td>
<td>0.5</td>
<td>1</td>
<td>-1</td>
<td>6</td>
</tr>
</tbody>
</table>

| \( T \) (SOS) | \( A \) | \( B \) | \( C \) | \( D \) | \(|(I + CP)^{-T} T^T|_{\infty}\) |
|--------------|-------|-------|-------|-------|----------------|
| \( T \) (NL OPT) | 1.026 | -1.537| 1.435 | 1     | 0.081 |

<table>
<thead>
<tr>
<th>( T ) (Example 2)</th>
<th>( A )</th>
<th>( B )</th>
<th>( C )</th>
<th>( D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T ) (SOS)</td>
<td>-0.934</td>
<td>-0.993</td>
<td>0.955</td>
<td>1.25</td>
</tr>
<tr>
<td>( T ) (NL OPT)</td>
<td>1.295</td>
<td>-0.931</td>
<td>3.321</td>
<td>1.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( T ) (Example 3)</th>
<th>( A )</th>
<th>( B )</th>
<th>( C )</th>
<th>( D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T ) (SOS)</td>
<td>0.785</td>
<td>-0.724</td>
<td>0.930</td>
<td>1.6</td>
</tr>
<tr>
<td>( T ) (NL OPT)</td>
<td>0.845</td>
<td>-0.724</td>
<td>0.930</td>
<td>1.6</td>
</tr>
</tbody>
</table>
TABLE V. STATE-SPACE DATA FOR T (EXAMPLE 4)

|     | A    | B    | C    | D    | ||(I + P2P2') − T||∞ |
|-----|------|------|------|------|----------------------------|
| T (SISO) | 1.152 | 0.502 | 1.144 | -0.580 | 6.082 | 40.306 |
| T (NL OPTIM) | 1.121 | 0.502 | 1.121 | -0.580 | 6.082 | 0.053 |

TABLE VI. COMPUTATION OF δL2,(P1, P2)

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
<th>Example 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB gapmetric</td>
<td>0.832</td>
<td>0.488</td>
<td>0.181</td>
</tr>
<tr>
<td>Our Method</td>
<td>0.832</td>
<td>0.493</td>
<td>0.180</td>
</tr>
</tbody>
</table>

As a final note, the optimization cost of SDPs differ depending on the tolerance level and problem size. For example for the LMI in (4), let matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and let $n = n_r^2 + n_z$, be the number of decision variables, and $m$ the number of rows of LMI, then the asymptotic computational complexity, or flop cost, of SDP solvers is in $O(n^3m^{2.5} + m^{3.5})$ for SeDuMi [30], and in $O(m^3)$ for MATLAB LMI-lab [47]. These aspects could potentially be used to evaluate the computational cost of our method.

VII. CONCLUSION

We have presented a novel approach to accurately compute the $\nu$-gain metric through a Semi-Definite Programming (SDP) procedure. Since our optimization problem is SDP based, our approach may allow for the inclusion of additional constraints, such as closed-loop performance criteria. For systems of small size, the Sum-Of-Squares machinery used here is computationally tractable. However, for larger systems, dimensionality related problems may soon occur, due to the exponential growth in decision variables.

REFERENCES


