Massachusetts Institute of Technology

Department of Electrical Engineering and Computer Science

6.245: MULTIVARIABLE CONTROL SYSTEMS

by A. Megretski

Convex Optimization[1]

Many optimization objectives generated by LTI system design and analysis do not fit within the frameworks of H2/H-Infinity optimization or Hankel optimal model reduction, but are still relatively easy to work with. In most cases, such objectives are characterized by *convexity* of the underlying constraints. This lecture is devoted to recognizing and working with convex constraints.

15.1 Basic Definitions of Finite Dimensional Convex Analysis

In this subsection, basic definitions of convex optimization with finite number of decision parameters are given.

15.1.1 Convex Sets

A subset Ω of $V = \mathbf{R}^n$ is called *convex* if

$$cv_1 + (1-c)v_2 \in \Omega \quad \text{whenever} \quad v_1, v_2 \in \Omega,\ c \in [0,1].$$

In other words, a set is convex whenever the line segment connecting any two points of Ω lies completely within Ω.

In many applications, the elements of Ω are, formally speaking, not vectors but other mathematical objects, such as matrices, polynomials, etc. What matters, however, is that Ω is a subset of a set V such that a one-to-one correspondence between $\mathbf{R}^n$ and V

[1]Version of April 12, 2004

is established for some n. We will refer to V as a (real finite dimensional) *vector space*, while keeping in mind that V is the same as $\mathbf{R}^n$ for some n. For example, the set $\mathbf{S}^n$ of all symmetric n- by-n matrices is a vector space, because of the natural one-to-one correspondence between $\mathbf{S}^n$ and $\mathbf{R}^{n(n+1)/2}$.

Using this definition directly, in some situations it would be rather difficult to check whether a given set is convex. The following simple statement is of a great help.

Lemma 15.1 *Let K be a set of* affine functionals *on $V = \mathbf{R}^n$, i.e. elements $f \in K$ are functions $f:\ V \to \mathbf{R}$ such that*

$$f(cv_1 + (1-c)v_2) = cv_1 + (1-c)v_2 \quad \forall\ c \in \mathbf{R}, \quad v_1, v_2 \in V.$$

Then the subset Ω of V defined by

$$\Omega = \{v \in V:\ f(v) \geq 0 \ \ \forall\ f \in K\}$$

is convex.

In other word, any set defined by linear inequalities is convex.

Proof Let $v_1, v_2 \in \Omega$ and $c \in [0, 1]$. Since $f(v_1) \geq 0$ and $f(v_2) \geq 0$ for all $f \in K$, and $c \geq 0$ and $1 - c \geq 0$, we conclude that

$$f(cv_1 + (1-c)v_2) = cf(v_1) + (1-c)f(v_2) \geq 0$$

for all $f \in K$. Hence $cv_1 + (1-c)v_2 \in K$. ■

Here is an example of how Lemma 15.1 can be used. Let us prove that the subset $\Omega = \mathbf{S}^n_+$ of the set $V = \mathbf{S}^n$ of symmetric n-by-n matrices, consisting of all *positive semidefinite matrices*, is convex.

Note that doing this via the "nonnegative eigenvalues" definition of positive semidefiniteness would be difficult. Luckily, there is another definition: a matrix $M \in \mathbf{S}^n_+$ is positive semidefinite if and only if $x'Mx \geq 0$ for all $x \in \mathbf{C}^n$. Note that any $x \in \mathbf{C}^n$ defines an affine (actually, a linear) functional $f = f_x :\ \mathbf{S}^n \to \mathbf{R}$ according to

$$f_x(M) = x'Mx.$$

Hence, $\mathbf{S}^n_+$ is a subset of $\mathbf{S}^n$ defined by some (infinite) set of linear inequalities. According to Lemma 15.1, $\mathbf{S}^n_+$ is a convex set.

15.1.2 Convex Functions

Let $f : \Omega \to \mathbf{R}$ be a function defined on a subset $\Omega \subset V = \mathbf{R}^n$. Function f is called *convex* if the set

$$\Gamma_f^+ = \{(v, y) \in \Omega \times \mathbf{R} : \ y \geq f(v)\},$$

is a convex subset of $V \times \mathbf{R}$.

According to this definition, $f : \Omega \to \mathbf{R}$ is convex if and only if the following two conditions hold:

(a) Ω is convex;

(b) the inequality

$$f(cv_1 + (1-c)v_2) \leq cf(v_1) + (1-c)f(v_2)$$

holds for all $v_1, v_2 \in V$, $c \in [0, 1]$.

Note that condition (b) has the meaning that any segment connecting two points on the graph of f lies em above the graph of f.

The definition of a convex function does not help much with proving that a given function is convex. The following three statements are of great help in establishing convexity of functions.

Let us call a function $f : \Omega \to \mathbf{R}$ defined on a subset Ω of $\mathbf{R}^n$ *twice differentiable* at a point $v_0 \in \Omega$ if there exists a symmetric matrix $W \in \mathbf{S}^n_{\mathbf{R}}$ and a row vector p such that

$$\frac{f(v) - f(v_0) - p(v - v_0) - 0.5(v - v_0)'W(v - v_0)}{\|v - v_0\|^2} \to 0 \ \text{ as } \ v \to v_0, \ v \in \Omega,$$

in which case $p = f'(v_0)$ is called the *first derivative* of f at v_0 and $W = f''(v_0)$ is called the *second derivative* of f at v_0.

Lemma 15.2 *Let $\Omega \subset \mathbf{R}^n$ be a convex subset of $\mathbf{R}^n$. Let $f : \Omega \to \mathbf{R}$ be a function which is twice differentiable and has a positive semidefinite second derivative $W = f''(v_0) \geq 0$ at any point $v_0 \in \Omega$. Then f is convex.*

For example, let Ω be the positive quadrant in $\mathbf{R}^2$, i.e. the set of vectors $[x; y] \in \mathbf{R}^2$ with positive components $x > 0$, $y > 0$. Obviously Ω is convex. Let the function $f : \Omega \to \mathbf{R}$ be defined by $f(x, y) = 1/xy$. According to Lemma 15.2 is convex, because the second derivative

$$W(x, y) = \begin{bmatrix} d^2f/dx^2 & d^2f/dxdy \\ d^2f/dydx & d^2f/dy^2 \end{bmatrix} = \begin{bmatrix} 2/x^3y & 1/x^2y^2 \\ 1/x^2y^2 & 2/xy^3 \end{bmatrix}$$

is positive definite on Ω.

Lemma 15.3 *Let $\Omega \subset V$ be a convex set of a $V = \mathbf{R}^n$. Let P be a set of affine functionals on V such that*

$$f(v) = \sup_{p \in P} p(v) < \infty \quad \forall\, v \in \Omega.$$

Then $f:\ \Omega \to \mathbf{R}$ is a convex function.

To give an example of how Lemma 15.3 can be used, let us prove that the function $f:\ \mathbf{C}^{n,m} \to \mathbf{R}$ defined by $f(M) = \sigma_{\max}(M)$ is convex, where $\mathbf{C}^{n,m}$ denotes the set of all n-by-m matrices with complex entries. Though $\Omega = \mathbf{C}^{n,m}$ is in a simple one-to-one correspondence with $\mathbf{R}^{2nm}$, using Lemma 15.2 to prove convexity of f is essentially impossible: f is not differentiable at many points, and its second derivative, where exists, is cumbersome to calculate. Luckily, from linear algebra we know that

$$\sigma_{\max}(M) = \max\{\mathrm{Re}(p' M q) :\ p \in \mathbf{C}^n,\ q \in \mathbf{C}^m,\ \|p\| = \|q\| = 1\}.$$

Since each individual function $M \mapsto \mathrm{Re}(p' M q)$ is linear, Lemma 15.3 implies that f is convex.

In addition to Lemma 15.2 and Lemma 15.3, which help establishing convexity "from scratch", the following statements can be used to derive convexity of one function from convexity of other functions.

Lemma 15.4 *Let V be a vector space, $\Omega \subset V$.*

(a) If $f:\ \Omega \to \mathbf{R}$ and $g:\ \Omega \to \mathbf{R}$ are convex functions then $h:\ \Omega \to \mathbf{R}$ defined by $h(v) = f(v) + g(v)$ is convex as well.

(b) If $f:\ \Omega \to \mathbf{R}$ is a convex function and $c > 0$ is a positive real number then $h:\ \Omega \to \mathbf{R}$ defined by $h(v) = cf(v)$ is convex.

(c) If $f:\ \Omega \to \mathbf{R}$ is a convex function, U is a vector space, and $L:\ U \to V$ is an affine *function, i.e.*

$$L(cu_1 + (1-c)u_2) = cL(u_1) + (1-c)L(u_2) \quad \forall\, c \in \mathbf{R},\ u_1, u_2 \in U,$$

then the set

$$L^{-1}(\Omega) = \{u \in U :\ L(u) \in \Omega\}$$

is convex, and the function $f \circ L:\ L^{-1}(\Omega) \to \mathbf{R}$ defined by $(f \circ L)(u) = f(L(u))$ is convex.

For example, let $g: \mathbf{S}^3_{\mathbf{R}} \to \mathbf{R}$ be defined on symmetric 2-by-2 matrices by

$$g\left(\begin{bmatrix} x & y \\ y & z \end{bmatrix}\right) = x^2 + y^2 + z^2.$$

To prove that g is convex, note that $g = f \circ L$ where $L: \mathbf{S}^3 \to \mathbf{R}^3$ is the affine (actually, linear) function defined by

$$L\left(\begin{bmatrix} x & y \\ y & z \end{bmatrix}\right) = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

and $f: \mathbf{R}^3 \to \mathbf{R}$ is defined by

$$f\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = x^2 + y^2 + z^2.$$

Lemma 15.2 can be used to establish convexity of f (the second derivative of f turns out to be the identity matrix). According to Lemma 15.4, g is convex as well.

15.1.3 Quasi-Convex Functions

Let $\Omega \subset V$ be a subset of a vector space. A function $f: \Omega \to \mathbf{R}$ is called *quasi-convex* if its *level sets*

$$\Omega_\gamma = \{v \in \Omega : f(v) < \gamma\}$$

are convex for all γ.

It is easy to prove that any convex function is quasi-convex. However, there are many important *quasi-convex* functions which are not convex. For example, let $\Omega = \{(x, y) : x > 0, y > 0\}$ be the positive quadrant in $\mathbf{R}^2$. The function $f: \Omega \to \mathbf{R}$ defined by $f(x, y) = -xy$ is not convex but quasi-convex.

A rather general definition leading to quasi-convex functions is given as follows.

Lemma 15.5 *Let $\Omega \subset V$ be a subset of a vector space. Let $P = \{(p, q)\}$ be a set of pairs of affine functionals $p, q: \Omega \to \mathbf{R}$ such that*

(a) inequality $p(v) \geq 0$ holds for all $v \in \Omega$, $(p, q) \in P$;

(b) for any $v \in \Omega$ there exists $(p, q) \in P$ such that $p(v) > 0$.

Then the function $f: \Omega \to \mathbf{R}$ defined by

$$f(v) = \inf\{\lambda : \lambda p(v) \geq q(v) \ \forall \ (p,q) \in P\} \tag{15.1}$$

is quasi-convex.

For example, the *largest generalized eigenvalue* function $f(v) = \lambda_{\max}(\alpha, \beta)$ defined on the set $\Omega = \{v\}$ of pairs $v = (\alpha, \beta)$ of matrices $\alpha, \beta \in \mathbf{S}^n$ such that α is positive semidefinite and $\alpha \neq 0$, is quasi-convex. To prove this, recall that

$$\lambda_{\max}(\alpha, \beta) = \inf\{\lambda : \lambda x'\alpha x \geq x'\beta x \ \forall \ x \in \mathbf{C}^n\}.$$

This is a representation of $\lambda_{\max}$ in the form (15.1) with $((p,q) = (p_x, q_x)$ defined by an $x \in \mathbf{C}^n$ according to

$$p_x(v) = x'\alpha x, \ q_x(v) = x'\beta x \quad \text{where} \quad v = (\alpha, \beta).$$

Since for any $\alpha \geq 0$, $\alpha > 0$ there exists $x \in \mathbf{C}$ such that $x'\alpha x > 0$, Lemma 15.5 implies that $\lambda_{\max}$ is quasi-concave on Ω.

15.2 Standard Convex Optimization Setups

There exists a variety of significantly different tasks commonly referred to as convex optimization problems.

15.2.1 Minimization of a Convex Function

The standard general form of a convex optimization problem is *minimization* $f(v) \to \min$ of a *convex function* $f: \Omega \to \mathbf{R}$.

The remarkable feature of such optimization is that for any point $v \in \Omega$ which is *not* a minimum of f and for any number $\gamma \in (\inf(f), f(v))$ there exists a vector u such that $v + tu \in \Omega$ and $f(v + tu) \leq f(v) + t(\gamma - f(v))$ for all $t \in [0, 1]$. (In other words, f is decreasing quickly in the direction u.) In particular, any local minimum of a convex function is its global minimum.

While it is reasonable to expect that convex optimization problems are easier to solve, and reducing a given design setup to a convex optimization is frequently a major step, it must be understood clearly that convex optimization problems are useful only when the task of calculating $f(v)$ for a given v (which includes checking that $v \in \Omega$) is not too complicated.

For example, let X be any finite set and let $g: X \to \mathbf{R}$ be *any* real-valued function on X. Minimizing g on X can be very tricky when the size of X is large (because there is very little to offer apart from the random search). However, after introducing the vector space V of all functions $v: X \to \mathbf{R}$, the convex set Ω can be defined as the set of all *probability distributions* on X, i.e. as the set of all $v \in V$ such that

$$v(x) \geq 0 \;\forall\; x, \quad \sum_{x \in X} v(x) = 1,$$

and $f: \Omega \to \mathbf{R}$ can be defined by

$$f(v) = \sum_{x \in X} g(x)v(x).$$

Then f is convex and, formally speaking, minimization of g on X is "equivalent" to minimization of f on Ω, in the sense that the argument of minimum of f is a function $v \in \Omega$ which is non-zero only at those $x \in X$ for which $g(x) = \min(g)$. However, unless some nice simplification takes place, $f(v)$ is "difficult" to evaluate for any particular v (the "brute force" way of doing this involves calculation of $g(x)$ for *all* $x \in X$), this "reduction" to the convex optimization does not make much sense.

15.2.2 Linear Programs

As it follows from Lemma 15.1, a convex set Ω can be defined by a family of linear inequalities. Similarly, according to Lemma 15.3, a convex function can be defined as supremum of a family of affine functions. The problem of finding the minimum of f on Ω when Ω is a subset of $\mathbf{R}^n$ defined by a *finite* family of linear inequalities, i.e.

$$\Omega = \{v \in \mathbf{R}^n : \; a_i'v \leq b_i, \; i = 1, \dots, m\}, \tag{15.2}$$

and $f: \Omega \to \mathbf{R}$ is defined as supremum of a *finite* family of affine functions,

$$f(v) = \max_{i=1,\dots,k} c_i'v + d_i, \tag{15.3}$$

where a_i, c_i are given vectors in $\mathbf{R}^n$, and b_i, d_i are given real numbers, is referred to as a *linear program.*

In fact, all linear programs defined by (15.2),(15.3) can be reduced to the case when f is a *linear* function, by appending an extra component v_{n+1} to v, so that the new decision variable becomes

$$\bar{v} = \begin{bmatrix} v \\ v_{n+1} \end{bmatrix} \in \mathbf{R}^{n+1},$$

introducing the additional linear inequalities

$$\bar{c}_i'\bar{v} = c_i'v - v_{n+1} \leq -d_i,$$

and defining the new objective function $\bar{f}$ by

$$\bar{f}(\bar{v}) = v_{n+1}.$$

Most linear programming optimization engines would work with the setup (15.2),(15.3), where $f(v) = Cv$ is a linear function. The common equivalent notation in this case is

$$Cv \to \min \quad \text{subject to} \quad Av \leq B,$$

where a_i' are the rows of A, b_i are the elements of the column vector B, and the inequality $Av \leq B$ is understood component-wise.

15.2.3 Semidefinite Programs

A *semidefinite program* is typically defined by an affine function $\alpha: \ \mathbf{R}^n \to \mathbf{S}_{\mathbf{R}}^N$ and a vector $c \in \mathbf{R}^n$, and is formulated as

$$c'v \to \min \quad \text{subject to} \quad \alpha(v) \geq 0. \tag{15.4}$$

Note that in the case when

$$\alpha(v) = \begin{bmatrix} b_1 - a_1'v & & 0 \\ & \ddots & \\ 0 & & b_N - a_N'v \end{bmatrix}$$

is a diagonal matrix valued function, the special semidefinite program becomes a general linear program. Therefore, linear programming is a special case of semidefinite programming.

Since a single matrix inequality $\alpha \geq 0$ represents an infinite number of inequalities $x'\alpha x \geq 0$, semidefinite programs can be used to represent constraints much more efficiently than linear programs. The KYP Lemma explains the special importance of linear matrix inequalities in system analysis and optimization. On the other hand, software for solving general semidefinite programs appears to be not as well developed as in the case of linear programming.

15.2.4 Smooth Convex Optimization

Smooth convex optimization involves minimization of a twice differentiable convex function $f : \Omega \to \mathbf{R}$ on an *open* convex set $\Omega \subset \mathbf{R}^n$ in the situation when $f(v)$ approaches infinity whenever v approaches the boundary of Ω or infinity.

This case can be solved very efficiently using an iterative algorithm which updates its current guess v_t at the minimum in the following way. Let $p_t' = f'(v_t)$, $W_t = f''(v_t) > 0$. Keeping in mind that

$$f(v_t + \delta) \approx \sigma_t(\delta) = f(v_t) + p_t'\delta + 0.5\delta' W_t \delta$$

can be approximated by a quadratic form, let

$$\delta_t = -W_t^{-1} p_t$$

be the argument of minimum of $\sigma_t(\delta)$. Let $\tau = \tau_t$ be the argument of minimum of $f_t(\tau) = f(v_t + \tau\delta)$ (since τ is a scalar, such a minimum is usually easy to find). Then set $v_{t+1} = v_t + \tau_t \delta_t$ and repeat the process.

Actually, non-smooth convex optimization problems (such as linear and semidefinite programs) are frequently solved by reducing them to a sequence of smooth convex optimizations.

15.2.5 Feasibility Search and Quasi-Convex Optimization

Convex feasibility search problems are formulated as the problems of finding an element in a convex set Ω described implicitly by a set of convex constraints. In most situations, it is easy to convert a convex feasibility problem to a convex optimization problem. For example, the problem of finding a $x \in \mathbf{R}^n$ satisfying a finite set of linear inequalities $a_i' x \le b_i$, $i = 1, \dots, N$, can be converted to a linear program

$$y \to \min \quad \text{subject to} \quad a_i' x - y \le b_i, \ (i = 1, \dots, N).$$

If $y = y_0 \le 0$ for some $v_0 = (y_0, x_0)$ satisfying the constraints then $x = x_0$ is a solution of the original feasibility problem. Otherwise, if y is always positive, the feasibility problem has no solution.

In turn, quasi-convex optimization problems can be reduced to convex feasibility search. Consider the problem of minimization of a given quasi-convex function $f : \Omega \to \mathbf{R}$. Assume for simplicity that the values of f are limited to an interval $[f_{\max}, f_{\min}]$. As in the algorithm for H-Infinity optimization, set $\gamma_- = f_{\min}$, $\gamma_+ = f_{\max}$, and repeat the following step until the ratio $(\gamma_+ - \gamma_-)/(f_{\max} - f_{\min})$ becomes small enough: solve the convex feasibility problem of finding $v \in \Omega$ such that $f(v) \le \gamma$ where $\gamma = 0.5(\gamma_- + \gamma_+)$; if such v exists, set $\gamma_- = \gamma$, otherwise set $\gamma_+ = \gamma$.

15.2.6 Relaxation to a Linear Program

In applications, general convex programs are frequently "relaxed" to linear programs by replacing infinite families of linear constraints with their finite "approximations". The resulting linear program then serves as a source of *lower bounds* of the minimal cost in the original optimization problem.

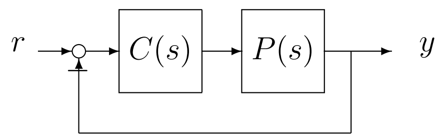

Figure 15.1: A SISO feedback design setup

Consider, for example, the task of designing an output feedback controller $C(s)$ from Figure 15.1 such that the closed loop response $y(t)$ to the unit step $r(t) = u(t)$ settles to the steady output value of $y(t) = 1$ in a given time T while minimizing the maximal overshoot $\max y(t) - 1$ and undershoot $-\min y(t)$. Note that while this setup is not expected to yield a practically useful controller (the order of C cannot be fixed without losing the convexity of the problem formulation) it brings valuable insignt about the limitations of the output feedback design, especially when the given plant P is not minimum-phase (i.e. has strictly unstable zeros).

Let

$$T^y(s) = T_0^y(s) + T_1^y(s)Q(s), \quad T^u(s) = T_0^u(s) + T_1^u(s)Q(s)$$

be the Q-parameterization of all closed loop transfer functions from $\dot{r}$ to y and r. Let

$$h^y(t) = h_0^y(t) + h_1^y(t) \star q(t), \quad h^u(t) = h_0^u(t) + h_1^u(t) \star q(t)$$

be the corresponding parameterization of the impulse responses, where $\star$ denotes convolution, and $q = q(t)$ is the impulse response of the $Q = Q(s)$ parameter. Let $V = \{v = q(\cdot)\}$ be the vector space of all functions $q = q(t)$ which decay exponentially as $t \to \infty$. Let Ω be the subset of V defined by the condition

$$\pm h^u(t) \leq \epsilon, \quad \pm(h^y(t) - 1) \leq \epsilon \quad \text{for} \quad t > T,$$

where $\epsilon > 0$ is a small parameter describing the amount of after-action permitted for $t > T$ Let $f : \Omega \to \mathbf{R}$ be defined as

$$f(q(\cdot)) = \max\{\max(h^y(t) - 1), \max(-h^y(t))\}.$$

By its definition as a maximum of affine functions of the decision parameter $q(\cdot)$, f is convex. The convex optimization problem $f(v) \to \min$ is equivalent to the search for a stabilizing controller which minimizes the overshoot and the undershoot while keeping the prescribed settling time T.

The exact formulation of the convex optimization problem involves an infinite number of linear inequalities (at all $t \in \mathbf{R}$). In practice the original setup should be replaced with an approximation relying on a finite set of linear inequalities. For example, this can be done by sampling the original inequalities at a finite number of time points $t = t_i$, $i = 1, \dots, N$.

15.3 Duality

Duality is extremely important for understanding convex optimization. Practically, it delivers a major way of derivin lower bounds in convex minimization problems.

15.3.1 Dual optimization problem and duality gap

According to the remarks made before, a rather eneral class of convex optimization problems is represented by the setup

$$f(v) = \max_{r \in \mathcal{R}} \{a_r v + b_r\} \to \min \text{ subject to } v \in \Omega = \{v : \max_{k \in \mathcal{K}} \{c_k v + d_k\} \leq 0\}, \tag{15.5}$$

where a_r, c_k are given row vectors indexed by $r \in \mathcal{R}$, $k \in \mathcal{K}$ (the sets $\mathcal{R}$, $\mathcal{K}$ are not necessarily finite), b_r, d_k are given real numbers, and v is a column decision vector. When $\mathcal{K}, \mathcal{R}$ are finite sets, (15.5) defines a linear program.

Consider some functions $u : \mathcal{R} \mapsto \mathbf{R}$ and $q : \mathcal{K} \mapsto \mathbf{R}$ which assign real numbers to the indexes, in such a way that only a countable set of values $u(r) = u_r$, $q(k) = q_k$ is positive, and

$$u_k \geq 0, \ \sum_k u_k = 1, \quad q_r \geq 0, \ \sum_r q_r \leq 1. \tag{15.6}$$

Obviously,

$$f(v) \geq \sum_r u_r (a_r v + b_r),$$

and

$$\sum_k q_k (c_k v + d_k) \leq 0 \ \ \forall \ v \in \Omega.$$

Hence

$$f(v) \geq \sum_r u_r b_r + \sum_k q_k d_k \ \ \forall \ v \in \Omega$$

whenever

$$\sum_r u_r a_r + \sum_k q_k c_k = 0, \tag{15.7}$$

in which case

$$\sum_r u_r b_r + \sum_k q_k d_k \tag{15.8}$$

is a lower bound for the minimum in (15.5). Trying to maximize the lower bound leads to the task of maximizing (15.8) subject to (15.6),(15.7). This task, a convex optimization problem itself, is called *dual* with respect to (15.5).

The key property of the dual problem is that its maximum (more precisely, supremum, since the maximum is not necessarily achievable) equals the minimum (infimum) in the original optimization problem (15.5).

15.3.2 The Hahn-Banach Theorem

The basis for all convex duality proofs is the fundamental *Hahn-Banach Theorem*. The theorem can be formulated in two forms: geometric (easier to understand) and functional (easier to prove).

By definition, an element v_0 of a real vector space V is called an *interior point* of a subset $\Omega \subset V$ if for every $v \in V$ there exists $\epsilon = \epsilon_v > 0$ such that $v_0 + tv \in \Omega$ for all $|t| < \epsilon_v$.

Theorem 15.1 *Let Ω is a convex subset of a real vector space V such that 0 is an interior point of Ω. If $v_0 \in V$ is not an interior point of Ω then there exists a linear function $L: V \mapsto \mathbf{R}$, $L \not\equiv 0$, such that*

$$L(v_0) \geq \sup_{v \in \Omega} L(v).$$

In other words, a point not strictly inside a convex set can be separated from the convex set by a hyperplane.

To give an alternative formulation of the Hahn-Banach Theorem, remember that a non-negative function $q: V \mapsto \mathbf{R}$ defined on a real vector space V is called a *semi-norm* if it is convex and positively homogeneous (i.e. $p(av) = ap(v)$ for all $a \geq 0$, $v \in V$).

Theorem 15.2 *Let $q: V \mapsto \mathbf{R}$ be a semi-norm on a real vector space V. Let V_0 be a linear subspace of V, and $h_0: V_0 \mapsto \mathbf{R}$ be a linear function such that $q(v) \geq h_0(v)$ for all $v \in V_0$. Then there exists a linear function $h: V \mapsto \mathbf{R}$ such that $h(v) = h_0(v)$ for all $v \in V_0$, and $h(v) \leq q(v)$ for all $v \in V$.*

To relate the two formulations, define $q(v)$ as the *Minkovski' functional* of Ω:

$$q(v) = \inf\{t > 0 :\ t^{-1}v \in \Omega\},$$

and set

$$V_0 = \{tv_0 :\ t \in \mathbf{R}\}, \quad h_0(tv_0) = t.$$

15.3.3 Duality gap for linear programs

To demonstrate utility of the Hahn-Banach theorem, let us use it to prove the "zero duality gap" statement for linear programs.

Theorem 15.3 *Let A, B, C be real matrices of dimensions n-by-m, n-by-1, and 1-by-m respectively. Assume that there exists $v_0 \in \mathbf{R}^m$ such that $Av_0 < B$. Then*

$$\sup\{Cv :\ v \in \mathbf{R}^m,\ Av \le B\} = \inf\{B'p :\ p \in \mathbf{R}^n,\ A'p = C',\ p \ge 0\}. \tag{15.9}$$

The inequalities $Av \le B$, $Av_0 < B$, and $p \ge 0$ in (15.9) are understood component-wise. Note also that inf over an empty set equals *plus infinity*. This can be explained by the fact that inf is the *maximal lower bound* of a set. Since every number is a lower bound for an empty set, its infimum equals $+\infty$. Theorem 15.3 remains valid when there exist no $p \ge 0$ such that $A'p = C'$, in which case it claims that inequality $Av \le B$ has infinitely many solutions, among which Cv can be made arbitrarily small.

Proof The inequality

$$\sup\{Cv :\ v \in \mathbf{R}^m,\ Av \le B\} \le \inf\{B'p :\ p \in \mathbf{R}^n,\ A'p = C',\ p \ge 0\}$$

is straightforward: multiplying $Av \le B$ by $p' \ge 0$ on the left yields $p'Av \le B'p$; when $A'p = C'$, this yields $Cv \le B'p$.

The proof of the inverse inequality

$$\sup\{Cv :\ v \in \mathbf{R}^m,\ Av \le B\} \ge \inf\{B'p :\ p \in \mathbf{R}^n,\ A'p = C',\ p \ge 0\}$$

relies on the Hahn-Banach theorem.

Let y be an upper bound for Cv subject to $Av \le B$. If $y = \infty$ then, according to the already proven inequality, there exist no $p \ge 0$ such that $A'p = C'$, and hence the desired equality holds.

If $y < \infty$, let e denote the n-by-1 vector with all entries equal to 1. Consider the set

$$\Omega = \left\{ x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} Cv - \delta + 1 \\ e - Av - \Delta \end{bmatrix} \in \mathbf{R}^{n+1} :\ \Delta > 0,\ \delta > 0 \right\}.$$

Then

(a) Ω is a convex set (as a linear transformation image of a set defined by linear inequalities);

(b) zero is an interior point of Ω (because it contains the open cube $|x_i| < 1$, which can be seen by setting $v = 0$);

(c) vector $[y + 1; e - B]$ does not belong to Ω (otherwise $Av + \Delta = B$ and $Cv - \delta = y$, which contradicts the assumption that $Cv \leq y$ whenever $Av \leq B$).

According to the Hahn-Banach Theorem, this means that there exists a non-zero linear functional

$$L \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} = L_0 x_0 + \bar{L}' \bar{x},$$

where $L_0 \in \mathbf{R}$, $\bar{L} \in \mathbf{R}^n$, defined on $\mathbf{R}^{n+1}$, such that

$$L_0(Cv - \delta + 1) + \bar{L}'(e - Av - \Delta) \leq L_0(y + 1) + \bar{L}'(e - B) \quad \forall\ \Delta > 0, \delta > 0, v. \qquad (15.10)$$

Looking separately at the coefficients at v, δ, Δ and at the constant term in (15.10) implies

$$L_0 C = \bar{L}' A, \ \ L_0 \geq 0, \ \ \bar{L} \geq 0, \ \ L_0 y \geq \bar{L}' B. \qquad (15.11)$$

Note that L_0 cannot be equal to zero: otherwise $\bar{L}'A = 0$ and $\bar{L}'B \geq 0$, which, after multiplying $Av_0 < B$ by $\bar{L} \geq 0$, $\bar{L} \neq 0$ yields a contradiction:

$$0 = \bar{L}' A v_0 < \bar{L}' B \leq L_0 y = 0.$$

If $L_0 > 0$ then for

$$p = \bar{L}/L_0$$

conditions (15.11) imply

$$A'p = C', \quad p \geq 0, \quad B'p \leq y.$$

■

15.4 Solving convex programs

This section describes software which can be used for solving convex optimization problems in this class, and gives examples of control system related problems solved using this software.

15.4.1 Linear programming

The optimization toolbox of MATLAB provides function *linprog.m* for solving linear programs. The simplest call format is

```
v=linprog(C',A,B)
```

to solve the problem of *minimizing* Cv subject to $Av \leq B$.

My past experience with this function is not very positive: it starts failing already for very moderately sized tasks. An alternative (and also a *free* option) is the SeDuMi package, which can be downloaded from

```
http://fewcal.kub.nl/sturm/software/sedumi.html
```

When SeDuMi is installed, it can be used to solve simultaneously the dual linear programs

$$Cx \to \max \quad \text{subject to} \quad B - Ax \geq 0$$

and

$$B'p \to \min \quad \text{subject to} \quad A'p = C', \ p \geq 0$$

by calling

```
[p,x]=sedumi(A',C',B);
```

Actually, the indended use of SeDuMi is solving semidefinite programs, which can be achieved by changing the interpretation of the ≥ 0 condition (set by the fourth argument of `sedumi`). In general, inequality $z \geq 0$ will be interpreted as $z \in \mathcal{K}$, where $\mathcal{K}$ is a *self-dual cone*. Practically speaking, by saying that $z \in \mathcal{K}$ one can specify that certain elements of vector z must form positive semidefinite matrices, instead of requiring the elements to be non-negative.

Note that both `linprog.m` and `sedumi.m` require the primal and dual optimization problems to be *strictly feasible* (i.e. inequalities $Ax < B$ and $p > 0$ subject to $A'p = C'$ must have solutions). One can argue that a well formulated convex optimization problem should satisfy this condition anyway.

15.4.2 Semidefinite programming

While SeDuMi is easy to apply for solving *some* semidefinite programs, it is frequently inconvenient for situations related to control systems analysis and design. A major need there is to be able to define matrix equalities or inequalities in a "block format", such as in the case of a Lyapunov inequality

$$AP + PA' = Y \geq 0, \;\; P > 0$$

where A is a given square matrix, and $P = P'$, $Y = Y'$ are matrix decision parameters. The LMI Control Toolbox of MATLAB provides interface commands for defining linear matrix inequalities in a block matrix format. However, this interface itself is quite scriptic, and hence is not easy to work with.

The package IQCbeta, freely available from

```
http://www.math.kth.se/~cykao/
```

and already installed on Athena, helps to cut significantly the coding effort when solving semidefinite programs.

Here is an example of a function which will minimize the largest eigenvalue of $PA+A'P$ where A is a given matrix, and P is the symmetric matrix decision variable satisfying $0 \leq P \leq I$.

```
function P=example_sdp_lyapunov(A)
% function P=example_sdp_lyapunov(A)
%
% demonstrates the use of IQCbeta by finding P=P' which minimizes
% the largest eigenvalue of PA+A'P subject to 0<=P<=I

n=size(A,1);         % problem dimension
abst_init_lmi;       % initialize the LMI solving environment
p=symmetric(n);      % p is n-by-n symmetric matrix decision variable
y=symmetric;         % y is a scalar decision variable
p>0;                 % define the matrix inequalities
p<eye(n);
p*A+A'*p<y*II(n);
lmi_mincx_tbx(y);    % call the SDP optimization engine
P=value(p);          % get value of the optimal p
```