FINDING A COMMON QUADRATIC LYAPUNOV FUNCTION USING CONICAL HULLS

Rianto Adhy Sasongko
Control and Power Group, Department of Electrical and Electronic Engineering, South Kensington Campus, Imperial College London, London SW7 2AZ, United Kingdom

J. C. Allwright
Control and Power Group, Department of Electrical and Electronic Engineering & Centre for Process Systems Engineering, South Kensington Campus, Imperial College London, London SW7 2AZ, United Kingdom

Keywords: Positive semidefinite matrices, cone, conicall hull.

Abstract: Consider a set of linear time-invariant continuous-time systems that is a convex hull with vertices formed by a given set of systems. The problem of finding a common Lyapunov function v, specified in terms of a symmetric positive definite matrix, for the convex hull of systems is tackled by searching for a symmetric positive definite (PD) matrix P which causes dv/dt to be negative definite for each vertex system. The approach involves an extension of an existing method for solving optimization problems for positive semidefinite (PSD) matrices that is based on a representation of the cone of PSD matrices as a conical hull. The condition that the derivative of the Lyapunov function for each vertex system is negative definite is converted naturally into the condition that the matrix P belongs to the interior of the intersection of several conical hulls: one for each vertex system to ensure $d\mathrm{v}/\mathrm{dt}$ for it is negative definite. The determination of a P in the intersection is viewed as the solution of a quadratic programme on the product space of the cones. Then the existing theory and algorithms for conical hull problems are adapted to the solution of the quadratic programme. The numerical results suggest that the proposed algorithm is faster than the projective method used in MATLAB for small problems. Effort is being devoted to improve it for larger problem.

1 INTRODUCTION

For a family of dynamical linear continuous-time systems, obtaining a common Lyapunov function (CLF) is desirable to ensure stability of all systems in the convex hull of that set of systems or stability when the system transits from one vertex of the convex hull to another at any time (D.Liberzon, 2003). The importance of a common Lyapunov function for stabilization of switched linear systems is discussed in (G.Xie, 2004), and the existence of a common Lyapunov function for a set of asymptotically stable systems is examined in (N.A.Bobylev, 2002) and (R.N.Shorten, 2003). An algorithm for obtaining a common Lyapunov function, called the gradient method, is discussed in (D.Liberzon, 2003). A semidefinite programming algorithm, called the projective method, (A.Nemirovskii, 1994) can also be used for solving such problems.

Here a new algorithm is proposed for finding a common Lyapunov function; it is based on the conical hull approach to the description of the set of symmetric positive semidefinite matrices, which will be introduced later.

Let $\mathbb{S}$ denote the set of symmetric matrices from $\mathbb{R}^{n\times n}$. Let $\mathbb{S}_{\geq}$ be the set of positive semidefinite matrices, and $\mathbb{S}_{>}$ the subset of positive definite matrices, from $\mathbb{S}$.

For a set of N stable $A_i \in \mathbb{R}^{n\times n}$, $i = 1 : N$, the function $\mathrm{v}(x) = x^T P x$ is a common Lyapunov function for the A_i if $P \in \mathbb{S}$ satisfies

$$A_i^T P + P A_i < 0, \quad \forall i \in [1 : N] \tag{1}$$

Solving (1) for such a P is a semidefinite programming problem with the symmetric positive definite matrix P as the variable.

In the following analysis, it will be convenient to describe such P by $\mathrm{vec(P)}$, which is the vector consisting of the entries of P in a certain order. In detail, $\mathrm{vec}(P) = [P(1,:)^T; P(2,:)^T; \ldots; P(n,:)^T] \in \mathbb{R}^{n^2}$ (using MATLAB notation). Of course, $P = \mathrm{vec}^{-1}(\mathrm{p})$. Consequently condition (1) can be written as:

$$\begin{array}{l} M_i p \in \mathrm{vec}(\mathbb{S}_{>}) \ , \ \forall i \in [1 : N] \\ p \in \mathrm{vec}(\mathbb{S}_{>}) \end{array} \tag{2}$$

for

$$M_i = -(A_i^T \otimes I + I \otimes A_i^T) \in \mathbb{R}^{n^2\times n^2} \tag{3}$$

Adhy Sasongko R. and C. Allwright J. (2006).
FINDING A COMMON QUADRATIC LYAPUNOV FUNCTION USING CONICAL HULLS.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 113-118
DOI: 10.5220/0001218001130118

where $\otimes$ denotes the *Kronecker product*.

Since the A_i are asymptotically stable, the M_i are non-singular. Therefore $P = \mathrm{vec}^{-1}(p)$ solves (1) if and only if

$$\begin{array}{l} p \in M_i^{-1}\mathrm{vec}(\mathbb{S}_>), \forall i \in 1:N \\ p \in \mathrm{vec}(\mathbb{S}_>)\ , \end{array} \tag{4}$$

or if and only if $p \in \mathrm{int}(\mathbb{I})$, the interior of $\mathbb{I}$, where I is defined by

$$\mathbb{I} = \left\{ \bigcap_{i=1}^{N} M_i^{-1}\mathrm{vec}(\mathbb{S}_>) \right\} \tag{5}$$

The solution p needs to be in the interior of $\mathbb{I}$ to guarantee the positive definiteness of the corresponding matrix $P = \mathrm{vec}^{-1}(\mathrm{p})$.

Hence, there is a CLF of the form $\mathrm{v}(x) = x^T P x$ if and only if

$$\mathbb{I} \neq \emptyset \tag{6}$$

and the problem of solving (1) is equivalent to that of finding an interior point p of the intersection $\mathbb{I}$, assuming it has one.

The main contribution here is the development of a computational method for finding a vector p in $\mathrm{interior}(\mathbb{I})$, if there is one.

We have chosen to investigate the above formulation since the other obvious formulation

$$\min_{\substack{P_i \in \mathbb{S}_\geq \\ i=[0:N]}} \sum_{i=1}^{N} \|P_i + A_i^T P_0 + P_0^T A_i\|^2 + \|P_0 - \beta I\|^2 \tag{7}$$

leads to one additional conical hull being involved (the purpose of β, a positive scalar, will become clear near (22)).

The material is presented in the following way. Some descriptions and properties of a conical hull representation of the set of symmetric PSD matrices is given in Section 2. An approach for finding a point in the interior of $\mathbb{I}$ using the conical hull approach is outlined in Section 3. More detail about the solution of the CLF problem using conical hulls is also given in this section. Numerical procedure and some results for the CLF problem are given in Section 4. Finally, the conclusions are given in Section 5.

2 A CONICAL HULL FOR PSD MATRICES

2.1 Conical Hull Characterization for Symmetric PSD Matrices

The background theory of symmetric matrices can be found in various references, some of them are (R.A.Horn, 1985) and (G.Strang, 1988). The definitions of this section and the basic conical hull theory and approach used throughout this paper are from (J.C.Allwright, 1988) and (J.C.Allwright, 1989); most of the proofs can be found in these papers.

Symmetric PSD matrices are defined uniquely by their diagonal and upper triangle of elements, so $\mathrm{vec}(S) = \{\mathrm{vec}(A) : A \in \mathbb{S}\}$ is a subset of a linear subspace of $\mathbb{R}^{n^2}$ that has dimension $r = n(n+1)/2$. An orthonormal basis for that subspace can easily be found. For instance, for the case $n = 3$, suitable basis vectors are $w_1 = e_1, w_2 = (e_2 + e_4)/\sqrt{2}, w_3 = (e_3 + e_7)/\sqrt{2}, w_4 = e_5, w_5 = (e_6 + e_8)/\sqrt{2}, w_6 = e_9$ where $e_i = [0\,0\ldots0\,1\,0\ \ldots\ 0]'$ with the 1 in row i. The corresponding basis matrix is

$$W = [w_1\ w_2\ \ldots\ w_r] \in R^{n^2 \times r} \tag{8}$$

and it is easy to check that $W^T W = I_r$, $R[W^T] = \mathbb{R}^r$ and $R[W] = \mathrm{vec}(S)$.

Using this basis, $\mathrm{vec}(A) \in \mathbb{R}^{n^2}$ can be written in terms of the vector $\overline{\mathrm{vec}}(A) \in \mathbb{R}^r$ as

$$\mathrm{vec}(A) = W\overline{\mathrm{vec}}(A) \in R^{n^2} \tag{9}$$

where

$$\overline{\mathrm{vec}}(A) = W^T \mathrm{vec}(A) \in R^r. \tag{10}$$

Since, generically, $\overline{\mathrm{vec}}(A)$ contains no redundant information regarding symmetric A, it can be thought of as a minimal description of such an A.

Now any matrix $A \in \mathbb{S}_\geq$ can be written as $A = M^2$ for some $M \in \mathbb{S}$. For example, if the spectral form of A is written as $V\Lambda V^{\mathrm{T}}$ with $\Lambda = \mathrm{diag}[\lambda_1\,\lambda_2\,\ldots\,\lambda_{\mathrm{n}}]$ then such an M is $V\ \mathrm{diag}(\sqrt{\lambda_1}\ldots\sqrt{\lambda_{\mathrm{n}}})\mathrm{V}^{\mathrm{T}}$. Letting $B = M(\|M\|_F)^{-1}$, where $\|M\|_F := (\sum_{i,j=1}^{n} m_{ij}^2)^{0.5}$ is the Frobenius norm, we see that our general A can be written as $A = \alpha B^2$ for $\alpha \in [0, \infty)$ and $B \in \mathbb{U} := \{B \in \mathbb{S} : \|B\|_F = 1\}$. This suggests, correctly, that $S_\geq = \mathrm{cone}\{\mathrm{B}^2 : \mathrm{B} \in \mathbb{U}\}$ where $\mathrm{cone}(\mathrm{X}) := \{\alpha \mathrm{x} : \alpha \in [0, \infty), \mathrm{x} \in \mathrm{X}\}$ is the conical hull of X.

In fact, it will be more convenient to use

$$\mathrm{vec}(\mathbb{S}_\geq) = \mathrm{cone}(\Omega) \tag{11}$$

where

$$\Omega = \mathrm{conv}\,\{\Psi\} \tag{12}$$

for

$$\begin{array}{l} \Psi = \left\{\mathrm{vec}(B^2) : B \in U\right\} \subset R^{n^2} \\ U = \left\{B \in R^{n \times n} : B^T = B\ ,\ \|B\|_F = 1\right\} \end{array} \tag{13}$$

and it is obvious that the set $\overline{\mathrm{vec}}(\mathbb{S}_\geq)$ that contains only the minimal description of every symmetric positive semidefinite matrix can be written as:

$$\overline{\mathrm{vec}}(\mathbb{S}_\geq) = \mathrm{cone}(W^T\Omega) \tag{14}$$

2.2 Properties of our Conical Hull of Symmetric PSD Matrices

Some definitions and properties of the conical hull of PSD matrices are presented in this section; this will be needed for the development of the algorithm. Recall that $\mathrm{vec}(\mathbb{S}_{\geq}) = \mathrm{cone}(\Omega)$ and $\overline{\mathrm{vec}}(\mathbb{S}_{\geq}) = \mathrm{cone}(\mathrm{W}^{\mathrm{T}}\Omega)$ for Ω of (12). Since the set $\mathbb{S}_{>}$ of symmetric positive definite matrices is the interior of the set $\mathbb{S}_{\geq}$

$$\overline{\mathrm{vec}}(\mathbb{S}_{>}) = \mathrm{int}(\mathrm{cone}(W^T\Omega)) \tag{15}$$

It follows from the definition (12) that Ω is a non-empty, convex, and compact set, and (not so obviously) that:

$$\min_{x \in \Omega} \quad \|x\|_2 \;=\; n^{-1/2} \tag{16}$$

A consequence of this fact is that:

$$\min_{x \in LW^T\Omega} \quad \|x\|_2 \;= \sigma_{\min}(L) n^{-1/2} \tag{17}$$

for all $L \in R^{r\times r}$, where $\sigma_{min}(L)$ is the minimum singular value of L.

It is interesting that Ω is a subset of an affine subset of R^r, which is actually the intersection of an affine set and $(\overline{\mathrm{vec}}(S_{\geq})$, which is a cone. Hence, somewhat surprisingly, Ω is a 'flat' set. This is illustrated in the following figure:

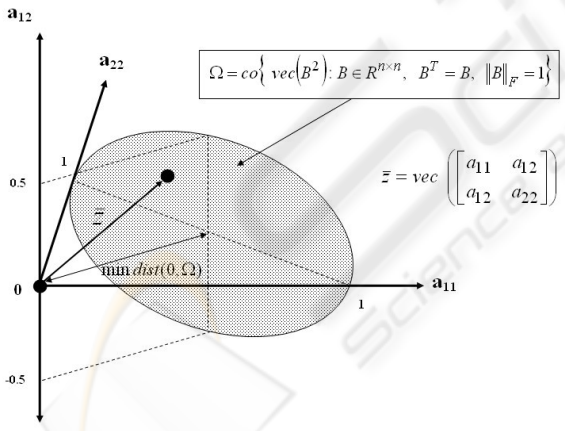

Figure 1: Depiction of Ω.

For the determination of feasible descent directions later, it will be important to be able to solve easily, for any vector $\bar{g} \in \mathbb{R}^r$, the optimization problem:

$$\min_{\gamma \in LW^T\Omega} \quad \bar{g}^T\gamma \tag{18}$$

It turns out that a minimizer $\hat{\gamma}$ can be found easily as

$$\hat{\gamma} = LW^T \mathrm{vec}(\nu\nu^T) \tag{19}$$

where ν is a normalized eigenvector corresponding to the minimum eigenvalue of Z, with Z defined as:

$$Z = [\mathrm{vec}^{-1}(\bar{g}) + \mathrm{vec}^{-1}]/2 \in R^{n\times n} \tag{20}$$

for

$$\bar{g} = WL^Tg \tag{21}$$

3 OPTIMIZATION ON CONICAL HULLS

Conceptually the CLF problem is that of finding a $\bar{z} \in \mathrm{int}(\bigcap \mathbb{K}_i)$ where the $\mathbb{K}_i$ are the convex sets $\mathrm{cone}(W^T M_i \Omega)$. The approach will be to first find a point in the intersection $\bigcap \mathbb{K}_i$ and then, if it is not in the interior, to perturb it into the interior (if there is one).

The basic idea for finding a point in $\bigcap \mathbb{K}_i$ is to find a $\bar{z}_i$ in each $\mathbb{K}_i$ such that the distance between all $\bar{z}_i$ is minimized. Then, if the minimal distance is zero, we have found a point $\hat{\bar{z}}$ which is in the intersection $\bigcap \mathbb{K}_i$. Obviously the origin is not in the interior of the intersection, so to encourage the optimization algorithm to find a non-zero point, a penalty term $\|\bar{z}_1 - \overline{\mathrm{vec}}(\beta I)\|^2$, with positive real β, is added to the cost to attract $\overline{\mathrm{vec}}^{-1}(\bar{z})$ towards the positive definite matrix βI. Hence the problem is

$$\min_{\substack{\bar{z}_i \in \mathbb{K}_i \\ i=[1:N]}} \sum_{\substack{i,j\in[1:N] \\ i\neq j}} \|\bar{z}_i - \bar{z}_j\|^2 + \|\bar{z}_1 - \overline{\mathrm{vec}}(\beta I)\|^2 \tag{22}$$

which is a convex problem. Next a cone is constructed as:

$$\mathbb{K} = \{\mathbb{K}_1 \times \mathbb{K}_2 \times \ldots \times \mathbb{K}_N\} \tag{23}$$

Then the optimization problem (22) can be reformulated as :

$$\min_{\bar{z} \in \mathbb{K}} \quad \mathbf{v}(\bar{z}) \tag{24}$$

where,

$$\begin{array}{l} \mathbf{v}(\bar{z}) = \bar{z}^T\bar{Q}\bar{z} + \bar{F}\bar{z} \\ \bar{Q} = \sum_{i,j=1,\ i\neq j}^{N} \bar{P}_{E_{ij}}^T \bar{P}_{E_{ij}} + \bar{P}_1^T\bar{P}_1 \\ \bar{F} = -2\bar{\beta}^T\bar{P}_1 \\ \bar{P}_{E_{ij}} = \bar{P}_i - \bar{P}_j \end{array} \tag{25}$$

Here $\bar{\beta} = \overline{\mathrm{vec}}(\beta I)$, $\bar{z} = (\bar{z}_1, \ldots, \bar{z}_N)^T$ with each $\bar{z}_i \in \mathbb{K}_i$, and the matrix $\bar{P}_i$ selects $\bar{z}_i$ from $\bar{z}$, in that $\bar{z}_i = \bar{P}_i\bar{z}$.

The set $\mathbb{K}$ itself is a convex cone, so the algorithm of (J.C.Allwright, 1988) for finding the closest point in $\mathrm{cone}(\Gamma)$ to a point d, i.e. for solving

$$\min_{x \in \mathrm{cone}(\Gamma)} \quad \|d - x\|^2, \tag{26}$$

can be adapted to solve the optimization problem above because $\|d-x\|^2 = \|d\|^2 - 2d^Tx + \|x\|^2$ and

this has the same form as the cost of (22).

To proceed further, the notion of a truncated cone is needed . The bounded subset $\mathrm{cone}_{\mathrm{T}}(\Gamma) = \{\alpha \mathrm{x} : \mathrm{x} \in \Gamma,\ \alpha \in [0,1]\}$ will be called the truncation of the complete conical hull $\mathrm{cone}(\Gamma) = \{\alpha \mathrm{x} : \mathrm{x} \in \Gamma,\ \alpha \in [0,\infty]\}$. Then the corresponding truncated cone $\mathbb{K}^\eta$ is defined by

$$\begin{aligned}\mathbb{K}^\eta = \quad & \big\{\mathrm{cone}_{\mathrm{T}}(\eta_1 \mathrm{W}^{\mathrm{T}} \bar{\mathrm{M}}_1 \Omega) \times \mathrm{cone}_{\mathrm{T}}(\eta_2 \mathrm{W}^{\mathrm{T}} \bar{\mathrm{M}}_2 \Omega) \times \\ & \ldots \times \mathrm{cone}_{\mathrm{T}}(\eta_{\mathrm{N}} \mathrm{W}^{\mathrm{T}} \bar{\mathrm{M}}_{\mathrm{N}} \Omega)\big\}\end{aligned} \tag{27}$$

It turns out that it is easy to compute a positive real η such that the unknown optimal solution of (22) lies in the bounded set $\mathbb{K}^\eta$. This yields the optimization problem

$$\begin{array}{ll}\min & \mathbf{v}(\bar{z}) \\ \bar{z} \in \mathbb{K}^\eta & \end{array} \tag{28}$$

An algorithm for solving this problem is discussed next.

3.1 Finding Supporting Hyperplanes

The concepts and algorithm of (J.C.Allwright, 1988) can now be applied to solve the optimization problem of (28). At each iteration, it first calculating the gradient $\bar{g}_k = 2\bar{z}_k^T \bar{Q} - 2\bar{\beta}^T(\bar{P}_1^T \bar{P}_1)$ of the quadratic cost $\mathbf{v}(\bar{z})$. A linearisation about $\bar{z}_k$ of $\mathbf{v}(\bar{z})$ is then provided by $\mathbf{v}(\bar{z}_k) + \bar{g}_k^T(\bar{z} - \bar{z}_k)$. The next step is to minimize this linearisation with respect to $\bar{z}$ from $\mathbb{K}^\eta$ by solving the problem:

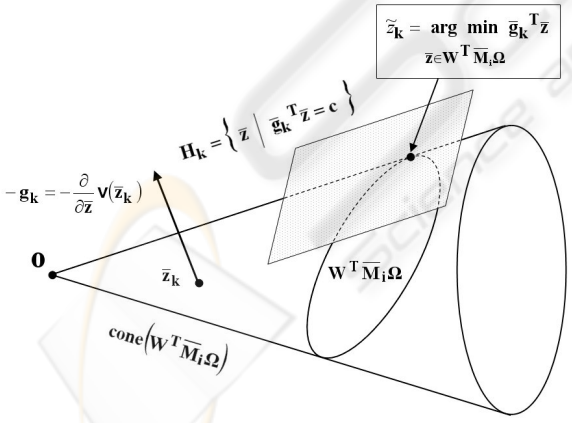

Figure 2: Linear minimization over cone.

$$\begin{array}{ll}\min & \bar{g}_k^T(\bar{z} - \bar{z}_k) \\ \bar{z} \in \mathbb{K}^\eta & \end{array} \tag{29}$$

For suppose an optimal solution for this problem is called $\tilde{z}_k$. Then $\tilde{z}_k - \bar{z}_k$ is a feasible descent direction from $\bar{z}_k$ and therefore can be used to decrease the value of the cost fuction $\mathbf{v}(\bar{z})$. It can be seen that the minimizer will be either in the set $\eta W^T \bar{M}_i \Omega$ or at the origin i.e. at the vertex of the cone. An interpretation of this optimization is that it is finding the supporting hyperplane for the set $\mathbb{K}^\eta$ that has normal $\bar{g}_k$. Owing to the structure of optimization problem (29), this linear programming problem can be decomposed into N separate minimization problems

$$\begin{array}{ll}\min & \bar{g}_{i_k}^T \bar{z} \\ \bar{z} \in \mathbb{K}_i^\eta & \\ i \in [1:N] & \end{array} \tag{30}$$

where the gradients $\bar{g}_{i_k}$ are from the partitioning of the gradient vector $\bar{g}_k$ as $\begin{bmatrix}\bar{g}_{1_k} & \bar{g}_{2_k} & \ldots & \bar{g}_{N_k}\end{bmatrix}^T$. According to (18-21), minimizing $\bar{g}^T(\bar{z})$ over the set $\mathbb{K}_i^\eta = \mathrm{cone}_{\mathrm{T}}(\eta_{\mathrm{i}} \mathrm{W}^{\mathrm{T}} \bar{\mathrm{M}}_{\mathrm{i}} \Omega)$ will give the following minimizer:

$$\tilde{z}_{i_k} = \eta_i W^T \mathrm{vec}(\nu_{i_k} \nu_{i_k}^T) \tag{31}$$

where η_i is truncation factor for appropriate set and ν_{i_k} is a normalized eigenvector corresponding to the minimum eigenvalue of Z_{i_k}, $\lambda_{min}[Z_{i_k}]$, which is defined as:

$$Z_{i_k} = \frac{(\mathrm{vec}^{-1}(\bar{\mathbf{g}}_{i_k})) + (\mathrm{vec}^{-1}(\bar{\mathbf{g}}_{i_k})^T)}{2} \in R^{n \times n} \tag{32}$$

with,

$$\bar{\mathbf{g}}_{i_k} = W[\bar{g}_{i_k}^T W^T \bar{M}_i W]^T \tag{33}$$

Hence the contact point between set $\mathbb{K}^\eta$ and supporting hyperplane $\mathbf{H}_k = \{\bar{z} \,|\bar{g}_k^T \bar{z} = C\}$ is the vector $\tilde{z}_k = \begin{bmatrix}\tilde{z}_{1_k}^T & \tilde{z}_{2_k}^T & \ldots \tilde{z}_{N_k}^T\end{bmatrix}^T$.

3.2 Quadratic Minimization Over a Polytope

One way to determine a point $\bar{z}_{k+1}$ giving $\mathbf{v}(\bar{z}_{k+1}) < \mathbf{v}(\bar{z}_k)$ would be to minimize $\mathbf{v}(\bar{z})$ on the set $\{\bar{z}_k + \omega(\tilde{z}_k - \bar{z}_k) : \omega \in [0,1]\}$. This involves optimization with respect to the scalar ω and is therefore relatively easy to carry out. This set is a subset of $\mathbb{K}^\eta$ so optimization on it can be regarded as optimization of $\mathbf{v}(\bar{z})$ on an approximation to $\mathbb{K}^\eta$. Obviously using a larger approximating set would tend to give a greater reduction in $\mathbf{v}(\bar{z})$. Such a set can be obtained as the convex hull of $\{0, \tilde{z}_k, \bar{z}_{k-1}, \bar{z}_k, \ldots\}$. Then $\bar{z}_{k+1}$ is chosen as the point in this convex hull that minimizes $\mathbf{v}(\bar{z})$. A way to do this optimization on the polytope is outlined next.

Having the contact points available, an approximation to the set $\mathbb{K}^\eta$ is given by the convex hull of the contact points (29) and the optimal points of (28) from previous iterations. Then the minimizer of (28), $\hat{\tilde{z}}$, can be calculated by solving a quadratic problem

over this convex hull, as follows:

Theorem 3.1 : Let $\hat{\bar{z}}_{k-N_v}, \hat{\bar{z}}_{k-N_v+1}, \ldots, \hat{\bar{z}}_{k-1}$ be the set of minimizers of the quadratic function $\mathbf{v}(\bar{z})$ over the convex hull defined at iterations $(k-N_v), (k-N_v+1), \ldots, (k-1)$, and let $\tilde{z}_{k-1}$ be the minimizer of $\bar{g}_{k-1}^T(\bar{z}-\bar{z}_{k-1})$ over $\mathbb{K}^\eta$ at iteration $(k-1)$. Then

$$\mathrm{v}(\hat{\bar{z}}_k) \leq \mathrm{v}(\tilde{z}_{k-1}) \leq \mathbf{v}(\hat{\bar{z}}_{k-1}) \tag{34}$$

where,

$$\hat{\bar{z}}_k = \underset{\bar{z}\in \mathrm{co}\left\{\hat{\bar{z}}_{k-N_v}, \hat{\bar{z}}_{k-N_v+1}, \ldots, \hat{\bar{z}}_{k-1}, \tilde{z}_{k-1}\right\}}{\arg\min\ \mathrm{v}(\bar{z})} \tag{35}$$

Here $\mathrm{co}\left\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_{N_v}\right\}$ represents the convex hull of the vectors $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_{N_v}$ and N_v denotes the number of vectors used for the convex hull. The integer k denotes the iteration number.

To solve optimization problem (35), we first represent the variables $\bar{z}_i$ of each set $\mathbb{K}_i$ as a convex combination of $\bar{v}_1^i, \bar{v}_2^i, \ldots, \bar{v}_{N_v}^i$, i.e.

$$\bar{z}_i = \textstyle\sum_{j=1}^{N_v} \mu_j^i \bar{v}_j^i \quad \text{for } \sum_{\mathrm{j=1}}^{\mathrm{N_v}} \mu_{\mathrm{j}}^{\mathrm{i}} = 1,\ \mu_{\mathrm{j}}^{\mathrm{i}} \geq 0 \tag{36}$$

for all $i \in [1:N]$ with each $v_j^i \in R^{(r\times 1)}$.

Using this in (35) transforms the problem into the quadratic programme in $\tilde{\mu}$

$$\begin{array}{ll} \min\limits_{\tilde{\mu}\in R^{(N_v\times N)\times 1}} & \tilde{\mu}^T\mathbf{V}^T\bar{Q}\mathbf{V}\tilde{\mu} + \bar{F}\mathbf{V}\tilde{\mu} \\ \text{subject to} & \mathbf{G}_c\tilde{\mu} = \left[\bar{1}_{((N_v\times N)\times 1)}\right] \\ & \tilde{\mu} \geq 0 \end{array} \tag{37}$$

where

$$\mathbf{V} = \begin{bmatrix} \bar{V}_1 & \bar{0}_{(r\times N_v)} & \cdots & \bar{0}_{(r\times N_v)} \\ \bar{0}_{(r\times N_v)} & \bar{V}_2 & \cdots & \bar{0}_{(r\times N_v)} \\ \vdots & & & \vdots \\ \bar{0}_{(r\times N_v)} & \bar{0}_{(r\times N_v)} & \cdots & \bar{V}_N \end{bmatrix} \tag{38}$$

$$\mathbf{G}_c = \begin{bmatrix} \bar{1}_{(1\times N_v)_1} & \bar{0}_{(1\times N_v)} & \cdots & \bar{0}_{(1\times N_v)} \\ \bar{0}_{(1\times N_v)} & \bar{1}_{(1\times N_v)_2} & \cdots & \bar{0}_{(1\times N_v)} \\ \vdots & & & \vdots \\ \bar{0}_{(1\times N_v)} & \cdots & & \bar{1}_{(1\times N_v)_N} \end{bmatrix} \tag{39}$$

with

$$\begin{array}{l} \bar{V}_i = \begin{bmatrix} \bar{v}_1^i & \bar{v}_2^i & \ldots & \bar{v}_{N_v}^i \end{bmatrix} \\ \tilde{\mu} = [\bar{\mu}^{1^T}\ \bar{\mu}^{2^T}\ \ldots\ \bar{\mu}^{N^T}]^T \\ \bar{\mu}^i = [\mu_1^i\ \mu_2^i\ \ldots\ \mu_{N_v}^i]^T;\ i = 1, \ldots, N. \end{array} \tag{40}$$

Here $\bar{0}_{(p\times q)}$ is a $p\times q$ zero matrix and $\bar{1}_{p\times q}$ is a $p\times q$ matrix with all its entries equal to 1.

The optimal solution, $\hat{\bar{z}}$, of the original problem (35) can then be calculated from the solution $\hat{\tilde{\mu}}$ of the transformed problem (37) using

$$\hat{\bar{z}} = [\mathbf{V}_f]\,\hat{\tilde{\mu}} \tag{41}$$

where $\mathbf{V}_f$ is as defined in (38) with its entries formed from the vectors specifying the convex hull defined in the last iteration.

The convergence of the sequence $\bar{z}_k$ to an optimizer for optimization problem (22) can be proved in much the same way that convergence for a similar optimization problem was proved in (J.C.Allwright, 1988).

3.3 Correction Using Gradient

It may happen that the solution $\hat{P} = \mathrm{vec}^{-1}(W\hat{\bar{z}}_i)$ from (41) above lies on one or more of the boundaries of the cones $\mathbb{K}_i^\eta$, in which case the solution will not satisfy at least one of the positive-definiteness constraints. Even if the all the positive-definiteness constraints are satisfied, it might be desired to have a matrix that is not so close to the boundary in that the smallest eigenvalue of $-(A_i^TP+PA_i)$ is undesirably small. In this case, a correction using the gradients of the constraints can be incorporated into the algorithm. This enables a small update δP to be applied to the solution $\hat{P}$ to give the modified solution $\hat{P}+\delta P$ where the correction can be obtained by finding a $\delta\tilde{P}$ that makes the following inequalities hold for all j:

$$\begin{array}{ll} \sum_{l,m\in[1:n]} \frac{\partial\lambda_j(\tilde{M}(i))}{\partial P_{l,m}}\delta\tilde{P}_{l,m} \leq 0, & i \in I_- \\ \sum_{l,m\in[1:n]} \frac{\partial\lambda_j(\tilde{M}(i))}{\partial P_{l,m}}\delta\tilde{P}_{l,m} < 0, & i \in I_0 \\ \sum_{l,m\in[1:n]} \frac{\partial\lambda_j(\tilde{M}(i))}{\partial P_{l,m}}\delta\tilde{P}_{l,m} < 0, & i \in I_+ \end{array} \tag{42}$$

Here (l,m) is the index of the elements of $\hat{P} = \overline{\mathrm{vec}}^{-1}(\hat{\bar{z}})$, and $\tilde{M}(i) = A_i^TP + PA_i$. I_-, I_0, and I_+ denote the index of sets where the initial point $\hat{P}$ lies inside the corresponding set, on its boundary or outside, respectively. It is obvious that for $i \in I_-$, $i \in I_0$, and $i \in I_+$, the value of $\tilde{M}(i)$ will be negative, zero, and positive respectively.
The gradient can obtained from:

$$\frac{\partial\lambda_j(\tilde{M}(i))}{\partial P_{l,m}} = x_jx_j^T \tag{43}$$

with x_j is the eigenvector corresponding to the eigenvalue $\lambda_j(\tilde{M}(i))$. The $\delta\tilde{P}$ will make the inequalities in (42) hold, and it can be used for updating δP i.e. we can set $\delta P = \omega\delta\tilde{P}$ by choosing an appropriate scalar $\omega > 0$, so that all constraints are satisfied by the corrected solution.

4 NUMERICAL CALCULATION

4.1 The Algorithm

The algorithm then is implemented in numerical calculation for solving our convex problem. The algorithm is arranged as follows:

1. set $k = 0$, choose initial point $\bar{z}_0$, error bound $\epsilon > 0$, number of vertices of the convex hull N_v, and calculate basis matrix W and $\bar{M}_i$ for all i
2. set $\bar{z} = \bar{z}_k$
3. calculate cost value $\mathbf{v}(\bar{z}_k)$ and the gradient $\bar{g}_k$, then find the the contact point $\tilde{\bar{z}}_k$ between set $\mathbb{K}^\eta$ and hyperplane $\mathbf{H}_k$ by solving $\bar{g}_k^T \bar{z}$ (29)
4. for $\mathrm{co}\{\bar{z}_k, \bar{z}_{k-1}, \ldots, \bar{z}_{k-N_v+1}, \tilde{\bar{z}}_k\}$, if $k < (N_v - 1)$ then set $\bar{z}_{k-1}, \ldots, \bar{z}_{k-N_v+1} = 0$, form $\mathbf{V}$ and $\mathbf{G}_c$, and calculate $\mathbf{V}^T \bar{Q} \mathbf{V}$ and $\bar{F}\mathbf{V}$
5. solve the quadratic programming problem (37) to get the minimizer $\hat{\bar{z}}_k$, and calculate $\mathbf{v}(\hat{\bar{z}}_k)$ and the lower bound $b_{low} = \mathbf{v}(\bar{z}_k) + \bar{g}_k^T(\hat{\bar{z}}_k - \bar{z}_k)$
6. if $v(\hat{\bar{z}}_k) - b_{low} < \epsilon$ then stop, otherwise set $\bar{z}_{k+1} = \hat{\bar{z}}_k$ and $k = k + 1$ and go to step 2
7. perform correction procedure of section 3.3 if necessary

4.2 Numerical Results

The algorithm has been tested for finding a common Lyapunov function for a various problems. The results were compared with those of the projective methods (A.Nemirovskii, 1994) available in the MATLAB LMI toolbox. A common symmetric PD matrix P was found after finite iterations using the conical hulls, and the computing times for it are listed in 2nd row of table 1. The 3rd row shows the processing time for the correction procedure. The conical hull algorithm and the MATLAB algorithm were executed 100 times to observe the consistency of the time calculations. Since the latest version of MATLAB exploits the JIT-accelerator, converting the M-script to C/C++ code (MEX-file) would not have improved the execution time.

Table 1: Numerical Results - Processing time (seconds).

$n = 10$	$N = 2$	$N = 4$	$N = 6$	$N = 10$
$C - Hull$	0.003	0.0502	0.11	0.384
$(corrct)$	0.023	0.0411	0.0576	0.06
$ProjMet$	0.0633	0.0805	0.1063	0.2484

The results show that the conical hull algorithm is faster for $N = 2, 4$. As the number of inequalities increases, the conical hull algorithm becomes slower, compared to the projective method. Since the proposed method uses distance minimization between points in all the cones, as the number of inequalities rises, the number of variables will also rise significantly. Probably, the reason the conical hull algorithm performs less well for bigger problem is that the QP involved at each iteration becomes relatively more costly to execute. To overcome this problem, an efficient way for managing the variables should be explored. It should be noticed that the proposed algorithm guarantees all the inequalities are satisfied accurately.

5 CONCLUSIONS

A new algorithm has been presented, based on the existing conical hull theory, for solving the problem of finding a common quadratic Lyapunov function for a family of stable dynamical systems. The numerical results suggest that the algorithm is better than the projective method for small problems. Further developments will be carried out to improve it for larger problems and apply it to control studies.

ACKNOWLEDGEMENTS

The first author would like to thank the Islamic Development Bank (IDB) who provides the fund that enables him to undergo the project.

REFERENCES

A.Nemirovskii (1994). The projective method for solving linear matrix inequalities. *Proceedings of the American Control Conference*, pages 840–844.

D.Liberzon (2003). Gradient algorithm for finding common lyapunov functions. *42nd IEEE Conference on Decision and Control*, pages 4782–4787.

G.Strang (1988). *Linear algebra and its applications*. Harcourt Brace Jovanovich, San Diego, 3 edition.

G.Xie (2004). Stability and stabilization of switched linear systems with state delay: continuous-time case. *16th Mathematical Theory of Networks and Systems Conference*.

J.C.Allwright (1988). Positive semidefinite matrices : Characterization via conical hulls and least-squares solution of a matrix equation. *SIAM journal of Control and Optimization*, 26:537–556.

J.C.Allwright (1989). On maximizing the minimum eigenvalue of a linear combination of symmetric matrices. *SIAM journal on Matrix Analysis and Applications*, 10:347–382.

N.A.Bobylev, e. (2002). On the stability of families of dynamical systems. *Differential Equations*, 38(4):464–470.

R.A.Horn (1985). *Matrix Analysis*. Cambridge University Press.

R.N.Shorten (2003). On common quadratic lyapunov functions for pairs of stable lti systems whose system matrices are in companion form. *IEEE Transactions on Automatic Control*, 48(4):618–621.