43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

From coefficients to samples: a new approach to SOS optimization

Johan Löfberg Pablo A. Parrilo
Automatic Control Laboratory
Swiss Federal Institute of Technology
Physikstrasse 3 - ETL - ETH Zürich
CH-8092 Zürich - Switzerland
{loefberg,parrilo}@control.ee.ethz.ch

***Abstract*— We introduce a new methodology for the numerical solution of semidefinite relaxations arising from the sum of squares (SOS) decomposition of multivariate polynomials. The method is based on a novel SOS representation, where polynomials are represented by a finite set of values at discrete sampling points. The techniques have very appealing theoretical and numerical properties; the associated semidefinite programs are better conditioned, and have a rank one property that enables a fast computation of the search directions in interior point methods. The results are illustrated with examples, and a preliminary implementation is compared with previous techniques.**

I. Introduction

Given a multivariate polynomial $p(x) \in \mathbb{R}[x] := \mathbb{R}[x_1, \ldots, x_n]$, we are interested in deciding whether there exist polynomials $f_i(x) \in \mathbb{R}[x]$ such that

$$p(x) = \sum_i f_i^2(x). \tag{1}$$

For simplicity, we discuss in this paper mainly the case of computing the sum of squares (SOS) representation of a given polynomial. However, the methods also apply, with the obvious modifications, to the case of sum of squares programs, i.e., the optimization over affine families of polynomials subject to sum of squares constraints. SOS programs have been applied to solve numerous questions in systems and control theory; see for instance [15], [13], [17], [9], the upcoming volume [6] and the references therein. Additionally, the techniques presented here can be applied towards the efficient solution of the Positivstellensatz-based relaxations of semialgebraic problems introduced in [15], [16], as well as the related ones in [10].

The standard method for computing a SOS decomposition can be briefly described as follows: given $p(x)$, we attempt to express it as a quadratic form in a new set of variables $\mathbf{u}$. A judicious choice of these new variables will depend on both the sparsity pattern and symmetry properties of p [14]. For the simplest case of a generic dense polynomial of total degree $2d$, the variables $\mathbf{u}$ will be all the monomials (in the variables $x_1, \ldots, x_n$) of degree less than or equal to d. Consequently, we attempt to express $p(x)$ as:

$$p(x) = \mathbf{u}^T Q \mathbf{u}, \qquad Q \succeq 0. \tag{2}$$

For the right- and left-hand sides to be equal, all the coefficients of the corresponding polynomials should be identical. Since Q is simultaneously constrained by linear equations and a PSD condition, the problem is *equivalent* to verifying whether a certain affine matrix subspace intersects the cone of positive definite matrices, and is therefore a semidefinite program (SDP).

The standard SOS formulation represents the input polynomial $p(x)$, as well as the $f_i(x)$, by using a standard monomial basis. In this paper we introduce a new representation and associated numerical method, obtained by combining a sampling-based technique along with an orthogonalization procedure. This will result in more efficient and numerically stable algorithms.

Earlier work [1], [8] had exploited in different ways some of the available structure in SOS programs, but only for the univariate case. For instance, Alkire and Vandenberghe [1] used the Toeplitz structure to obtain efficient methods for problems with autocorrelation constraints. The work by Genin *et al.* [8] uses displacement rank techniques for fast evaluation of Hessian and gradients, as well as a Chebyshev basis to improve the numerical conditioning.

A very appealing feature of our formulation, besides its applicability to the multivariate case, is its simplicity and transparency. It has been observed several times in the literature that the main reason why structured matrix computations via displacement rank can be done efficiently is the existence of an underlying polynomial algebra (see e.g. [12]). Our techniques bring this connection upfront, and completely dispense of any monomial or polynomial representation since they work directly with the corresponding functional values. Another advantage is that our approach converts convolution of coefficients to pointwise polynomial multiplication.

The method we propose can be interpreted in several ways. Besides the already mentioned, another approach is to consider the "standard" SOS formulation, and perform simultaneous nonsingular transformations on the primal and dual spaces:

- On the constraint side (dual), rather than matching coefficients we require the polynomials to agree on a fixed set of sample points.

- On the primal side, we parametrize the decision variables (i.e., the matrix Q) not by monomials in the standard basis, but by a set of polynomials that are orthogonal with respect to an atomic measure with support on the chosen samples.

The structure of the paper is as follows: we describe first the univariate trigonometric, to illustrate the ideas in familiar setting. Next we outline the general methodology in the multivariate case, and discuss its numerical and computational implications, followed by preliminary numerical results. We conclude by presenting our conclusions and future research directions.

II. The univariate, trigonometric case

In this section we describe first the methods in detail, focusing on the univariate trigonometric case. The reasons are mainly pedagogical, since it is a well-understood case where the relevant objects have been thoroughly studied in the past, and are very familiar to the engineering community.

Definition 1: A trigonometric polynomial of degree m has the form

$$p(t) = p_0 + \sum_{k=1}^{m} (p_k \cos kt + p_{-k} \sin kt). \tag{3}$$

In particular, a trigonometric polynomial of degree m has $2m+1$ coefficients, and is 2π-periodic.
We are interested in trigonometric polynomials which are nonnegative on the unit circle, i.e., those that satisfy $p(t) \geq 0, \forall t \in [0, 2\pi]$.

As is well-known, we can represent univariate trigonometric polynomials either by their coefficients ("time domain") or a finite set of values on the unit circle ("frequency domain"). The relationship between the two representations is given by the Discrete Fourier Transform (DFT), and the corresponding transformation can be efficiently computed using Fast Fourier Transform (FFT) techniques.

To explore in more detail this relationship, consider the so-called Dirichlet kernel:

$$K_N(t) := \frac{1}{N} \frac{\sin \frac{Nt}{2}}{\sin \frac{t}{2}},$$

where for the odd case $N = 2m+1$ we have:

$$K_N(t) = \frac{1}{N} \left(1 + 2 \sum_{k=1}^{m} \cos kt \right)$$

and in the even case $N = 2m+2$:

$$K_N(t) = \frac{2}{N} \sum_{k=0}^{m} \cos(k + \tfrac{1}{2})t.$$

This function has the property that $K_N(0) = 1$, and $K_N(2k\pi/N) = 0$, for $0 \neq k \in \mathbb{Z}$ (see Figure 1). The Dirichlet kernel is a classical construction, used for instance to prove results about the pointwise convergence of Fourier series. It can be interpreted as the result of Lagrange interpolation in N points, where all functional values are zero, except for a single point where it takes the value one.

Every trigonometric polynomial of degree m can be expressed in the Lagrange basis. Concretely, if $p(t)$ has degree m, we have:

$$p(t) = \sum_{k=-m}^{m} p(k\tau) \, K_{2m+1}(t - k\tau), \tag{4}$$

where $\tau = \frac{2\pi}{2m+1}$. This means that we can equivalently represent a polynomial of degree m by its $2m+1$ coefficients, or by the values it takes at $2m+1$ equidistant points.

A. SOS for trigonometric polynomials

The following lemma is essentially a restatement of the fact that nonnegative univariate polynomials are sums of squares.

Lemma 1: If $p(t)$ is a nonnegative trigonometric polynomial of degree N, then there exists a decomposition $p(t) = v^T Q v$, where $Q \in \mathcal{S}_+^{N+1}$. If $N = 2k+1$ is odd, then

$$v = [\cos(\tfrac{t}{2}), \sin(\tfrac{t}{2}), \ldots, \cos(kt + \tfrac{t}{2}), \sin(kt + \tfrac{t}{2})]^T,$$

otherwise $N = 2k$ and

$$v = [1, \cos(t), \sin(t), \ldots, \cos(kt), \sin(kt)]^T.$$

Proof: (Sketch) Using the substitution:

$$t = 2 \arctan x, \quad \Rightarrow \quad \cos t = \frac{1 - x^2}{1 + x^2}, \quad \sin t = \frac{2x}{1 + x^2},$$

we can convert a trigonometric polynomial into a rational function, whose denominator is a power of $(1 + x^2)$. Therefore, nonnegativity can be certified using the nonnegativity of the numerator. Since univariate polynomials are SOS, the desired result is obtained by rewriting the SOS decomposition of $\tilde{p}$ in the original variable t. ■
We can also write a decomposition that uses the interpolation kernels, as presented below.

Lemma 2: If $p(t)$ is a nonnegative trigonometric polynomial of degree N, then there exists a SOS decomposition

$$\begin{aligned} p(t) &= v^T Q v = \sum_{jk} Q_{jk} v_j v_k \\ &= \sum_{jk} Q_{jk} K_{N+1}(t - j\tau) K_{N+1}(t - k\tau) \end{aligned}$$

where $Q \in \mathcal{S}_+^{N+1}$, $\tau = \frac{2\pi}{N+1}$, and

$$v = [K_{N+1}(t) \;\; K_{N+1}(t - \tau) \;\; \ldots \;\; K_{N+1}(t - N\tau)]^T \tag{5}$$

This decomposition has the property that the diagonal elements satisfy $Q_{kk} = p(k\tau)$.
The theorem follows directly from Lemma 1, by expressing the vectors v in the statement of the lemma in terms of the Dirichlet kernel via (4).

Remark 1: Notice that in the odd N case, the polynomial $p(t)$, while being a sum of squares, is not a sum of squares

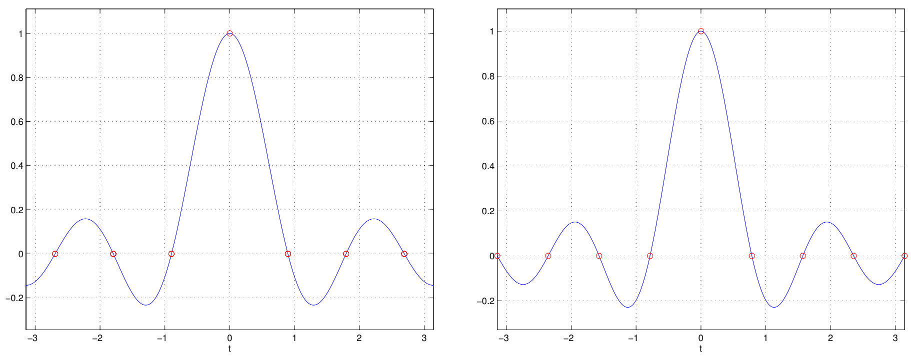

Fig. 1. The Dirichlet interpolation kernels for $N = 7$ and $N = 8$.

of *trigonometric* polynomials, in the sense defined above, but rather of polynomials in the *half-angle* $\frac{t}{2}$.

Example 1: Consider the trigonometric polynomial of degree $N = 2$

$$p(t) = 5 + 4\cos(t) - 2\sin(t) + 2\cos(2t).$$

After the transformation, we have

$$\tilde{p}(x) = \frac{3x^4 - 4x^3 - 2x^2 - 4x + 11}{(1 + x^2)^2},$$

and the representation:

$$p(t) = \begin{bmatrix} 1 \\ \cos t \\ \sin t \end{bmatrix}^T \begin{bmatrix} 2 & 2 & -1 \\ 2 & 5 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \cos t \\ \sin t \end{bmatrix}.$$

We can convert this representation to the Dirichlet basis. Letting $\alpha = \frac{\sqrt{3}}{2}, \tau = 2\pi/3$, we have the identities:

$$\begin{bmatrix} 1 \\ \cos t \\ \sin t \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \alpha & -\alpha \end{bmatrix} \begin{bmatrix} K_3(t) \\ K_3(t-\tau) \\ K_3(t-2\tau) \end{bmatrix},$$

and therefore the alternative SOS representation:

$$p(t) = \begin{bmatrix} K_3(t) \\ K_3(t-\tau) \\ K_3(t-2\tau) \end{bmatrix}^T \begin{bmatrix} 11 & \frac{1}{2} - \alpha & \frac{1}{2} + \alpha \\ \frac{1}{2} - \alpha & 2 - 2\alpha & \frac{1}{2} \\ \frac{1}{2} + \alpha & \frac{1}{2} & 2 + 2\alpha \end{bmatrix} \begin{bmatrix} K_3(t) \\ K_3(t-\tau) \\ K_3(t-2\tau) \end{bmatrix}.$$

In particular, notice that the diagonal elements are exactly $p(0), p(2\pi/3), p(4\pi/3)$.

Remark 2: Lemma 1 requires the solution of an $(N+1) \times (N+1)$ SDP with a *real symmetric* constraint, as opposed to the complex Hermitian one required by Fejér's representation theorem for nonnegative trigonometric polynomials. This has several numerical advantages. We can also arrive at the same result by coupling Fejér's theorem and a simple symmetry reduction argument; the details are omitted in this version of the paper.

III. Properties and Advantages

The representation in Lemma 2 has many advantages, both conceptual and numerical. Many of these are consequences of the interesting properties of the Dirichlet kernel. For instance, we have the following:

Theorem 1: Let $w := \frac{2\pi}{N}$. Then, the Dirichlet kernel $K_N(t)$ satisfies:

$$\sum_{k=0}^{N-1} K_N^2(t - kw) = 1. \quad (6)$$

This implies that the vector v in (5) has norm equal to one for all values of t. Therefore, we have the inequalities

$$\lambda_{min}(Q) \leq \min_t p(t) \leq Q_{kk},$$

so we can obtain very simple upper and lower bounds of the minimum from *any* quadratic representation of $p(t)$.

Example 2: Consider again the polynomial from Example 1. We have

$$0.1676 \approx \lambda_{min}(Q) \leq \min_t p(t) \approx 0.2067$$

It is also interesting to notice the connections with Hermite interpolation. If the matrix Q is diagonal, then $\tilde{p}(t) := \sum_k p(k\tau) K_N^2(t - k\tau)$, thus providing a trigonometric polynomial interpolating the given samples, and having zero derivative at the interpolation points. If all the $p(k\tau)$ are nonnegative numbers, then $\tilde{p}(t) \geq 0$ for all t.

Despite the fact that the number of samples needed for a polynomial of degree N is always odd $(2N + 1)$, there are also convenient numerical advantages in choosing an even number of equispaced points in connection with this representation. Notice that one of the samples is redundant, since we only need to determine $2N + 1$ coefficients. However, it is convenient to include this extra sample, since it can be computed at no extra cost, and as discussed later, additional data can enhance the numerical stability.

The functional values of $p(t)$ at the equispaced points $t = k\tau/2$ can be obtained very easily from the matrix Q, since

- The values at the even-numbered points are exactly the diagonal elements of Q.
- The values at the odd-numbered roots are obtained as $p(t) = v_i^T Q v_i$, for some suitable vectors v_i. The vectors v_i are cyclic permutations of a single vector, whose components are obtained by evaluating the Dirichlet kernel $K_N(t)$ at the points $(2k + 1)\tau/2$. All

products can be efficiently computed using FFT and interleaving operations.

Notice that these two properties imply that in the SDP formulation, the constraint matrices either have only one diagonal element, or have rank one. As we will see, these appealing features fully extend to the multivariate case.

IV. SDP for general multivariate polynomials

In the previous section we used the specific structure of trigonometric polynomials and Lagrange interpolants. While these translate to a certain extent to the multivariate case, in that case the results would not be as clean and convenient due to the potential ill-conditioning of the transformation between samples and coefficients.

Nevertheless, by far the most important feature in the approach has been to impose the equality constraint between the left- and right-hand sides of (1) at specially chosen sample points, and not between the coefficients of the polynomial, as in the standard SOS approach. This readily extends to multivariate polynomials, and forms the core of our general method. We dispense altogether of any polynomial representation by coefficients, and operate mostly with the functional values.

We assume the polynomial p is defined by N coefficients; for instance, for dense polynomials of degree $2d$, then $N = \binom{n+2d}{2d}$. The general sum of squares problem is then given by:

$$Q \succeq 0, \qquad \text{s.t.} \quad p(x_i) = v(x_i)^T Q v(x_i), \quad i = 1, \ldots, N,$$

where $v(x_i)$ are polynomials $v(x)$, evaluated at given points $x_i \in \mathbb{R}^n$. We will delay the discussion on the specific choice of the samples x_i and the polynomial basis $v(x)$ for Section VI, and proceed here with the developments.

Similarly to the trigonometric case described in earlier sections, we need a nonsingularity condition on the points x_i to guarantee that we are able to recover the polynomial $p(x)$ from its sample values. In the interpolation parlance, we require the corresponding interpolation problem to be *poised* (see for instance the survey [7], and the references therein), meaning that the only polynomial with the required monomial structure that vanishes in the chosen samples is identically zero. This requires that the matrices $v(x_i)v(x_i)^T$ must span an N-dimensional subspace. We define the matrix V as:

$$V = [v_1 \ldots v_N]^T = [v(x_1) \ldots v(x_N)]^T. \tag{7}$$

Notice that V has a multivariate Vandermonde structure. We proceed assuming the nonsingularity condition holds; more details on this on the next section.

By defining the vector $b = [p(x_1) \ldots p(x_N)]^T$ and the matrices $A_i = v_i v_i^T$, we see that we have an SDP feasibility problem in primal form

$$Q \succeq 0, \qquad \text{s.t.} \quad b_i = Q \bullet A_i, \quad i = 1, \ldots, N. \tag{8}$$

A. Optimizing lower bounds

An important application of SOS techniques is constrained or unconstrained polynomial optimization problems. For the simplest unconstrained case we can obtain a lower bound on the optimal solution by considering the problem $\max t, p(x) - t$ is SOS. This has the SDP formulation

$$\max_{Q \succeq 0, t} t, \qquad p(x_i) - t = v(x_i)^T Q v(x_i)$$

The dual of this problem can be shown to be

$$\begin{array}{rll} \min\limits_{S \succeq 0, y} & b^T y & \\ \text{subject to} & S - V^T \mathrm{diag}(y) V & = 0 \\ & \sum_{i=1}^N y_i & = 1 \end{array}$$

This dual form gives some further insight to our new formulation. To find a lower bound on a polynomial, we minimize a linear combination of given function values, while constraining the corresponding linear combination of the outer products of the kernels to be positive semidefinite. We also note that y can be interpreted as the homogeneous barycentric coordinates of the minimizer. More importantly though, the dual form will enable us to perform a very appealing transformation of the SDP.

V. Numerical solutions and rank one SDPs

In this section we propose a new method for the numerical solution of SOS-based SDPs, based on the developments of the previous sections.

A. Orthogonalization

For numerical reasons, it is desirable to do a coordinate transformation on the dual problem. Effectively, we can perform a congruence transformation that will render the columns of V orthogonal to each other. This can be efficiently achieved by using "economy sized" SVD or QR decompositions. In the latter case, we have $V = QR$, where Q has the same dimensions as V and R is square upper triangular and nonsingular. The problem therefore reduces to:

$$\begin{array}{rll} \min\limits_{S \succeq 0, y} & b^T y & \\ \text{subject to} & S - Q^T \mathrm{diag}(y) Q & = 0 \\ & \sum_{i=1}^N y_i & = 1 \end{array}$$

The primal problem will be transformed correspondingly.

The columns of Q can now be interpreted as the values that a set of polynomials takes on the sample points. These are in fact a set of *orthogonal polynomials*, under the inner product given by a discrete measure with support on the chosen samples. In particular, notice that the class of orthogonal polynomials is not chosen *a priori*, but rather is obtained as a subproduct of the QR procedure. This has obvious advantages in the case of minimization over arbitrary compact semialgebraic sets, where in general no "good" polynomial bases are available.

B. Rank one SDPs in interior point methods

In the primal SDP (8), we immediately notice a very particular feature: the constraint matrices $A_i = v_i v_i^T$ all have *rank one*. This kind of structure can be fruitfully exploited when computing gradients and Hessians of a logarithmic barrier function, or the Newton system in a primal-dual interior point method. For instance, the dual SDP solver DSDP by Benson *et al.* [2] is one among the few that currently exploit this, in combination with sparsity. Important to note though is that the idea exploited in [2] can be extended to primal and primal-dual solvers also.

A major bottleneck in interior-point implementations of SDP solvers is often the Hessian assembly, i.e. the mere construction of the Hessian. The $N \times N$ Hessian is given by $H_{ij} = (WA_i) \bullet (UA_j)$, where the matrices W and U depend on the interior-point algorithm used. In a primal solver $W = U = Q$, a dual solver uses $W = U = S^{-1}$ while a primal-dual solver based on, e.g., the XZ direction uses $W = S^{-1}, U = Q$ [3]. Regardless of algorithm, we note that $H_{ij} = (Wv_i v_i^T) \bullet (Uv_j v_j^T) = (v_j^T W v_i)(v_i^T U v_j)$ Hence, the Hessian is given by $(VWV^T) \circ (VUV^T)$. Compilation of the Hessian and gradient can thus be performed in $O(N^3)$ operations, and neatly written as simple dense matrix-vector manipulations.

For unstructured A_i the Hessian assembly is an $O(N^4)$ operation since N matrix multiplications of $N \times N$ matrices are performed. Note though that if we solve the original SOS problem using a standard coefficient based approach, the matrices A_i are sparse. By exploiting this sparsity, the practical complexity could be better than $O(N^4)$. The benefit of the proposed formulation is that the whole problem is defined by one dense matrix V, and all computations can be done using highly optimized level 3 BLAS matrix-matrix operations.

VI. Sample location and conditioning

In principle, as long as the matrix V in (7) satisfies the poisedness condition, the presented formulation is exactly equivalent to the standard one. Nevertheless, the location of the sampling points can clearly have a big effect on the numerical conditioning of the resulting SDP.

Remark 3: As opposed to most classical interpolation problems, the condition number of the transformation is not directly relevant in this problem. We do not care so much about the possible ill-conditioning of the transformation between coefficients and sample values, as we do about our ability to *extrapolate* from the given samples the behavior of the polynomial in the rest of the feasible set.

For the multivariate case, it is not clear what the "best" location for the samples is, and it seems likely that an exact answer to such a question is a difficult problem. It should be noted that several questions of this kind are open: for instance, the location of optimal interpolation points in the unit sphere, or the "best" estimate for the minimum of polynomial on a compact domain using only N samples. This relates to the classical open problem of computing Lebesgue constants in Lagrange interpolation. Nevertheless, many special cases have been analyzed, and some relatively good heuristics are known for the general case.

We mention the following well-known results:

- Trigonometric polynomials: as mentioned in a previous section, in this case a set of optimal samples are equidistant points in $[-\pi, \pi]$.
- Univariate polynomials on an interval $[a, b]$: the Chebyshev points $a + (b - a)\cos(k\pi), k = 0, \ldots, N$ are known to be near-optimal.
- Dense polynomials in n variables and degree $2d$. It is well-known [7] that for the set of rational points in the unit simplex with denominator $2d$ the interpolation problem is poised.

In general, it is easy to randomly generate sets for which the interpolation problem is poised; their numerical properties can however be poor. A more complete analysis is certainly necessary.

Another important point to be mentioned is that our methodology extends in a very natural way to the case of *redundant* samples, i.e., more than the minimum number required to define the subspace. This allows the use of known good point sets, although of cardinality that is not minimal, like tensor products of univariate Chebyshev grids. In the SDP, these extra points will translate into redundant constraints, that nevertheless carry important information.

VII. Implementation

The rank one structure can be exploited in almost any SDP algorithm, so it is not clear if one should use a primal, dual or a primal-dual solver.

1) Primal solver: The main motivations for using a primal solver is that the Hessian and gradient are defined using essentially only the primal variable Q and the algorithm involves no inverses that might lead to numerical instability when the optima is approached [3] . A problem is that we do not have a feasible initial solution Q, hence a big-M or more advanced infeasible algorithm is needed [19].

2) Dual solver: A dual solver benefits from simple construction of an initial feasible solution, namely $y_i = N^{-1}$. Additionally, the complexity in each iteration of the interior-point algorithm is very low as we will see below. A possible complication of a dual solver is that the Hessian involves the inverse of the dual slack S. For our SOS problems, this matrix will not only be singular in the optima, but typically we have only a small number of non-zero eigenvalues. Hence, numerical problems as the optima is approached can be an issue.

3) Primal-dual solver: The standard choice to solve SDPs is a primal-dual algorithm. The reason is that the number of interior-point iterations typically is lower than what is necessary in a strictly primal or dual solver. However, although the iteration count might be decreased, the complexity in each iteration is higher since more dense matrix manipulations are performed, and many of these cannot exploit our rank one structure.

A. Implementation of a dual solver

As a proof of concept, we outline a basic implementation of a dual solver, and report some initial computational results. A dual approach was chosen due to its simplicity. To keep things clear, we implement a classical logarithmic barrier approach [5], [4]. More advanced dual algorithms differ essentially in how they update the barrier parameter μ, in order to decrease the number of iterations.

Given : Model V and b, feasible initial y, initial barrier parameter μ and parameters $0 < \tau, \theta < 1$
repeat
Update barrier parameter $\mu := \theta\mu$
repeat
Calculate Hessian H and gradient g
Calculate search direction p
Calculate suitable step length r
Update $y := y + rp$
until $p^T H p \leq \tau$
until $\mu \leq \frac{\epsilon}{4n}$

The Hessian was derived above and requires two dense matrix multiplications. The gradient of a logarithmic barrier function for our SDP is $g = \mu^{-1}b - \text{diag}(VS^{-1}V^T)$, hence it requires no major additional computations since all data is available once the Hessian is calculated. The main additional computational effort lies in the solution of $Hp = g$, and step-length selection using a Cholesky based back-tracking on the dual slack $S = V^T\text{diag}(y + rp)V$.

B. Computational results

The simple dual solver described above was implemented[1] and applied to trigonometric polynomials of the type (3) for polynomial degrees m ranging from 10 to 200. The SDP problems were solved to an absolute accuracy $\epsilon = 10^{-6}$ using $\theta = 0.2$ and $\tau = 0.9$. Averaged CPU time is reported in the figure below.

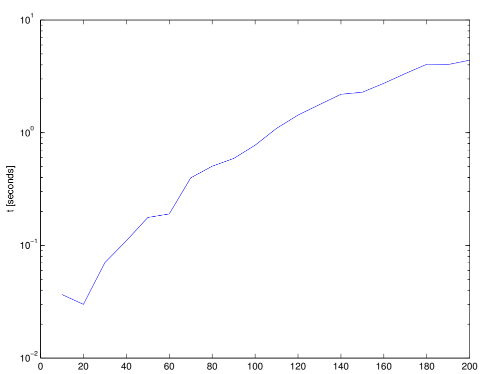

Fig. 2. Computational results for SOS decomposition of trigonometric polynomials using a structure exploiting dual solver.

Considering the fact that the solver is based on a very simple dual algorithm with fixed update of μ, and is implemented entirely in MATLAB code using less than 100 lines of code, the results are encouraging.

[1]MATLAB 6.5.1, 2.2GHz Intel processor with 512 Mb memory

VIII. Conclusions

We have shown the conceptual and algorithmic advantages of an alternative formulation for the numerical solution of SDPs arising from the sum of squares decomposition for multivariate polynomials. The results notably increase the applicability of the techniques, and promise very good practical performance based on our preliminary implementation. The integration with sum-of-squares software SOSTOOLS [18] and YALMIP [11] will be completed in the near future.

References

[1] B. Alkire and L. Vandenberghe. Convex optimization problems involving finite autocorrelation sequences. *Mathematical Programming Ser. A*, 93(3):331–359, 2002.
[2] S. Benson, Y. Ye, and X. Zhang. *DSDP, A Dual Scaling Algorithm for Positive Semidefinite Programming*. Available from http://www-unix.mcs.anl.gov/˜benson/dsdp/.
[3] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization*. Kluwer Academic Publishers, 2002.
[4] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming: Algorithms and Complexity*. Mathematics and Its Applications. Kluwer Academic Publishers, 1994.
[5] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Systems & Control: Foundations and Applications. John Wiley and Sons, 1968.
[6] A. Garulli and D. Henrion. Positive polynomials in Control. Springer Lecture Notes in Computer Science, to appear, 2004.
[7] M. Gasca and T. Sauer. Polynomial interpolation in several variables. *Adv. Comput. Math.*, 12(4):377–410, 2000.
[8] Y. Genin, Y. Hachez, Y. Nesterov, and P. V. Dooren. Optimization problems over positive pseudopolynomial matrices. *SIAM J. Matrix Anal. Appl.*, 25(1):57–79 (electronic), 2003.
[9] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard. Some controls applications of sum of squares. In *Proceedings of the 42th IEEE Conference on Decision and Control*, pages 4676–4681, 2003.
[10] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2001.
[11] J. Löfberg. *YALMIP 3*, 2004. http://control.ee.ethz.ch/˜joloef/yalmip.msql.
[12] B. Mourrain and V. Y. Pan. Multivariate polynomials, duality, and structured matrices. *J. Complexity*, 16(1):110–180, 2000. Real computation and complexity (Schloss Dagstuhl, 1998).
[13] A. Papachristodoulou and S. Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *Proceedings of the 41th IEEE Conference on Decision and Control*, 2002.
[14] P. Parrilo. Exploiting structure in sum of squares programs. In *Proceedings of the 42th IEEE Conference on Decision and Control*, 2003.
[15] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, May 2000. Available at http://www.cds.caltech.edu/˜pablo/.
[16] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Prog.*, 96(2, Ser. B):293–320, 2003.
[17] S. Prajna and A. Papachristodoulou. Analysis of switched and hybrid systems – beyond piecewise quadratic methods. In *Proceedings of the American Control Conference*, 2003.
[18] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2002. Available from http://www.cds.caltech.edu/sostools and http://control.ee.ethz.ch/˜parrilo/sostools.
[19] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, Mar. 1996.