Ann Oper Res (2011) 189:343–356
DOI 10.1007/s10479-009-0585-5

Randomized methods based on new Monte Carlo schemes for control and optimization

Boris T. Polyak · Elena N. Gryazina

Published online: 15 July 2009

Abstract We address randomized methods for control and optimization based on generating points uniformly distributed in a set. For control systems this sets are either stability domain in the space of feedback controllers, or quadratic stability domain, or robust stability domain, or level set for a performance specification. By generating random points in the prescribed set one can optimize some additional performance index. To implement such approach we exploit two modern Monte Carlo schemes for generating points which are approximately uniformly distributed in a given convex set. Both methods use boundary oracle to find an intersection of a ray and the set. The first method is Hit-and-Run, the second is sometimes called Shake-and-Bake. We estimate the rate of convergence for such methods and demonstrate the link with the center of gravity method. Numerical simulation results look very promising.

Keywords Randomized algorithms · Monte Carlo · Optimization · Random search · Linear systems · Stabilization

1 Introduction

Recent years exhibited the growing interest to randomized algorithms in control and optimization; e.g., see Tempo et al. (2004). There are numerous reasons for such interest, the discussion can be found in Campi (2008). Historically, first random search methods for optimization were proposed in 1960-th (Rastrigin 1968), however rigorous analysis (Nemirovski and Yudin 1983) demonstrated that optimistic hopes on their effectiveness for global optimization were exaggerated. Nevertheless now one can see the revival of randomized approaches for optimization. Present paper proposes modern Monte Carlo schemes

B.T. Polyak (✉) · E.N. Gryazina
Institute of Control Sciences, Russian Academy of Sciences, 65 Profsoyuznaya st., Moscow 117997, Russia
e-mail: boris@ipu.ru

E.N. Gryazina
e-mail: gryazina@ipu.ru

(so-called Markov Chain Monte Carlo (MCMC); e.g., see Rubinstein and Kroese 2008; Gilks et al. 1996; Diaconis 2009) for convex optimization and control problems.

Up to now randomized algorithms in optimization are mostly oriented on discrete optimization and NP-hard problems; e.g., see Rubinstein and Kroese (2008, 2004), Diaconis (2009), Mitzenmacher and Upfal (2005). There are few publications related to convex case (Bertsimas and Vempala 2004). On the contrary, in control field most efforts were directed on convex structure of the problem; this is why in control problems quadratic stability is used instead of stability, quadratic robust stability instead of robust stability etc. However it remains a challenging problem to deal with basic notions (such as stability) in spite of nonconvexity of the domains under consideration. It seems that so called *Hit-and-Run* (*HR*) *method* provides an useful opportunity to achieve this goal. The method was originally proposed in Turchin (1971) and discussed in details in Smith (1984), it is a version of Monte Carlo method to generate points which are approximately uniformly distributed in a given set. Its properties are discussed in Lovasz (1999) while its accelerated versions are proposed in Kaufman and Smith (1998). One of the pioneering works in the field of convex optimization is due to Bertsimas and Vempala (2004) where Hit-and-Run method was used. Surprisingly, up to our knowledge it has not been exploited in control applications. We guess that HR is the promising tool for stabilization and optimization of linear systems. It allows generating random points inside the stability domain or inside performance specification domain in the space of gain matrices for feedback. Thus we can, for instance, generate stabilizing controllers of the fixed structure and optimize some performance index. The only assumption is that one admissible controller is available. Another useful example of MCMC is so-called *Shake-and-Bake* (*SB*) *method*. It has been developed in Borovkov (1991) (see also Borovkov 1994) and became a useful technique in physics (Comets et al. 2006). This method can be also exploited for optimization and control problems.

The structure of the paper is as follows. In Sect. 2 the optimization problem is formulated. For convex case the cutting plane method based on uniformly generated points in the set is presented and the main result on the expected rate of convergence is given. Section 3 is devoted to implementation of the "ideal" Monte Carlo. We consider Markov-chain Monte Carlo schemes for generating samples *asymptotically* uniformly distributed in a bounded set. We describe *boundary oracle* which is needed for the implementation of the technique. Boundary oracle can be found either in explicit form or it can be constructed numerically. Two generating schemes (Hit-and-Run and Shake-and-Bake) are discussed. We also provide some examples of their behavior in optimization problems. Section 4 contains the general scheme of HR method applied to control problems. It describes boundary oracle for several sets arising in control. Subsection 4.1 treats stabilization of SISO or MIMO systems. HR method allows solving such hard problems as stabilization via static output feedback (provided that one stabilizing controller is given). Next Subsect. 4.2 is devoted mostly to convex case (robust quadratic stabilization problems). Section 5 gives some conclusive remarks.

2 Optimization: problem formulation and "ideal" Monte Carlo

We consider the problem

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & x \in X \end{array} \tag{1}$$

where X is a convex bounded closed set in $\mathbb{R}^n$ with nonempty interior. Of course an arbitrary convex optimization problem can be converted into format (1). For instance, if the original problem is min $f(x)$ s.t. $x \in Q$, Q and f are convex, then we introduce a slack variable t and proceed to

$$\begin{array}{ll} \min & t \\ \text{s.t.} & x \in Q, \qquad f(x) - t \le 0. \end{array}$$

There exist powerful deterministic methods for convex optimization such as interior-point algorithms (Ben-Tal and Nemirovski 2001; Nesterov and Nemirovsky 1994); they are proved to be polynomial-time and very efficient in practical computations. We suggest random algorithms that are quite efficient in some cases. Suppose that we can generate a sample of N independent uniformly distributed points in X and in the convex sets X_k arising in the process of calculations. This is very strong assumption, availability of such generator is an exception. Of course, we can always apply rejection method: take a simple set (an ellipsoid or a box) containing X_k, generate points uniformly in this set and reject those points which are not in X_k. However the proportion of rejected points is in general too large for high-dimensional problems. In the next sections we provide implementable alternatives for this approach.

The *cutting plane method* based on a uniform generator looks as follows.

1. Set $k = 1$, $X_1 = X$.
2. Generate N points $x^1, x^2, \dots x^N$ independently uniformly distributed in X_k.
3. Find $f_k = \min_{i=1,\dots,N} c^T x^i$.
4. Set $X_{k+1} = X_k \cap \{x : c^T x \le f_k\}$ and proceed to Step 2.

The main result on the expected rate of convergence of the algorithm reads as follows. Denote $f^* = \max_{x \in X} c^T x$, $f_* = \min_{x \in X} c^T x$, $h = f^* - f_*$; $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1} dt$ is Euler beta-function (see `beta(a,b)` command in MATLAB).

Theorem 1 *After k iterations of the algorithm*

$$E\left[f_k\right] - f_* \le h q^k, \quad q = \frac{1}{n} B\left(N+1, \frac{1}{n}\right) \le \left(\frac{1}{N+1}\right)^{\frac{1}{n}}. \tag{2}$$

Thus the algorithm converges (in mean) with geometric rate.

The proof of the theorem can be found in Dabbene et al. (2008); it has much in common with the proof of the relating result in Bertsimas and Vempala (2004) and exploits Brunn-Minkowski inequality. It is shown that the worst-case body X is a cone (with c being a normal to its base) and the estimate is sharp for such X. The case of $N = 1$, $k = 1$ is of special interest.

Theorem 2 *Let x^1 be a random point uniformly distributed in X. Then*

$$E\left[c^T x^1\right] - f_* \le h\left(1 - \frac{1}{n+1}\right).$$

Having in mind that $E[x^1] = g$ (center of gravity of X) and $B(2, \frac{1}{n}) = \frac{n^2}{n+1}$ we get that for arbitrary c one has $c^T g - f_* \le h(1 - \frac{1}{n+1})$. This is the famous Radon theorem (Radon 1916), thus Theorem 1 is its extension. The deterministic version of random algorithm above is *center of gravity* method: take $x^k = g^k$ (center of gravity of X_k) and set $X_{k+1} = X_k \cap \{x : c^T x \le c^T g^k\}$. Similar method has been proposed in Levin (1965), Newman (1965) for optimization problem formulated in the form min $f(x)$ s.t. $x \in B$, where f is a convex function and B is a ball. The cutting plane for construction of X_{k+1} is given by $\nabla f(x^k)^T(x - x^k) \leq 0$ where $\nabla f(x)$ is a subgradient of f at x and x^k is the center of gravity of X_k. The proof is based on Grunbaum theorem for volumes of subsets cut by the hyperplane. It is interesting to note that the estimate based on Radon theorem is better.

Theorem 1 provides results on expected convergence of the method. Estimates on convergence with high probability can be also obtained (Dabbene et al. 2008).

As we have mentioned, uniform sampling is not available in general situations, and at the first glance the value of the relating results (like Theorem 1) is minor. However these results are of interest when we check how close is the implemented distribution to the uniform one. Such tests will be used in the next section.

3 Implementable random algorithms: boundary oracle

For implementation of the "ideal" Monte Carlo method we need a mechanism for generating uniform random samples from X. In this section we describe Markov Chain Monte Carlo schemes for generating samples *asymptotically* uniformly distributed in a bounded closed set $X \in \mathbb{R}^n$. Suppose we have a starting point $x^0 \in X$. We call *boundary oracle* an algorithm which provides $L = \{t \in \mathbb{R} : x^0 + td \in X\}$, where d is a vector specifying the direction in $\mathbb{R}^n$. In the simplest case, when X is convex, this set is the closed interval $[\underline{t}, \overline{t}]$, where $\underline{t} = \inf\{x^0 + td \in X\}$, $\overline{t} = \sup\{x^0 + td \in X\}$. In more general situations boundary oracle provides all intersections of the straight line $x^0 + td, -\infty < t < +\infty$ with X. We also denote *complete boundary oracle* a boundary oracle algorithm that provides also an internal normal vectors to X at boundary points. Boundary oracle is available for numerous specific sets X. For linear matrix inequalities (LMI) (see Boyd et al. 1994) set

$$X = \left\{ x \in \mathbb{R}^n : A_0 + \sum_{i=1}^{n} x_i A_i \preceq 0 \right\} \tag{3}$$

(A_i are symmetric matrices of a certain size for all i, $A \preceq 0$ means that A is negative semidefinite) to derive a semidefinite boundary oracle we exploit the following result for $A = A_0 + \sum_{i=1}^n x_i^0 A_i$, $B = \sum_{i=1}^n d_i A_i$.

Lemma 1 (Polyak and Shcherbakov 2006b) *Let $A \prec 0$ and $B = B^T$. Then the matrix $A + tB$ is negative definite for $t \in (\underline{t}, \overline{t})$:*

$$\underline{t} = \begin{cases} \max_{t_i<0} t_i, & \\ -\infty, & \textit{if all } t_i > 0, \end{cases} \qquad \overline{t} = \begin{cases} \min_{t_i>0} t_i, & \\ +\infty, & \textit{if all } t_i < 0 \end{cases}$$

where t_i are the generalized eigenvalues of the matrix pencil $(A, -B)$, i.e., $Ae_i = -t_i B e_i$. For $t \notin (\underline{t}, \overline{t})$ the matrix $A + tB$ loses negative definiteness.

Another LMI constrained set is the set of symmetric matrices P defined by Lyapunov inequality:

$$X = \{P : AP + PA^T + C \preceq 0,\ P \succeq 0\} \tag{4}$$

where A is a stable matrix and $C \succ 0$. This set is always convex, and boundary oracle can be found explicitly. Indeed, take $P_0 \in X$ and generate $D = D^T$—a matrix specifying the direction. Then $A(P_0+tD)+(P_0+tD)A^T+C \preceq 0 \Leftrightarrow F+tG \prec 0$, $F = AP_0+P_0A^T+C$, $G = AD+DA^T$. For this case $L=(\underline{t},\overline{t})$ and $\overline{t}=\min\lambda_i$, $\underline{t}=\min\mu_i$, where λ_i are positive real eigenvalues of matrix pencil $F,-G$, while μ_i are positive real eigenvalues of matrix pencil F,G.

Boundary oracle for quadratic matrix inequalities (QMI) sets

$$X=\left\{P : AP+PA^T+PBB^TP+C \preceq 0,\ P \succeq 0\right\} \tag{5}$$

can be obtained similarly.

For the sets given by linear algebraic inequalities

$$X=\left\{x\in\mathbb{R}^n : c_i^T x \le a_i,\ i=1,\dots,m\right\} \tag{6}$$

the boundary oracle for x^0+td is $[\underline{t},\overline{t}]$,

$$\underline{t}=\min_{i:\, c_i^T d>0}\frac{a_i-c_i^Tx^0}{c_i^Td},\qquad \overline{t}=\max_{i:\, c_i^T d>0}\frac{a_i-c_i^Tx^0}{c_i^Td}.$$

3.1 Hit-and-Run

We start with presenting the idea and results relating to HR method in general setting. Suppose there is a bounded set $X\in\mathbb{R}^n$ (in general *nonconvex* and *not simply connected*) and a point $x^0\in X$. In every step we choose a random vector d uniformly distributed on the unit sphere in $\mathbb{R}^n$. HR method generates points in X as follows:

$x^1=x^0+t_1d$,	t_1 is uniformly distributed on L given by the boundary oracle.
Then x^0	is replaced with x^1, L is updated with respect to x^1 and so on.

The simplest theoretical result on the behavior of HR method states that if X does not contain lower dimensional parts, then the method achieves the neighborhood of any point of X with nonzero probability and asymptotically the distribution of points x^i tends to uniform one.

Theorem 3 (Smith 1984) *Suppose X coincides with the closure of interior points of X. Then for any measurable set $A\subset X$ the probability $P_i(A)=P(x^i\in A|x^0)$ can be estimated as $|P_i(A)-P(A)|\le q^i$, where $P(A)=\mathrm{Vol}(A)/\mathrm{Vol}(X)$, $q<1$.*

Unfortunately q strongly depends on geometry of X and dimension n and can be close enough to 1. Tighter bounds for the rate of convergence for convex X can be found in Lovasz (1999), Hit-and-Run modifications for accelerating the rate of convergence are described in Kaufman and Smith (1998). The behavior of Hit-and-Run method depends also on the starting point. Let $X=[0,1]^n\subset\mathbb{R}^n$ and $x^0=0$. In this case with high probability (equal $1-2^{-n}$) some components of d have different signs, and $x^1=x^0$.

We examine the method by the comparison of the obtained value $\frac{\min f_i-f_*}{h}$ (or $\frac{f^*-\max f_i}{h}$) (here $f_i=c^Tx^i$, x^i are generated by Hit-and-Run) with the theoretical estimate of this value given by Theorem 1 (valid for uniform distribution). Consider the standard SDP (semidefinite programming) problem

$$\begin{array}{ll}\min & c^Tx\\ \text{s.t.} & A_0+\sum_{i=1}^n x_iA_i\preceq 0,\end{array} \tag{7}$$

Table 1 Comparison of estimates for HR and uniform distribution—LMI set

n	N	$\frac{f^* - \max f_i}{h}$	$\frac{\min f_i - f_*}{h}$	$\frac{1}{n} B(N+1, \frac{1}{n})$
2	100	0.0489	0.0290	0.0883
2	1000	0.0079	0.0043	0.0280
2	5000	0.0035	0.0025	0.0125
2	10000	0.0013	0.0014	0.0089
10	100	0.3682	0.4524	0.5999
10	1000	0.2548	0.3892	0.4768
10	5000	0.2262	0.3602	0.4059
10	10000	0.2085	0.3524	0.3787

where A_i are randomly generated such that

(i) the feasible set has non-empty interior (for simplicity, take $A_0 \prec 0$);
(ii) the feasible set is bounded.

To satisfy the latter condition we generate A_i, $i = 1, \dots, n$ as follows:

$$M = 2\mathrm{rand}(m/2) - 1, \qquad M = M + M^T, \qquad A_i = \mathrm{blkdiag}(M, -M),$$

A_i is a block-diagonal matrix.

We generate N points for various dimension n via Hit-and-Run method and take empirical expectation of minimal and maximal function values. Exact minimal and maximal function values f_* and f^* are obtained by standard SDP solver Yalmip (Lofberg 2004). The results are presented in Table 1.

Hit-and-Run points give better expectation of minimum and maximum since the feasible set for randomly generated A_i is usually well-conditioned (we remind that the estimates of Theorem 1 are sharp for cone-shape sets).

For the case of the worst geometry we consider the following problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \|x\|_1 \le 1, \\ & x_i \ge 0, \\ & c = [1, \dots, 1]. \end{aligned} \tag{8}$$

The feasible set is a simplex and the averaged minimal function value should be in accordance with Theorem 1. The obtained result are shown in Table 2 (note that $f_* = 0$).

We observe that the expected minimum for Hit-and-Run points is approximately the same as the theoretical expectation and it becomes worse with the growth of dimension. The situation dramatically changes for ill-conditioned sets. For instance, taking $c = [1, \dots, 1, 10^4]$ in (8) we find out that Hit-and-Run points concentrate in a very narrow region even in $\mathbb{R}^2$, see Fig. 1.

Having generated the sample $x^1, x^2, \dots, x^N \in X_k$ coming back to optimization problem (1) we can apply cutting plane method described in Sect. 2. To avoid situations like above we need a "good" interior starting point (warm-start). For this purpose the algorithm will be slightly modified. Instead of cutting with $(c, x) \le f_k = \min(c, x^i)$ we take $X_{k+1} = X_k \cap \{(c, x) \le \tilde{\varphi}\}$, $\tilde{\varphi}$ being 10% quantile of $\varphi_i = (c, x^i)$, $i = 1, \dots, N$. The average of the remaining 10% points with $\varphi_i \le \tilde{\varphi}$ is exploited as the initial point for Hit-and-Run.

Table 2 Comparison of estimates for HR and uniform distribution—simplex set

n	N	$\min f_i$	$\frac{1}{n}B\left(N+1,\frac{1}{n}\right)$
2	100	0.0916	0.0883
2	1000	0.0318	0.0280
2	5000	0.0149	0.0125
2	10000	0.0147	0.0089
3	100	0.2405	0.192
3	1000	0.1341	0.0893
3	5000	0.0447	0.0522
3	10000	0.0461	0.0414
10	100	0.7449	0.5999
10	1000	0.6132	0.4768
10	5000	0.4693	0.4059
10	10000	0.4584	0.3787

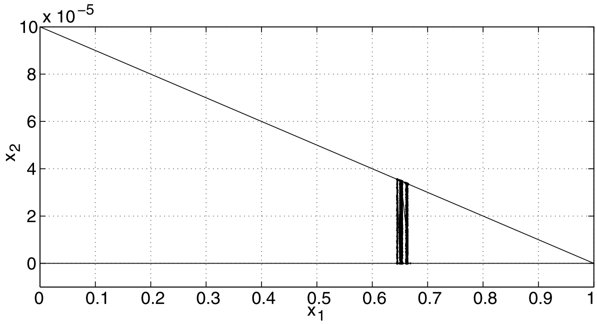


Fig. 1 Hit-and-Run fails for ill-conditioned set in $\mathbb{R}^2$

The methods were tested over a range of SDP problems (7) with randomly generated data. The problems were solved via cutting plane method using HR samples. The discussion of the results can be found in Polyak and Shcherbakov (2006a). We applied modified HR where $\min x_i$ was replaced with averaged X_i and various heuristic acceleration methods (scaling, projecting, accelerating step) were exploited.

3.2 Shake-and-Bake

An alternative way is to generate asymptotically uniformly distributed samples in the boundary of the set $X \in \mathbb{R}^n$. In some cases these samples may cause better estimate of the convergence rate. Shake-and-Bake algorithm is proposed in Borovkov (1991, 1994) in order to generate random vectors in a connected domain with smooth boundary or on the boundary itself. SB was exploited for studying the stochastic billiards with the cosine law of reflection (Comets et al. 2006).

Suppose x^0 is a boundary point of X and n^0 is the unit internal normal vector for ∂X at the point x^0. Since the set X is assumed to be piece-wise linear the probability to reach a boundary point with a unique internal normal is one. SB method generates points in ∂X as follows:

$$\begin{array}{ll} x^1 = x^0 + \bar{t}d, & \bar{t} \text{ is given by the boundary oracle in the} \\ & \text{direction } d,\ \|d\| = 1, \\ d = gn^0 + r, & g = \sqrt{1 - \xi^{\frac{2}{n-1}}},\ \xi \text{ is uniform random variable in } [0, 1], \\ & r \text{ is random uniform direction } \|r\| = 1,\ (n^0, r) = 0. \end{array}$$

Then x^0 is replaced with x^1, normal n^1 is calculated and so on. The special choice of g is due to desirable cosine law of reflection that guarantees asymptotically uniform distribution of samples x^i on the boundary of X, asymptotically uniform samples on X can be obtained taking uniform random points in the chord $[x^{i-1}, x^i]$.

For the implementation of SB algorithm we need the complete boundary oracle that provides an internal normal vector at the boundary points besides the intersection of the line and the set X. For the set (3) internal normal at the point $x^0 \in \partial X$ is a vector n with components $n_i = -(A_i e, e)$, where e is the eigenvector corresponding to zero eigenvalue of the matrix $A_0 + \sum_{i=1}^n x_i^0 A_i$ provided that multiplicity of the zero eigenvalue is one. In more general case, we describe a cone of admissible directions $K = \{d : (d, n^k) \geq 0,\ k = 1, \ldots, m\}$ where vectors n^k with components $n_i^k = -(A_i e^k, e^k)$ are formed by different eigenvalues e^i corresponding to zero eigenvalue of multiplicity m (or to eigenvalues close to zero).

For the Lyapunov inequality set (4) the normal at the point P_0 is given by the matrix

$$N = -(ee^T A - A^T ee^T) \tag{9}$$

where e is the eigenvector corresponding to zero eigenvalue of the matrix $AP_0 + P_0 A^T + C$. Since the zero eigenvalue has multiplicity m and there are m different eigenvalues $e^1, \ldots, e^m$ a cone of admissible directions is given by $K = \{D : \langle D, N^i \rangle \geq 0\}$, $N^i = -(e^i(e^i)^T A - A^T e^i (e^i)^T)$, inner product of symmetric matrices $\langle A, B \rangle = \mathrm{tr}(AB)$.

For the set (6) internal normal coincides with vector c_i since the boundary point is at the i-th equality.

SB can be extended for sets with nonsmooth boundary. Then a normal is not available for arbitrary point but there is a set of vectors that produce the admissible directions cone and we choose a uniform random direction d in this cone. The example of points generated by SB for a nonconvex set with nonsmooth boundary are depicted in Fig. 2.

4 Applications to control

In control applications the set X is the set of design variables (e.g., controller parameters or uncertainties). It is the admissible set with respect to some specifications (e.g., the set of stabilizing controllers) and the admissible points are most often denoted by k. We keep the notation as $k \in X$ throughout this section.

We provide boundary oracle for several sets arising in control applications.

1. Stability set for polynomials. Consider the affine family of polynomials

$$P(s, k) = P_0(s) + \sum_{i=1}^n k_i P_i(s) \tag{10}$$

where $P_i(s)$ are m-th degree polynomials. The polynomial $P(s)$ is stable (Hurwitz) when all its roots have negative real parts. Define the set X in the space of parameters $k = (k_1, \ldots, k_n)$

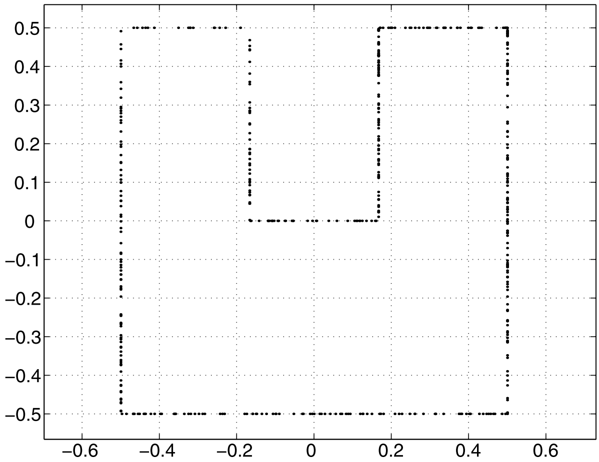


Fig. 2 Shake-and-Bake method for a nonconvex set with nonsmooth boundary

which corresponds to stable polynomials:

$$X = \{k : P(s,k) \text{ is Hurwitz}\}. \tag{11}$$

The geometry of such sets and of their boundaries is well studied, see Polyak and Shcherbakov (2006a). HR method looks as follows. We assume that a stable polynomial $P(s,k^0)$ is given. Then we generate random $d \in \mathbb{R}^n$ uniformly distributed on the unit sphere and take $P(s,k^0+td) = A(s) + tB(s)$, $A(s) = P(s,k^0)$, $B(s) = \sum_{i=1}^n d_i P_i(s)$. The explicit algorithm for finding $L = \{t \in \mathbb{R} : A(s) + tB(s) \text{ is Hurwitz}\}$ is available, see Theorem 2 and Algorithm 1 in Gryazina and Polyak (2006). In general L consists of not more than $m/2+1$ intervals.

2. Stability set for matrices. For a family of matrices $A + BKC$, where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$ are given and $K \in \mathbb{R}^{m\times l}$ is a variable (which represents either uncertainty or control gain) we can distinguish the set of stabilizing gains:

$$X = \{K : A + BKC \text{ is Hurwitz}\}, \tag{12}$$

i.e. all eigen values of $A + BKC$ have negative real parts.

The structure of this set is analyzed in Gryazina and Polyak (2006). It can be nonconvex and can consist of many disjoint domains. To construct the boundary oracle we generate matrix $D = Y/\|Y\|$, $Y =$ `randn(m,l)` which is uniformly distributed on the unit sphere in the space of matrices equipped with Frobenius norm. Then we get straight line $A + B(K^0 + tD)C = F + tG$, $F = A + BK^0C$, $G = BDC$ for a matrix $K^0 \in X$. Then $L = \{t \in \mathbb{R} : F + tG \text{ is Hurwitz}\}$. L consists of finite number of intervals, the algorithm for calculating their end points is presented in Gryazina and Polyak (2006), Sect. 4. However sometimes "brute force" approach is more simple. Introduce $f(t) = \max \Re\, \mathrm{eig}(F + tG)$, then the end points of the intervals are solutions of the equation $f(t) = 0$ and can be found by use of standard 1D equation solvers (such as command `fsolve` in Matlab).

3. Robust stability set. For the affine family of polynomials with uncertain parameters $q \in Q$ this set is defined as

$$X = \left\{k : P_0(s,q) + \sum_{i=1}^n k_i P_i(s,q) \text{ is Hurwitz for all } q \in Q\right\}. \tag{13}$$

If Q is a finite set $\{q_1, \dots, q_m\}$ and m is small, the set X is the intersection of m sets corresponding to m uncertainties q_i, thus the boundary oracle is the intersection of corresponding boundary oracles: $L = \bigcap L_i$. There are also some other cases, when L can be calculated explicitly, for instance $p_i(s, q)$ being interval polynomials. However in more general situations we apply different approach working with robust stability problems (see Subsect. 4.2 below).

4. Quadratic stability set. This set is defined as solution of some LMIs. The typical example is the set of symmetric matrices P defined by Lyapunov inequality (4).

4.1 Stabilization

We assume starting point $k^0 \in X$ to be known to demonstrate the applications of the HR algorithm.

1. Consider linear time-invariant single-input single-output plant $G(s) = \frac{a(s)}{b(s)}$ where $a(s), b(s)$ are given polynomials of order m. We wish to stabilize it with low order controller $C(s) = \frac{f(s)}{g(s)}$ where polynomials $f(s), g(s)$ have fixed orders (for instance, it can be PID-controller). We assume that one stabilizing controller $C^0(s) = f^0(s)/g^0(s)$ is known.

The closed-loop characteristic polynomial is

$$P(s) = a(s)f(s) + b(s)g(s). \tag{14}$$

If we treat the coefficients of the polynomials $f(s), g(s)$ as parameters k, we are at the setup of (10).

Example 1 (Fujisaki et al. 2008) Given a plant

$$P(s) = \frac{17(s+1)(16s+1)(s^2-s+1)}{s(-s+1)(-s+90)(4s^2+s+1)}$$

and a fixed order controller $C(s)$ of the form

$$C(s) = \frac{k_1 + k_2 s + k_3 s^2}{k_4 + k_5 s + k_6 s^2}.$$

The problem is to find controller parameters that guarantee $\|W(s)S(s)\|_\infty < 1$ where $S(s) = \frac{1}{1+C(s)P(s)}$ is a sensitivity transfer function and $W(s) = \frac{55(1+3s)}{1+800s}$ is a weighted function, which is usually chosen from engineering specifications. Starting with a controller found in Gryazina and Polyak (2006)

$$C^0(s) = \frac{-0.532 - 0.5407s - 2.0868s^2}{1 - 0.3645s - 1.2592s^2} \tag{15}$$

we restrict controller parameters k to stay in 0.1-box neighborhood of the original parameter values and generate 1000 stabilizing controllers via Hit-and-Run method. Then for each controller we calculate $\|W(s)S(s)\|_\infty$, for 217 points it appears to be less than one. Finally, we choose the best controller

$$C^*(s) = \frac{-0.537 - 0.5743s - 2.1114s^2}{1 - 0.3025s - 1.2128s^2}$$

that leads to $\|W(s)S(s)\|_\infty = 0.8206$ compared to 0.9822 for controller (15). So here Hit-and-Run allows performing local improvement.

2. Proceed to static output feedback stabilization for uncertain multi-input multi-output plant:

$$\dot{x} = A(q)x + B(q)u, \qquad y = C(q)x, \qquad u = Ky, \tag{16}$$

the objective is to find robustly stabilizing gains K provided we know one of them.

Example 2 Here

$$A = \begin{bmatrix} -0.0366 & 0.271 & 0.0188 & -0.4555 \\ 0.0482 & -1.01 & 0.0024 & -4.0208 \\ 0.1002 & q_1 & -0.707 & q_2 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.4422 & 0.1761 \\ q_3 & -7.5922 \\ -5.52 & 4.49 \\ 0 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix},$$

$q \in Q_\rho = \{q : |q_i - q_i^0| \le \rho\gamma_i\}$, $q^0 = [0.3681, 1.42, 3.5446]$; $\gamma = [0.05, 0.01, 0.04]$. The original problem here is to find a controller robustly stabilizing the closed-loop system with $\rho = 1$ and a decay rate of at least $\alpha = 0.1$. This problem arises in control of helicopters: (Singh and Coelho 1984) and it was studied in Bhattacharyya (1987), El Ghaoui et al. (1997), Tempo et al. (2004).

We apply our technique that allows finding better controller robustly stabilizing the system with a wider uncertainty range and, perhaps, a larger decay rate.

The first step is to generate controllers stabilizing the nominal system, i.e., with $q = q^0$. The closed-loop system matrix is $A_c = A + BKC$ and we also can apply HR method tailored for this problem. Starting with the stabilizing controller $K = [-0.4357;\ 9.5652]$ (see El Ghaoui et al. 1997) we generate 1000 points that belong to the intersection of the stability domain and the bounding box $\|K\|_\infty \le 100$.

Then we select a controller that guarantee a decay rate $\alpha = 0.1$, there are 187 controllers among 1000 that satisfy this requirement. Taking for the nominal matrix $A_0 = A + \alpha I$ and the selected controller as a starting point we generate 1000 controllers for the required α. Figure 3 shows that these controllers correspond to a segment (where the density of points is higher) among those generated in the first step. Boundary points are naturally obtained in HR procedure and they are also depicted.

Then we take into consideration the uncertainty with enlarged uncertainty intervals, i.e., $\rho > 1$. For each controller that guarantees a decay rate $\alpha = 0.1$ we check if it stabilizes 1000 random points uniformly generated in the box Q_ρ. For $\rho = 40$ (i.e. 40 times larger than original intervals) we still can find several suitable controllers. Their parameters are situated in the middle of the segment. Take, for instance, $K = [7.1096;\ 57.6346]$. Straightforward validation shows that this controller is indeed robustly stabilizing.

4.2 Robust quadratic stabilization

The general setup has been described earlier. We illustrate how this technique works for one example.

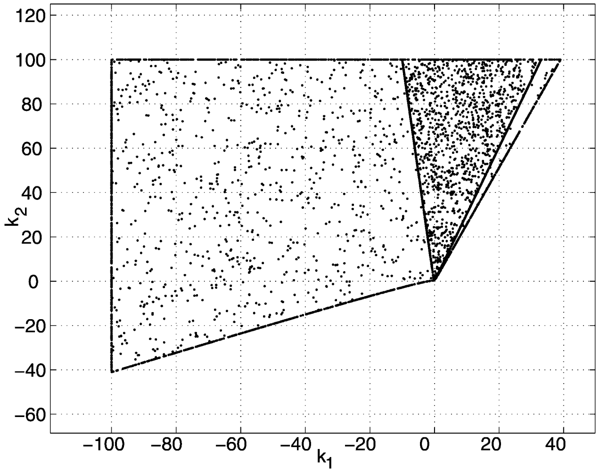


Fig. 3 Stabilizing controller parameters for nominal system

Example 3 Here we investigate the example originated in Barmish (1985). Consider a system with uncertainty (16) with

$$A = \begin{bmatrix} q_1 & 1 \\ 0 & q_1 \end{bmatrix}, \qquad B = \begin{bmatrix} q_2 \\ 1 \end{bmatrix}, \qquad q \in Q_\rho = \{q : |q_i| \le \rho, \rho = 0.5\}.$$

For the problem of quadratic robust stabilization in Barmish (1985) a very complicated nonlinear control is suggested. We strive to find a linear control $K = [k_1; k_2]$ solving the same problem.

The stability domain for the nominal system ($q_i = 0$, $i = 1, 2$) can be easily found: $k_1 < 0$, $k_2 < 0$. First we generate controllers quadratically stabilizing the nominal system, i.e. such K that for $A_c = A + BK$ there exist $P > 0$: $A_c^T P + P A_c < 0$. Multiplying by $Q = P^{-1}$ we have LMI in Q and Y:

$$Q > 0, \quad QA^T + AQ + BY + Y^T B^T < 0, \quad Y = KQ.$$

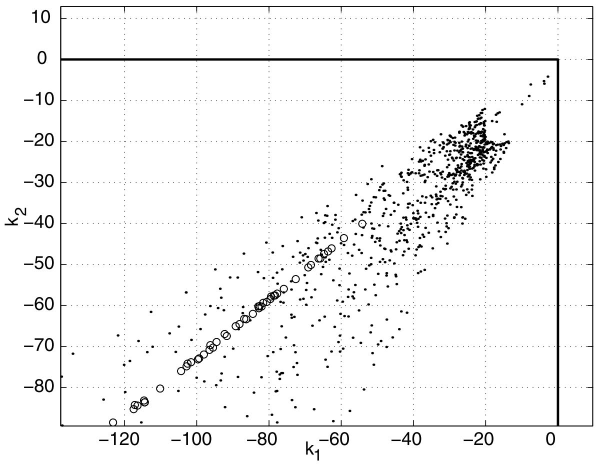
For a starting point we take feasible solution of LMI using YALMIP (Lofberg 2004). HR allows generating any number of feasible points (and correspondingly controller parameters).

Then there are two ways to deal with uncertainty. First is straightforward checking robust quadratic stabilization for each controller that quadratically stabilized the nominal system by generating required number of uncertain samples. This approach can give a probabilistic solution. Another approach is applicable when it is sufficient to check feasibility of a certain (not very large) number of LMIs corresponding to uncertain bounds. In this example it is sufficient to check quadratic stabilizability of 4 vertex samples. In this case HR is applicable taking

$$X = \bigcap_i \{Q > 0,\ QA_i^T + A_i Q + B_i Y + Y^T B_i^T < 0\},$$

where index i corresponds to the vertex sample. For generating quadratic robust stabilizing controllers the boundary oracle for the set (4) is exploited taking $Q = Q_0 + J$, $Y = Y_0 + G$

Fig. 4 Robust quadratic stabilizing controllers



and $F = Q_0 A_i^T + A_i Q_0 + B_i Y_0 + Y_0^T B_i^T$, $R = J A_i^T + A_i J + B_i G + G^T B_i^T$, where matrix J and vector G specify random direction in a corresponding space.

Note that this points are asymptotically uniform in the space of Q, Y matrices but not in the space of controller parameters $K = Y Q^{-1}$. Figure 4 depicts robust stabilizing controllers for the original uncertain set with $\rho = 0.5$ (points).

Now we want to increase ρ. For $\rho = 0.8$ there are no quadratic robust stabilizing controllers but for $\rho = 0.7$ their parameters are marked with "○" in Fig. 4. Note that the absolute parameter values are greater than that for $\rho = 0.5$.

The controller parameters may happen to be large enough but we can also deal with box-restrictions for controller parameters, e.g. $\|K\|_\infty \le \gamma$ with starting point K^0 satisfying this condition. Another natural box-restriction is $\|K - K^0\|_\infty \le \gamma$ as it was used in Example 1. For every Hit-and-Run step we solve one-dimensional problem of boundary oracle in t, for every feasible point ($t = 0$) the restriction holds. Then find the closest to zero positive and negative t such that

$$\|K(t)\|_\infty - \gamma = \|Y(t) Q(t)^{-1}\|_\infty - \gamma = 0,$$

where $Y(t) = Y_0 + tG$, $Q(t) = Q_0 + tJ$. These t should be treated as additional candidates for $\underline{t}$ and $\overline{t}$ of the boundary oracle in HR algorithm.

5 Conclusions

The randomized methods (like HR and SB) have serious advantages. They are applicable for numerous control and optimization problems, are simple in implementation (because boundary oracles are available in explicit form), generated points give a good representation of the feasible set. The first results of numerical simulation look promising.

However the distribution of points generated by the proposed methods strongly depends on the geometry of the set. It is far from uniform for narrow-shaped bodies and high-dimensional sets. We plan further research on computational schemes of the algorithms for these cases.

References

Barmish, B. (1985). Necessary and sufficient condition for quadratic stabilizability of an uncertain system. *Journal of Optimization Theory and Applications*, *46*(4), 399–408.

Ben-Tal, A., & Nemirovski, A. (2001). *Lectures on modern convex optimization: analysis, algorithms and engineering applications*. SIAM: Philadelphia.

Bertsimas, D., & Vempala, S. (2004). Solving convex problems by random walks. *Journal of the ACM*, *51*, 540–556.

Bhattacharyya, S. (1987). *Robust stabilization against structured perturbations*. New York: Springer.

Borovkov, K. A. (1991). On a new variant of the Monte Carlo method. *Theory of Probability and its Applications*, *36*(2), 355–360.

Borovkov, K. A. (1994). On simulation of random vectors with given densities in regions and on their boundaries. *Journal of Applied Probability*, *31*(1), 205–220.

Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. SIAM: Philadelphia.

Campi, M. (2008). *Randomized algorithms in systems and control—Why resorting to the fate is wise*? (Manuscript).

Comets, F., Popov, S., Schutz, G. M., & Vachkovskaia, M. (2006). *Billiards in a general domain with random reflections*. arXiv:math/0612799v3 [math.PR].

Dabbene, F., Shcherbakov, P., & Polyak, B. T. (2008). A randomized cutting plane method with probabilistic geometric convergence. In *Proc. of the 47th CDC* (pp. 3044–3049). Mexico: Cancun.

Diaconis, P. (2009). The Markov chain Monte Carlo revolution. *American Mathematical Society. Bulletin*, *46*(2), 179–205.

El Ghaoui, Oustry, F., & AitRami, M. (1997). A cone complementary linearization algorithm for static output-feedback and related problems. *IEEE Transactions on Automatic Control*, *AC-42*, 1171–1176.

Fujisaki, Y., Oishi, Y., & Tempo, R. (2008). Mixed deterministic/randomized methods for fixed order controller design. *IEEE Transactions on Automatic Control*, *AC-53*, 2033–2047.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov chain Monte Carlo*. London: Chapman and Hall.

Gryazina, E., & Polyak, B. (2006). Stability regions in the parameter space: D-decomposition revisited. *Automatica*, *42*, 13–26.

Kaufman, D. E., & Smith, R. L. (1998). Direction choice for accelerated convergence in Hit-and-Run sampling. *Operations Research*, *46*(1), 84–95.

Levin, A. Y. (1965). On an algorithm for the minimization of convex functions over convex sets. *Soviet Mathematics Doklady*, *160*(6), 286–290.

Lofberg, J. (2004). YALMIP: A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD conference*. http://control.ee.ethz.ch/~joloef/yalmip.php.

Lovasz, L. (1999). Hit-and-Run mixes fast. *Mathematical Programming*, *86*, 443–461.

Mitzenmacher, M., & Upfal, E. (2005). *Probability and computing: randomized algorithms and probabilistic analysis*. New York: Cambridge University Press.

Nemirovski, A., & Yudin, D. (1983). *Informational complexity and efficient methods for solution of convex extremal problems*. New York: Wiley.

Nesterov, Y., & Nemirovsky, A. (1994). *Interior point polynomial methods in convex programming*. SIAM: Philadelphia.

Newman, D. J. (1965). Location of the maximum on unimodal surfaces. *Journal of the ACM*, *12*, 395–398.

Polyak, B. T., & Shcherbakov, P. (2006a). Randomized algorithm for solving semidefinite programs. In *Stochastic optimization in informatics* (pp. 38–70). St-Peterburg State University (in Russian).

Polyak, B. T., & Shcherbakov, P. (2006b). The D-decomposition technique for linear matrix inequalities. *Automation and Remote Control*, *67*(11), 1847–1861.

Radon, J. (1916). Uber eine Erweiterung des Begriffs der konvexen Functionen, mit einer Anwendung auf die Theorei der konvexen Korper. *Sitzungsberichte der Wiener Akademie*, *125*, 241–258.

Rastrigin, L. A. (1968). *Statistical search method*. Moscow: Nauka (in Russian).

Rubinstein, R. Y., & Kroese, D. P. (2004). *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Berlin: Springer.

Rubinstein, R. Y., & Kroese, D. P. (2008). *Simulation and the Monte Carlo method*. New York: Wiley.

Singh, S., & Coelho, A. (1984). Nonlinear control of mismatched uncertain linear systems and application to control of aircraft. *Journal of Dynamic Systems, Measurement and Control*, *106*, 203–210.

Smith, R. L. (1984). Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, *32*(6), 1296–1308.

Tempo, R., Calafiore, G., & Dabbene, F. (2004). *Communications and control engineering series: Randomized algorithms for analysis and control of uncertain systems*. London: Springer.

Turchin, V. F. (1971). On computation of multi-dimensional integrals by the Monte-Carlo method. *Theory of Probability and its Applications*, *16*(4), 720–724.