Not a Part of ROB501

EECS 598-005

Hybrid Systems: Specification, Verification and Control

Fall 2015

Lecture Notes 10-12-13-14

Necmiye Ozay, 10/12/2015-10/21/2015-10/26/2015-10/28/2015

These notes are mostly based on sections 3.1-3.2 of [1]. The discussion on SDPs and SoS is based on [2].

1 Introduction to optimization

Optimization problems are quite common in all branches of engineering. Many problems in stability analysis, verification or control synthesis of continuous or hybrid systems can be recast as an optimization or feasibility problem. Convex optimization problems constitute a class of optimization problems which can be solved globally (i.e., with guarantees to find the best possible solution) and efficiently. These notes aim to provide basic working knowledge of convex optimization and relevant tools (in particular, linear programming, semidefinite programing and sum-of-squares programming).

1.1 Convex functions

Remark 1. *Some representative proofs are provided to show how convexity of a function can be proved by using the basic definitions. Similar ideas can be used for other examples.*

Definition 1. *A function* $f : \mathbb{R}^n \to \mathbb{R}$ *is called* affine *if* $f(\theta x + (1-\theta)y) = \theta f(x) + (1-\theta)f(y)$ *for all* $\theta \in [0,1]$ *and* $x, y \in \mathbb{R}^n$ *(i.e.,* f *is equal to a linear function plus a constant,* $f(x) = a^T x + b$ *for some* $a \in \mathbb{R}^n$ *and* $b \in \mathbb{R}$*).*

Definition 2. *A function* $f : \mathbb{R}^n \to \mathbb{R}$ *is called* convex *if* $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$ *for all* $\theta \in [0,1]$ *and* $x, y \in \mathbb{R}^n$*. We say* f *is* concave *if* $-f$ *is convex.*

Example 1. *Some examples of convex functions:*

- *Affine functions are both convex and concave.*
- e^{ax} *is a convex function on* $\mathbb{R}$ *for any* $a \in \mathbb{R}$.
- $|x|^p$*, for* $p \geq 1$ *is a convex function on* $\mathbb{R}$.

- *p-norms (i.e.,* $||x||_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$*, for* $p \geq 1$ *or* $||x||_\infty = \max_i |x_i|$*) are convex functions on* $\mathbb{R}^n$*.*

- $f(X) = tr(A^T X) + b$ *is a convex function on* $\mathbb{R}^{n \times n}$*, where* tr *stands for trace. Indeed,* f *is an affine function.*

- *Spectral norm* $f(X) = ||X||_2 = \sigma_{max}(X) = (\lambda_{max}(X^T X))^{1/2}$ *is a convex function on* $\mathbb{R}^{n \times n}$*, where* σ_{max} *is the maximum singular value and* λ_{max} *is the maximum eigenvalue.*

1.1.1 Properties of convex functions

- If f_1 and f_2 are convex, then $f(x) = f_1(x) + f_2(x)$ is convex (proved in class).

- If f_1 and f_2 are convex, then $f(x) = \max\{f_1(x) + f_2(x)\}$ is convex.

- $g(x) = \sup_{y \in S \subseteq \mathbb{R}^m} f(x, y)$ is convex if $f(x, y)$ is convex in x for each fixed $y \in S$ (this is a generalization of the property in the previous bullet from two functions to arbitrary number of functions parametrized by y).

- Convex functions on $\mathbb{R}^n$ are continuous.

- If $f : \mathbb{R}^n \to \mathbb{R}$ is convex and $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, then $g : \mathbb{R}^m \to \mathbb{R}$ such that $g(x) = f(Ax + b)$ is convex.

  Proof: Need to show $g(\theta x + (1-\theta)y) \leq \theta g(x) + (1-\theta)g(y)$ for all $\theta \in [0, 1]$ and $x, y \in \mathbb{R}^m$. By definition:

$$\begin{aligned}
g(\theta x + (1-\theta)y) &= f(A(\theta x + (1-\theta)y) + b) \\
&= f(\theta Ax + (1-\theta)Ay + \theta b + (1-\theta)b) \\
&= f(\theta(Ax + b) + (1-\theta)(Ay + b)) \\
&\leq \theta f(Ax + b) + (1-\theta)f(Ay + b) \\
&= \theta g(x) + (1-\theta)g(y).
\end{aligned} \tag{1}$$

  Since θ, x and y are arbitrary, g is convex.

- If $f : \mathbb{R}^n \to \mathbb{R}$ is convex and $a, b \in \mathbb{R}$, $a \geq 0$, $g(x) = af(x) + b$ is convex.

- If $f : \mathbb{R}^n \to \mathbb{R}$ is convex, then α-*sublevel set* $S_{f,\alpha} = \{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}$ of f is a convex set for any $\alpha \in \mathbb{R}$ (the converse is not true, there are functions for which α-sublevel sets for all α are convex but the function is not convex; e.g., non-convex monotonically increasing functions).

Some of the properties listed above can be used to identify whether a function is convex by looking into its building blocks.

1.1.2 First and second order conditions for convexity

Here are a few other methods for identifying if a function is convex or not if it is known to satisfy certain differentiability conditions.

- Let f be a differentiable function, with the gradient

  $$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)^T .$$

  Then f is convex if and only if

  $$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{for all } x, y \in \mathbb{R}^n. \tag{2}$$

  Note that the right hand side in Eq. (2) is the first order Taylor series approximation of $f(y)$ for a fixed x. Therefore, when f is a convex function, the first order Taylor series approximation is a global under-estimator (under-approximation) of $f(y)$.

- Let f be a twice differentiable function, with the Hessian

  $$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial^2 x_1} & \frac{\partial f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial f(x)}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial f(x)}{\partial^2 x_n} \end{bmatrix} .$$

  Then f is convex if and only if the Hessian is positive semidefinite ($\nabla^2 f(x) \succeq 0$) for all $x \in \mathbb{R}^n$.

Example 2. *Consider the least squares objective function $f(x) = ||Ax - b||_2^2$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. We have $\nabla f(x) = 2A^T(Ax - b)$ and $\nabla^2 f(x) = 2A^T A$. Since $2A^T A \succeq 0$, f is convex.*

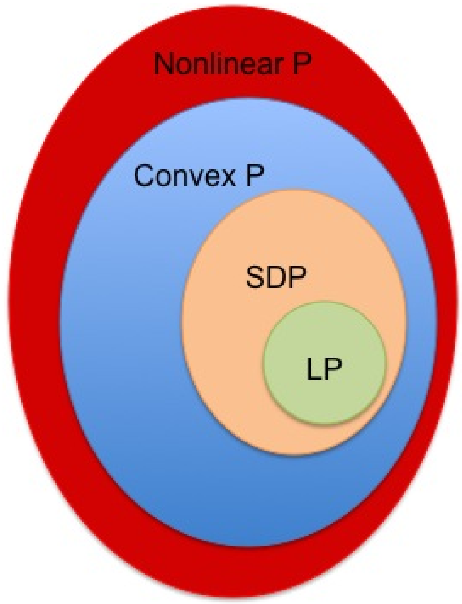

Figure 1: Optimization problems considered in this note can be categorized as in this picture, where nonlinear programming problems constitute the largest class. Convex programming problems are a subset of nonlinear programming problems. Semidefinite programming (SDP) problems are a subset of convex programs. Finally linear programming (LP) problems are a subset of SDPs.

1.2 Mathematical optimization

This section focuses on optimization problems in $\mathbb{R}^n$ [1]. Figure 1 shows a hierarchy of optimization problems that will be considered in these notes.

A generic *nonlinear programming problem* in $\mathbb{R}^n$ has the following form:

$$\begin{array}{ll} \text{minimize}_x & f_0(x) \\ \text{subject to} & f_i(x) \leq 0 \quad \text{for all } i = 1, \ldots, m \\ & h_i(x) = 0 \quad \text{for all } i = 1, \ldots, p \end{array} \tag{3}$$

where each f_i or h_i is a function from $\mathbb{R}^n$ to $\mathbb{R}$. In problem (3), f_0 is called the *objective function* or cost function, $x = (x_1, \ldots, x_n)^T$ are the optimization (or decision) variables. The set $F = \{x \mid f_i(x) \leq 0 \text{ for all } i = 1, \ldots, m;\ h_i(x) = 0 \text{ for all } i = 1, \ldots, p\}$ is

[1] Note that all finite dimensional vector spaces are isomorphic to $\mathbb{R}^n$ for some n. Therefore, optimization problems that involve matrix variables or finitely parametrized functions with vector space structure can be seen in this category.

called the *feasible set*, where $f_i(x) \leq 0$ for $i = 1, \ldots, m$ are inequality constraints and $h_i(x) = 0$ for $i = 1, \ldots, p$ are equality constraints. Equality constraints can be omitted without loss of generality. If $F = \emptyset$, the problem (3) is said to be *infeasible*. Any $x \in F$ is *feasible* for the problem (3). A point $x^* \in F$ such that $f_0(x^*) \leq f_0(y)$ for all $y \in F$ is said to be a *globally optimal solution* of the problem (3), and $f_0(x^*)$ is the *optimal value*. Note that the optimal value can be unbounded. If $f_0(x^*) \leq f_0(y)$ for all y in some local neighborhood of x^*, then x^* is a *locally optimal solution*.

Optimization problems are closely related to feasibility problems, that is to check whether a feasible set F is empty or not. A feasibility problem is typically written as follows:

$$\begin{array}{rll} \text{find} & x & \\ \text{such that} & f_i(x) \leq 0 & \text{for all } i = 1, \ldots, m \\ & h_i(x) = 0 & \text{for all } i = 1, \ldots, p. \end{array} \tag{4}$$

Definition 3. *A problem of the form* (3) *is called a* convex programming problem *if f_i for $i = 0, \ldots, m$ (i.e., objective function and the inequality constraint functions) are convex functions and h_i for $i = 1, \ldots, p$ (inequality constraint functions) are affine functions.*

The feasible set of a convex programming problem is a convex set. Sometimes convex programming problems are defined as optimization problems where the objective function and feasible set are both convex. This alternative definition is more general than the one given in Def. 3. Nevertheless, convex programming problems share the following important property which makes it easier to solve them globally:

- Any locally optimal solution of a convex programming problem is a globally optimal solution.

Next a special class of convex programming problems, namely linear programming problems, is defined.

1.2.1 Linear programming

Definition 4. *A problem of the form* (3) *is called a* linear programming (LP) problem *if f_i for $i = 0, \ldots, m$ and h_i for $i = 1, \ldots, p$ are affine functions.*

An LP is typically written in the following form:

$$\begin{array}{rl} \text{minimize}_x & c_0^T x \\ \text{subject to} & Ax \leq b \\ & Cx = d \end{array} \tag{5}$$

where $c_0 \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $C \in \mathbb{R}^{p\times n}$ and $b \in \mathbb{R}^p$. In problem (5), the objective function is a linear function (the constant term in an affine function does not affect the optimal solutions, so can be dropped). Also, the constraints are written in matrix form. Note that the feasible set of an LP is a polyhedron.

There are many efficient algorithms to solve LPs. Also, many interesting problems can be converted to an LP. Some examples are provided next.

Example 3. *Given a point $x \in \mathbb{R}^n$, and a set of points $v_1, \ldots, v_k \in \mathbb{R}^n$, whether $x \in \mathrm{conv}\{v_1, \ldots, v_k\}$ can be checked via the following linear feasibility problem:*

$$\begin{array}{rl} \textit{find} & \alpha \\ \textit{such that} & -\mathbf{I}\alpha \le \mathbf{0} \\ & [v1, \ldots, v_k]\alpha = x \\ & [1, \ldots, 1]\alpha = 1 \end{array} \tag{6}$$

where the decision variables $\alpha \in \mathbb{R}^k$, **I** *and* **0** *are the identity matrix and zero vector of size k.*

1.2.2 Semidefinite programming

A matrix $M \in \mathbb{R}^{n\times n}$ is called *symmetric* if $M = M^T$. the set of $n \times n$ symmetric real matrices are denoted by $\mathcal{S}^n$.

Proposition 1. *The following statements are equivalent:*

- *The matrix $M \in \mathcal{S}^n$ is* positive semidefinite *($M \succeq 0$).*
- *For all $x \in \mathbb{R}^n$, $x^T M x \ge 0$.*
- *All eigenvalues of M are nonnegative.*
- *All $2^n - 1$ principal minors*[2] *of M are nonnegative.*
- *There exists a factorization $M = B^T B$.*

Proposition 2. *The following statements are equivalent:*

- *The matrix $M \in \mathcal{S}^n$ is* positive definite *($M \succ 0$).*

[2] A principal submatrix of an $n \times n$ matrix M is a matrix formed by deleting some number of rows and the corresponding columns of M. The determinant of a principal submatrix is called a principal minor. The leading principal minor is the determinant of the leading principal submatrix obtained by deleting the last $n - k$ rows and columns of an $n \times n$ matrix.

- *For all* $x \in \mathbb{R}^n$, $x^T M x > 0$ *for all* $x \neq 0$.
- *All eigenvalues of* M *are positive.*
- *All* n *leading principal minors of* M *are positive.*
- *There exists a factorization* $M = B^T B$, *with* B *square and nonsingular.*

A matrix M is called negative (semi)definite if $-M$ is positive (semi)definite. We write $M_1 \succeq M_2$ if $M_1 - M_2 \succeq 0$. An $n \times n$ real matrix M is called *antisymmetric* or *skew symmetric* if $M = -M^T$.

Definition 5. *A* linear matrix inequality (LMI) *in variables* $x_i \in \mathbb{R}$ *for* $i = 1, \ldots, k$ *is of the form:*

$$F_0 + x_1 F_1 + \ldots x_k F_k \succeq 0,$$

where $F_i \in \mathcal{S}^n$ *for* $i = 0, \ldots, k$.

Here are some additional useful facts:

- If T is nonsingular, $M \succ 0$ if and only if $T^T M T \succ 0$.
- If $M \in \mathbb{R}^{n \times n}$ is skew symmetric, $x^T M x = 0$ for all $x \in \mathbb{R}^n$.
- (Schur complement formula) The following conditions are equivalent:
  - $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succ 0$
  - $\begin{cases} A \succ 0 \\ C - BA^{-1}B^T \succ 0 \end{cases}$
  - $\begin{cases} C \succ 0 \\ A - BC^{-1}B^T \succ 0 \end{cases}$
- (S-procedure) Given $F_0, \ldots F_k \in \mathcal{S}^n$, we have $x^T F_0 x \geq 0$ for all x such that $x^T F_i x \geq 0$ for $i = 1, \ldots, k$ if there exists non-negative multipliers $0 \leq \lambda_i \in \mathbb{R}$ $i = 1, \ldots, k$ such that $x^T F_0 x \geq \sum_{i=1}^k \lambda_i x^T F_i x$ for all $x \in \mathbb{R}^n$. In other words, if $F_0 \succeq \sum_{i=1}^k \lambda_i F_i$, then

  $$\{x \in \mathbb{R}^n \mid x^T F_0 x \geq 0\} \supseteq \{x \in \mathbb{R}^n \mid x^T F_i x \geq 0 \text{ for } i = 1, \ldots, k\}.$$

Note that the condition $F_0 \succeq \sum_{i=1}^{k} \lambda_i F_i$ is an LMI in λ_i's. Also, the condition is necessary and sufficient in the case of a single quadratic form (i.e., $k = 1$) provided that there exists an $x \in \mathbb{R}^n$ such that $x^T F_1 x > 0$ (the proof of necessity is not easy but can be found in [3]).

A semidefinite program (SDP) can be written in the following form:

$$\begin{array}{rl} \text{minimize}_x & c_0^T x \\ \text{subject to} & \sum_{i=1}^{m} x_i A_i \succeq B \\ & Cx = d, \end{array} \tag{7}$$

where the constraints are linear matrix inequalities[3] and linear equality constraints. The following optimization problem with matrix variables is also an SDP:

$$\begin{array}{rl} \text{minimize}_{X \in \mathcal{S}^n} & \langle C_0, X \rangle \\ \text{subject to} & \langle C_i, X \rangle = d_i, \quad i = 1, \ldots, m \\ & X \succeq 0, \end{array} \tag{8}$$

where $\langle A, B \rangle \doteq tr(A^T B)$ is the inner product on $\mathcal{S}^n$. Note that it is possible to rewrite (8) in the form of (7).

1.2.3 Sum-of-squares (SoS) programming

This section describes how certain functional optimization/feasibility problems (problems where unknown variables are functions) can be converted to semidefinite programming problems. Such problems are relevant for instance while searching for Lyapunov functions or barrier certificates for stability and safety verification respectively.

Definition 6. *Given*[4] $\boldsymbol{\alpha} \in \mathbb{N}^n$, *a* monomial *in n variables is a function $m_{\boldsymbol{\alpha}} : \mathbb{R}^n \to \mathbb{R}$ defined as*

$$m_{\boldsymbol{\alpha}}(x) = x^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdot \ldots \cdot x_n^{\alpha_n}.$$

The degree of a monomial is defined as $\deg m_{\boldsymbol{\alpha}} \doteq \sum_{i=1}^{n} \alpha_i$.

Definition 7. *A* polynomial *in n variables (aka, multivariate polynomial) is a function $p : \mathbb{R}^n \to \mathbb{R}$ defined as a linear combination of monomials. That is*

$$p(x) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} x^{\boldsymbol{\alpha}},$$

[3] without loss of generality one can assume a single LMI, since multiple LMIs can be concatenated diagonally to generate single LMI.

[4] $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]$ is a non-negative integer vector of size n

where $c_{\boldsymbol{\alpha}} \in \mathbb{R}$ for all $\boldsymbol{\alpha} \in \mathcal{A}$, and $\mathcal{A} \subset \mathbb{N}^n$ is a finite set. The set of polynomials in n variables is denoted as $\mathbb{R}[x_1, \ldots, x_n]$ (or, $\mathbb{R}[x]$ when the dimension of x is clear from context). The degree of a polynomial $p(x) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} c_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}$ is defined as $\deg\, p \doteq \max_{\boldsymbol{\alpha}\in\mathcal{A}, c_{\boldsymbol{\alpha}}\neq 0}(\deg\, m_{\boldsymbol{\alpha}})$.

Definition 8. *A polynomial p is called* sum-of-squares (SoS) *polynomial if there exists polynomials p_i for $i = 1, \ldots, k$ such that*

$$p(x) = \sum_{i=1}^{k} p_i^2(x).$$

Obviously, SoS polynomials are non-negative (i.e., $p(x) \geq 0$ for all $x \in \mathbb{R}^n$). But there are non-negative polynomials that are not SoS.

Given a polynomial p in n variables with degree less than d, it can be written as $p(x) = \mathbf{c}^T z_{n,d}$, where $\mathbf{c} \in \mathbb{R}^{\binom{n+d}{d}}$ is the vector of coefficients and $z_{n,d}$ is a vector that contains all monomials in n variables up to degree d ($z_{n,d}$ is also called a monomial basis).

Example 4. *Consider the polynomial $p(x) = x_1^2 - x_2^2$. We have $n = 2$, $d = 2$ and a monomial basis $z_{2,2} = [1, x_1, x_2, x_1^2, x_1x_2, x_2^2]^T$. The polynomial $p(x)$ can be written in this basis uniquely as follows*

$$p(x) = x_1^2 - x_2^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}.$$

Given a polynomial p in n variables with degree less than $2d$, it can be written as $p(x) = z_{n,d}^T Q z_{n,d}$, where $Q \in \mathcal{S}^{\binom{n+d}{d}}$ is a symmetric matrix whose entries depend affinely on the coefficients of $p(x)$ and $z_{n,d}$ is a monomial basis as defined above.

Example 5. *Consider the polynomial $p(x) = x_1^2 - x_2^2$. We have $n = 2$, $2d = 2$ and a monomial basis $z_{2,1} = [1, x_1, x_2]^T$. The polynomial $p(x)$ can be written as*

$$p(x) = x_1^2 - x_2^2 = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}.$$

Proposition 3. *A polynomial p is sum of squares if and only if $\exists Q \succeq 0$ s.t.*

$$p = z^T Q z,$$

where z is a vector of monomials.

Proof. ($\Longrightarrow$) Assume p is SoS in n variables with degree less than $2d$. By definition of SoS polynomials, there exist p_i, $i = 1, \dots, k$ for some k, each with degree less than d, such that $p(x) = \sum_{i=1}^k p_i^2(x)$. Let $z_{n,d}$ be a monomial basis. Then, each p_i can be written as $p_i(x) = \mathbf{c}_i^T z_{n,d}$. Therefore,

$$\begin{aligned} p(x) &= \sum_{i=1}^k p_i^2(x) = \sum_{i=1}^k (\mathbf{c}_i^T z_{n,d})^2 \\ &= \sum_{i=1}^k z_{n,d}^T \mathbf{c}_i \mathbf{c}_i^T z_{n,d} = \sum_{i=1}^k z_{n,d}^T C_i z_{n,d} \\ &= z_{n,d}^T (\sum_{i=1}^k C_i) z_{n,d}. \end{aligned} \tag{9}$$

Define $Q = \sum_{i=1}^k C_i$, then we have $p(x) = z_{n,d}^T Q z_{n,d}$. Since each $C_i = \mathbf{c}_i \mathbf{c}_i^T$ is positive semidefinite by construction, $Q \succeq 0$.

($\Longleftarrow$) Assume $p = z^T Q z$ where z has size m_1, $Q \in \mathbb{R}^{m_1 \times m_1}$, and $Q \succeq 0$. Since $Q \succeq 0$, there exists $B = [b_1, \dots, b_{m_2}] \in \mathbb{R}^{m_1 \times m_2}$ such that $Q = BB^T$. Then,

$$\begin{aligned} p(x) &= z^T Q z = z^T \begin{bmatrix} b_1 & \dots & b_{m_2} \end{bmatrix} \begin{bmatrix} b_1^T \\ \dots \\ b_{m_2}^T \end{bmatrix} z \\ &= z^T (\sum_{i=1}^{m_2} b_i b_i^T) z = \sum_{i=1}^{m_2} z^T b_i b_i^T z; \quad (\text{let } p_i(x) \doteq b_i^T z) \\ &= \sum_{i=1}^{m_2} p_i^2(x). \end{aligned} \tag{10}$$

It follows from the last equality that p is SoS. □

Proposition 3 establishes a relation between the coefficients of a polynomial and entries of a semidefinite matrix and forms the basis of reducing sum-of-squares programming problems to semi-definite programming problems.

Here are some additional useful facts:

- (Derivatives) Let p be a polynomial in n variables with degree less than d, with the representation $p(x) = \mathbf{c}^T z_{n,d}$ in polynomial basis $z_{n,d}$. Let $\tilde{p}(x) = \frac{\partial p(x)}{\partial x_j}$ have representation $p(x) = \tilde{\mathbf{c}}^T z_{n,d}$ on the same monomial basis, then $\tilde{\mathbf{c}}$ is depends linearly on $\mathbf{c}$. In particular, let $\mathbf{c} = [c_{\boldsymbol{\alpha}_1}, \ldots, c_{\boldsymbol{\alpha}_l}]$, where $c_{\boldsymbol{\alpha}_i} \in \mathbb{R}$, $\boldsymbol{\alpha}_i \in \mathbb{N}^n$ for $i = 1, \ldots, l$, and $l = \binom{n+d}{d}$. Also let $\tilde{\mathbf{c}} = [\tilde{c}_{\boldsymbol{\alpha}_1}, \ldots, \tilde{c}_{\boldsymbol{\alpha}_l}]$. Then, $\tilde{c}_{\boldsymbol{\alpha}_i} = (\boldsymbol{\alpha}_i(j) + 1) c_{\boldsymbol{\alpha}_i + e_j}$, for $\deg \boldsymbol{\alpha}_i < d$, and $\tilde{c}_{\boldsymbol{\alpha}_i} = 0$ otherwise. Here $e_j \in \mathbb{N}^n$ is a vector of all zeros and a single 1 at the j^{th} coordinate and $\boldsymbol{\alpha}_i(j)$ is the j^{th} coordinate of $\boldsymbol{\alpha}_i$. Therefore, there is a linear map H such that $\tilde{\mathbf{c}} = H\mathbf{c}$.

  Example 6. *Consider an arbitrary (possibly unknown) polynomial*

  $$p(x) = c_{[0,0]} + c_{[1,0]}x_1 + c_{[0,1]}x_2 + c_{[2,0]}x_1^2 + c_{[1,1]}x_1x_2 + c_{[0,2]}x_2^2$$

  in $n = 2$ variables with degree $d = 2$. Let

  $$\tilde{p}(x) = \frac{\partial p(x)}{\partial x_1} = \tilde{c}_{[0,0]} + \tilde{c}_{[1,0]}x_1 + \tilde{c}_{[0,1]}x_2 + \tilde{c}_{[2,0]}x_1^2 + \tilde{c}_{[1,1]}x_1x_2 + \tilde{c}_{[0,2]}x_2^2.$$

  Then, we have $\tilde{c}_{[0,0]} = c_{[1,0]}$, $\tilde{c}_{[1,0]} = 2c_{[2,0]}$, $\tilde{c}_{[0,1]} = c_{[1,1]}$, $\tilde{c}_{[2,0]} = 0$ $\tilde{c}_{[1,1]} = 0$ and $\tilde{c}_{[0,2]} = 0$. In matrix form, we have

  $$\tilde{\mathbf{c}} = H\mathbf{c} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{c} \tag{11}$$

  Example 7. *Consider the polynomial $p(x) = 3x_1^2 - x_2^2 = \mathbf{c}^T z_{2,2}$, for which $\mathbf{c} = \begin{bmatrix} 0 & 0 & 0 & 3 & 0 & -1 \end{bmatrix}^T$. Then, $\frac{\partial p(x)}{\partial x_1} = 6x_1 = \tilde{\mathbf{c}}^T z_{2,2}$, for which $\tilde{\mathbf{c}} = \begin{bmatrix} 0 & 6 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. It is easy to check that $\tilde{\mathbf{c}} = H\mathbf{c}$, where H is the matrix defined in Eq.* (11).

- (Addition) Let p_1, p_2 be polynomials in n variables with degree less than d, with the representations $p_1(x) = \mathbf{c}^T z_{n,d}$ and $p_2(x) = \mathbf{c}'^T z_{n,d}$. Then the polynomial $\tilde{p}(x) = p_1(x) + p_2(x)$ has the representation $\tilde{p}(x) = \tilde{\mathbf{c}}^T z_{n,d}$, where $\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{c}'$.

- (Multiplication) Let f be a given polynomial in n variables with degree less than d_1 and p be an arbitrary (possibly unknown) polynomial in n variables with degree less than d_1, with representations $f(x) = \mathbf{g}^T z_{n,d}$ and $p(x) = \mathbf{c}^T z_{n,d}$, respectively, where $d = d_1 + d_2$. Then the polynomial $\tilde{p}(x) = f(x)p(x)$ has the representation $\tilde{p}(x) = \tilde{\mathbf{c}}^T z_{n,d}$, where $\tilde{c}_{\boldsymbol{\alpha}_i} = \sum_{k,l:\boldsymbol{\alpha}_k+\boldsymbol{\alpha}_l=\boldsymbol{\alpha}_i} g_{\boldsymbol{\alpha}_l} c_{\boldsymbol{\alpha}_k}$. Therefore, there is a linear map $H(\mathbf{g})$ that depends on the coefficients of f such that $\tilde{\mathbf{c}} = H(\mathbf{g})\mathbf{c}$.

Example 8. *Consider an arbitrary (possibly unknown) polynomial*

$$p(x) = c_{[0,0]} + c_{[1,0]}x_1 + c_{[0,1]}x_2$$

in $n = 2$ variables with degree $d_1 = 1$. Let $f(x) = 3x_1 - x_2$ which is a polynomial in $n = 2$ variables with degree $d_2 = 1$. They have representations $p(x) = [c_{[0,0]}, c_{[1,0]}, c_{[0,1]}, 0, 0, 0]^T z_{n,d}$ and $f(x) = [0, 3, -1, 0, 0, 0]^T z_{n,d}$ in the basis $z_{n,d}$. Let

$$\tilde{p}(x) = f(x)p(x) = \tilde{c}_{[0,0]} + \tilde{c}_{[1,0]}x_1 + \tilde{c}_{[0,1]}x_2 + \tilde{c}_{[2,0]}x_1^2 + \tilde{c}_{[1,1]}x_1x_2 + \tilde{c}_{[0,2]}x_2^2.$$

Then, we have $\tilde{c}_{[0,0]} = 0$, $\tilde{c}_{[1,0]} = 3c_{[0,0]}$, $\tilde{c}_{[0,1]} = -c_{[0,0]}$, $\tilde{c}_{[2,0]} = 3c_{[1,0]}$ $\tilde{c}_{[1,1]} = -c_{[1,0]} + 3c_{[0,1]}$ and $\tilde{c}_{[0,2]} = -c_{[0,1]}$. In matrix form, we have

$$\tilde{\mathbf{c}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \mathbf{c}. \tag{12}$$

- (Generalized S-procedure) Given polynomials $f_0, \ldots f_k \in \mathcal{S}^n$, we have $f_0(x) \geq 0$ for all x such that $f_i(x) \geq 0$ for $i = 1, \ldots, k$ if there exists non-negative (relaxed to sum-of-squares) multipliers $\sigma_i(x) \in \mathbb{R}[x]$ $i = 1, \ldots, k$ such that $f_0(x) \geq \sum_{i=1}^k \sigma_i(x)f_i(x)$ for all $x \in \mathbb{R}^n$. In other words, if $f_0(x) \geq \sum_{i=1}^k \sigma_i(x)f_i(x)$, then

  $$\{x \in \mathbb{R}^n \mid f_0(x) \geq 0\} \supseteq \{x \in \mathbb{R}^n \mid f_i(x) \geq 0 \text{ for } i = 1, \ldots, k\}.$$

  The condition $f_0(x) \geq \sum_{i=1}^k \sigma_i(x)f_i(x)$ can be also written as $f_0(x) = \sigma_0(x) + \sum_{i=1}^k \sigma_i(x)f_i(x)$ with $\sigma_i(x)$ sum-of-squares, and it induces a set of affine equality conditions in the coefficients of the unknown polynomials σ_i (observe that it involves the multiplication and addition properties of the previous bullets).

A sum-of squares programming problem in feasibility form has the following structure [4]:

FEASIBILITY:

Find

polynomials $p_i(x)$, for $i = 1, 2, ..., \hat{N}$
sums of squares $p_i(x)$, for $i = (\hat{N}+1), ..., N$

such that

$$a_{0,j}(x) + \sum_{i=1}^{N} p_i(x) a_{i,j}(x) = 0, \quad \text{for } j = 1, 2, ..., \hat{J}, \tag{2.3}$$

$$a_{0,j}(x) + \sum_{i=1}^{N} p_i(x) a_{i,j}(x) \quad \text{are sums of squares } (\geq 0)^2,$$
$$\text{for } j = (\hat{J}+1), (\hat{J}+2), ..., J. \tag{2.4}$$

where p_i's are the unknown polynomials. Note that the feasible set is a convex set, that is if a set of p_is and p_i's are feasible so are $\lambda p_i + (1-\lambda)p_i'$ for $\lambda \in [0, 1]$.

Similarly, a sum of squares optimization problem takes the following form [4]:

OPTIMIZATION:

Minimize the linear objective function

$$w^T c,$$

where c is a vector formed from the (unknown) coefficients of

polynomials $p_i(x)$, for $i = 1, 2, ..., \hat{N}$
sums of squares $p_i(x)$, for $i = (\hat{N}+1), ..., N$

such that

$$a_{0,j}(x) + \sum_{i=1}^{N} p_i(x) a_{i,j}(x) = 0, \quad \text{for } j = 1, 2, ..., \hat{J}, \tag{2.5}$$

$$a_{0,j}(x) + \sum_{i=1}^{N} p_i(x) a_{i,j}(x) \quad \text{are sums of squares } (\geq 0),$$
$$\text{for } j = (\hat{J}+1), (\hat{J}+2), ..., J, \tag{2.6}$$

Example 9. *Consider a switched system governed by the following differential equation*

$$\dot{x}(t) = f_i(x(t)) \ \ for \ \ i \in \{1, \dots, k\},$$

where the mode signal i changes finitely many times in any finite interval, f_i's are polynomial vector fields, and 0 is an equilibrium point for each mode. If there exists a differentiable function $V(x)$ such that $V(x)$ such that $V(0) = 0$ and

$$\begin{array}{l} V(x) > 0 \quad \forall x \neq 0 \\ \frac{\partial V}{\partial x} f_i(x) < 0 \quad \forall x \neq 0 \;\; \forall i \in \{1, \ldots, k\}, \end{array} \tag{13}$$

then, the origin of the switched system is globally asymptotically stable under arbitrary switching (aka, globally absolutely stable) [5]. Problem (13) *can be converted to a sum-of-squares programming problem as follows:*

$$\begin{array}{rl} \textit{find} & p_0(x), p_1(x), \ldots, p_n(x) \\ \textit{such that} & p_0(0) = 0 \\ & p_0 \textit{ is SoS} \\ & p_i = \frac{\partial p_0}{\partial x_j} \;\; \forall j \in \{1, \ldots, n\} \\ & -\left(\begin{bmatrix} p_1(x) & \ldots & p_n(x) \end{bmatrix} f_i(x)\right) \textit{ is SoS } \; \forall i \in \{1, \ldots, k\}. \end{array} \tag{14}$$

Note that the first condition is affine on coefficients of p_0 as it basically tells the constant term of the polynomial should be zero. The second condition is a sum-of-squares condition enforcing positivity. The third condition is a set of affine conditions on coefficients of $p_0, \ldots, p_n$ based on differentiation property. And the last condition is a set of affine sum-of-squares conditions on coefficients of $p_1, \ldots, p_n$. Therefore, problem (14) *is convex and can be solved using semidefinite programming solvers* [5]*. An example where the vector fields f_i are linear is provided on ctools website.*

Remark 2. *Being sum-of-squares only implies non-negativity. In Lyapunov theory, we typically need positive definite functions that are zero only at zero. This condition is not easy to impose in numerical solvers. One way of imposing it is to use $p(x) - \sigma(x) = SoS$ where $\sigma(x) > 0$ is a known positive polynomial. However, this is typically not required. Because of the way SDP solvers work, they almost always find solutions that are positive definite. So, you can use $p(x) = SoS$ and do a post-processing to ensure that the solution found is positive.*

1.3 Remarks on optimization software

There are several optimization solvers specialized for different types of problems. Here is a fairly incomplete list:

[5] see Remark 2

- There are certain high-level modeling languages that can recognize a large class of convex programming problems and provide interfaces to appropriate solvers:
  - YALMIP (MATLAB): `http://users.isy.liu.se/johanl/yalmip/`
  - cvx (MATLAB): `http://cvxr.com/cvx/`
  - cvxopt (Python): `http://cvxopt.org/` Indeed YALMIP provides interfaces to more general nonlinear programming solvers and handles sum-of-squares constraints as well.
- MATLAB has its own optimization toolbox that provides linear programming (e.g., linprog) and nonlinear programming (e.g., fmincon) solvers.
- SOSTOOLS (MATLAB): `http://sysos.eng.ox.ac.uk/sostools/` is a toolbox for solving sum-of-squares programs.
- There are commercial solvers in various languages (educational licenses available):
  - mosek (`http://www.mosek.com/`),
  - cplex (`http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`),
  - gurobi (`http://www.gurobi.com/`), etc.
- This page provides benchmarks for different optimization problems and compares a large set of solvers on these benchmark problems: `http://plato.asu.edu/bench.html`.

References

[1] Boyd, Stephen, and Vandenberghe Lieven. Convex optimization. Cambridge university press, 2004.

[2] Parrilo, Pablo. Lecture notes for the course "Algebraic techniques and semidefinite optimization", 2006 (available from MIT open courseware).

[3] Polik, Imre, and Tamas Terlaky. "A survey of the S-lemma." SIAM review 49.3 (2007): 371-418.

[4] Papachristodoulou, Antonis, et al. "SOSTOOLS Version 3.00 Sum of Squares Optimization Toolbox for MATLAB." arXiv preprint `http://arxiv.org/abs/1310.4716` (2013).

[5] Prajna, Stephen, and Antonis Papachristodoulou. "Analysis of switched and hybrid systems-beyond piecewise quadratic methods." American Control Conference, 2003. Proceedings of the 2003. Vol. 4. IEEE, 2003.