Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Hit-and-Run: new design technique for stabilization, robustness and optimization of linear systems

B. T. Polyak E. N. Gryazina

Institute of Control Science,
Russian Academy of Sciences, Moscow, Russia
(e-mail: {boris,gryazina}@ ipu.ru)

Abstract: New randomized algorithms for stabilization and optimal control for linear systems are proposed. They are based on Hit-and-Run method, which allows generating random points in convex or nonconvex domains. These domains 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 domain one can optimize some additional performance index. The approach demonstrated its high efficiency for numerous classical examples of design problems.

Keywords: Linear systems, design, stabilization, optimal control, randomized methods, Hit-and-Run, robustness

1. INTRODUCTION

Randomized algorithms are now widely used effective tools for various analysis and design linear control problems, see Tempo et al. (2004). Up to now they are mostly oriented on convex structure of the problem; this is why 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. The method found numerous applications in numerical analysis, for instance in convex optimization, Bertsimas and Vempala (2004). 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.

The structure of the paper is as follows. In Section 2 we describe HR method and *boundary oracle* which is needed for its implementation. Boundary oracle can be found either in explicit form (e.g. for stability domain of SISO systems) or it can be constructed numerically. Section 3 contains the general scheme of HR method applied to control problems. Next Sections address various classes of such problems. Section 4 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 Section 5 is devoted mostly to convex case (robust quadratic stabilization problems). Uncertainty in the system can be given in various forms, including interval matrix uncertainty. All Sections contain examples, borrowed from the literature. We demonstrate the efficiency of the proposed approach if compared with known results.

2. HIT-AND-RUN AND BOUNDARY ORACLE

We start with presenting the idea and results relating to HR method in general setting. Suppose there is a bounded set $\mathcal{K} \subset \mathbb{R}^n$ and a point $k^0 \in \mathcal{K}$. In every step we choose a random vector d uniformly distributed on the unit sphere in $\mathbb{R}^n$. Such vectors can be easily generated by command $d = s/||s||, s =$ `randn(n,1)` in Matlab. We call *boundary oracle* an algorithm which provides $L = \{t \in R : k^0 + td \in \mathcal{K}\}$. In the simplest case, when $\mathcal{K}$ is convex, this set is the interval $(-\underline{t}, \bar{t})$ where $\bar{t} = \sup\{t : k^0 + td \in \mathcal{K}\}, \underline{t} = \sup\{t : k^0 - td \in \mathcal{K}$. In more general situations boundary oracle provides all intersections of the straight line $k^0 + td, -\infty < t < +\infty$ with $\mathcal{K}$.

HR method generates points in $\mathcal{K}$ as follows:

$$k^1 = k^0 + t_1 d, \quad t_1 \text{ is uniformly distributed on } L.$$

Then k^0 is replaced with k^1, L is updated with respect to k^1 and so on.

The simplest theoretical result on the behavior of HR method states that if $\mathcal{K}$ does not contain lower dimensional parts, then the method achieves the neighborhood of any point of $\mathcal{K}$ with nonzero probability and asymptotically the distribution of points k_i tends to uniform one.

Theorem 1. (compare Smith (1984), Theorem 3). *Suppose $\mathcal{K}$ is open or coincides with the closure of interior points* *of $\mathcal{K}$. Then for any measurable set $A \subset \mathcal{K}$ probability $P_i(A) = P(k_i \in A|k_0)$ can be estimated as $|P_i(A) - P(A)| \leq q^i$, where $P(A) = Vol(A)/Vol(\mathcal{K})$ and $q < 1$ does not depend on k_0.*

Unfortunately q strongly depends on geometry of $\mathcal{K}$ and dimension n and can be close enough to 1. Tighter bounds for the rate of convergence can be found in Lovasz (1999) for convex $\mathcal{K}$. For some problems boundary oracle is hard to calculate, and we will use simplified version of it, where the first intersection of the line with $\mathcal{K}$ is found:

$$\bar{t} = \sup\{t : k^0 + \tau d \in \mathcal{K},\ \forall \tau \in [0, t]\},$$
$$\underline{t} = \sup\{t : k^0 - \tau d \in \mathcal{K},\ \forall \tau \in [0, t]\}$$

Thus we take the segment $[-\underline{t}, \bar{t}]$ instead of L. For this version Theorem 1 is false for disjoint $\mathcal{K}$. However for simply connected sets with some additional properties an analog of the result is true.

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

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

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

where $P_i(s)$ are m-th degree polynomials, and define the set $\mathcal{K}$ in the space of parameters $k = (k_1, \dots, k_n)$ which correspond to stable polynomials:

$$\mathcal{K} = \{k : P(s, k) \text{ is Hurwitz}\} \tag{2}$$

The geometry of such sets and of their boundaries is well studied, see Gryazina and Polyak (2006). 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 $L = \{t : A(s) + tB(s) \text{ is Hurwitz}\}$ 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:

$$\mathcal{K} = \{K : A + BKC \text{ is Hurwitz}\} \tag{3}$$

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 $K^0 \in \mathcal{K}$. Then $L = \{t : F + tG \text{ is Hurwitz}\}$. L consists of finite number of intervals, the algorithm for calculating their end points is presented in Gryazina and Polyak (2006), Section 4. However sometimes "brute force" approach is more simple. Introduce $f(t) = \max \Re \operatorname{eig}(F + tG)$, 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

$$\mathcal{K} = \{k : P_0(s, q) + \sum_{i=1}^{n} k_i P_i(s, q) \text{ is Hurwitz for all } q \in Q\} \tag{4}$$

If Q is a finite set $\{q_1, \dots, q_m\}$ and m is small, the set $\mathcal{K}$ 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 Section 5 below).

4. Quadratic stability set. This set is defined as solution of some LMIs, see S. Boyd and Balakrishnan (1994). The typical example is the set of symmetric matrices P defined by Lyapunov inequality:

$$\mathcal{K} = \{P > 0 : AP + PA^T \leq H\} \tag{5}$$

where A is a stable matrix and $H < 0$. This set is always convex, and boundary oracle can be found explicitly. Indeed, take $P_0 \in \mathcal{K}$, generate $D = D^T, ||D||_F = 1$ — a matrix, uniformly distributed on the unit sphere in Frobenius norm. Then $A(P_0 + tD) + (P_0 + tD)A^T \leq H \iff F + tR < 0, F = AP_0 + P_0A^T - H,\ R = AD + DA^T$. Then (see Polyak and Scherbakov (2006)) $L = (-\underline{t}, \bar{t})$ and $\bar{t} = \min \lambda_i,\ \underline{t} = \min \mu_i,\ \lambda_i$ are positive real eigenvalues of matrix pencil $F, -R$, μ_i are positive real eigenvalues of matrix pencil F, R.

We do not discuss all other versions of boundary oracles for different sets; they can be constructed in similar way.

3. GENERAL SCHEME OF HR METHOD

Now, when boundary oracle is specified, we can apply HR method. The general setup for use of the method in control design is as follows.

1. Given the set $\mathcal{K}$ of design variables (e.g. controller parameters or uncertainties). The set $\mathcal{K}$ is the admissible set with respect to some specifications (e.g. the set of stabilizing controllers). It is given in implicit form — just boundary oracle is available.

2.The only assumption is that we have a point k^0 in $\mathcal{K}$. Starting with this point we generate the sequence $k^0, k^1, \dots, k^N$ in $\mathcal{K}$ by use of HR method. These points are approximately uniformly distributed in $\mathcal{K}$ and give good representation of the entire $\mathcal{K}$.

3. We have a performance index $J(k)$ and some extra specifications. They can be any engineering characteristics — gain or phase margin, overshoot or other time-response characteristics, robustness margin — or mathematical objectives such as H_2 or H_∞ norm. First, we reject points k^i which violate these specifications. Second, we calculate $J(k^i)$ for remaining samples and find the optimal one k^*.

4. In the neighborhood of k^* we can repeat the process, taking the intersection of $\mathcal{K}$ with some trust region (say a ball centered at k^* of radius ε) as new admissible set and k^* as the initial point.

The first challenge, however, is establishing whether solutions exist, i.e. obtaining the starting point k^0. We propose the following routine for this need. For the stabilization problem an arbitrary $\hat{k}$ should be taken and the value $\sigma = \max \Re(\texttt{roots}\ P(s, \hat{k}))$ (for polynomial stabilization) or $\sigma = \max \Re(\texttt{eig}\ A(\hat{K}))$ (for matrix stabilization) is calculated. Then we generate HR points in the region

$$\hat{\mathcal{K}} = \{k : \Re\texttt{roots}P(s, k) \le \sigma\} \qquad \text{or}$$

$$\hat{\mathcal{K}} = \{K : \Re\texttt{eig}(A + BKC) \le \sigma\}$$

instead of $\mathcal{K}$ and minimize σ. Since $\sigma < 0$ is obtained for a generated point $\tilde{k}$ take $k^0 = \tilde{k}$ as a starting point.

For the quadratic stabilization problem the approach is similar. Take arbitrary $\hat{P} > 0$ and set $\sigma = \max \Re\texttt{eig}(A\hat{P} + \hat{P}A^T - H)$ then $\hat{P}$ is a feasible point for the region

$$\hat{\mathcal{K}} = \{P > 0 : AP + PA^T \le H + \sigma I\}.$$

We generate HR points in $\hat{\mathcal{K}}$ and minimize σ. Obtaining $\sigma < 0$ for a generated point $\tilde{P}$ means that $P^0 = \tilde{P}$ is a feasible point for the initial $\mathcal{K}$ and can be treated as a starting point. Unfortunately, we can not guarantee the convergence of the proposed routine but it showed high efficiency.

Further we assume starting point k^0 to be known to demonstrate the applications of the HR algorithm.

4. STABILIZATION OF SISO AND MIMO SYSTEMS

1. Consider LTI SISO 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

$$P(s) = a(s)f(s) + b(s)g(s). \tag{6}$$

If we treat the coefficients of the polynomials $f(s), g(s)$ as parameters k, we are at the setup of (1). Thus we can apply the general scheme of HR method from Section 3.

It is of interest to compare this technique with the one reported in Kiselev and Polyak (1999). In this paper the case of two parameters was considered; this admits graphical representation of $\mathcal{K}$. Generation of points in $\mathcal{K}$ was done by-hand, with no attempts to cover this set uniformly. Nevertheless solution of many classical examples was improved in Kiselev and Polyak (1999). Our approach is the extension of this technique to multi-dimensional case with more systematical exploration of the admissible set.

Example 1. This example is taken from Demo of Robust Control Toolbox for MATLAB 5.0 and it was improved in Kiselev and Polyak (1999). Given a SISO plant

$$P(s) = \frac{9000}{s^3 + 30s^2 + 700s + 1000}.$$

The objective is to find a controller that minimizes

$$F = \left\| \begin{matrix} W_1 S \\ W_2 T \end{matrix} \right\|_\infty,$$

$$W_1 = \frac{1.5(1 + s/30)^2}{0.01(1 + s)^2}, \quad W_2 = \frac{1 + s/40}{3.16(1 + s/300)}.$$

The sixth-order controller proposed by Robust Control Toolbox provides $F = 0.998$. In Kiselev and Polyak (1999) the second-order controller

$$C^0(s) = \frac{0.052s^2 + 1.16s + 10.41}{(1.01s + 0.902)(0.011s + 1.005)}.$$

was found that provides $F = 1.296$.

For application of HR method in this example we generate 1000 stabilizing controllers starting with C^0. The stability domain is not bounded in $\mathbb{R}^6$ space of the parameters so we need to specify bounds for the controller parameters. First we take 1-box neighborhood of the starting controller parameters and it allows $\approx 1\%$ improvement of the quality criteria. Then for 1000 points generated in 0.1-box neighborhood the local improvement is $\approx 11\%$ for the controller

$$C^*(s) = \frac{0.059s^2 + 1.111s + 10.47}{0.005s^2 + 0.938s + 0.807}.$$

Example 2 Y. Fujisaki and Tempo (2007). Given a SISO 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 $||W(s)S(s)||_\infty < 1$ with $W(s) = \frac{55(1+3s)}{1+800s}$. Starting with a controller found in Y. Fujisaki and Tempo (2007)

$$C^0(s) = \frac{-0.532 - 0.5407s - 2.0868s^2}{1 - 0.3645s - 1.2592s^2} \tag{7}$$

we restrict controller parameters 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

$$C^*(s) = \frac{-0.5317 - 0.5686s - 2.0905s^2}{0.9901 - 0.2995s - 1.2008s^2}$$

that leads to $||W(s)S(s)||_\infty = 0.8206$ compared to 0.9822 for controller (7). So here Hit-and-Run also allows performing local improvement.

2. Proceed to static output feedback stabilization for uncertain LTI MIMO plant:

$$\dot{x} = A(q)x + B(q)u, \; y = C(q)x, \; u = Ky, \tag{8}$$

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

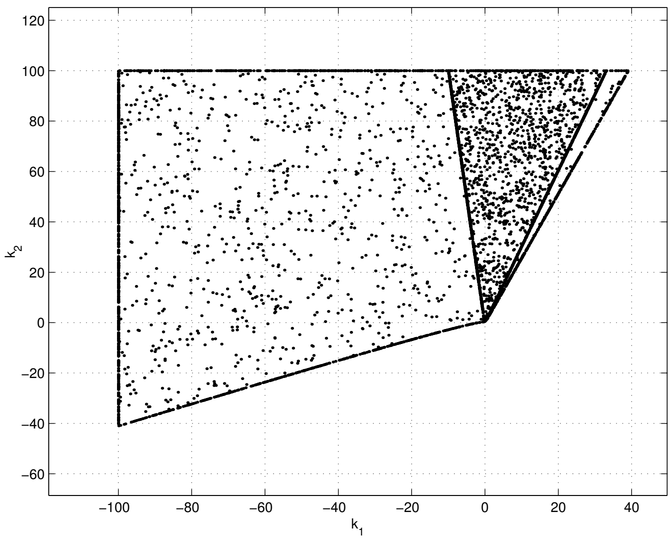

Fig. 1. Stabilizing controller parameters for nominal system

Example 3.

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

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

$q \in Q_\rho = \{q : |q_i - q_i^0| \le \rho\gamma_i\}$, $q^0 = [0.3681, 1.42, 3.5446]$; $\gamma = [0.05, 0.01, 0.04]$. 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 was studied in Bhattacharyya (1987), El Ghaoui and AitRami (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 $A_c = A + BKC$, thus we are in the framework of (3) and can apply HR method tailored for this problem. Starting with the stabilizing controller $K = [-0.4357;\ 9.5652]$ (see El Ghaoui and AitRami (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. Fig. 1 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.

5. ROBUST QUADRATIC STABILIZATION

The general setup has been described in Part 4 of Section 2. We illustrate how this technique works for one example.

Example 4. Here we investigate the example originated in Barmish (1985). Consider a system with uncertainty (8) with

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

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 $P > 0$: $A_c^T P + PA_c < 0$. 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

$$\mathcal{K} = \bigcap_i \{Q > 0, \quad QA_i^T + A_iQ + B_iY + Y^T B_i^T < 0\},$$

where index i corresponds to the vertex sample. For generating quadratic robust stabilizing controllers the boundary oracle for the set (5) is exploited taking $Q = Q_0 + J$, $Y = Y_0 + G$ and $F = Q_0A_i^T + A_iQ_0 + B_iY_0 + Y_0^T B_i^T$, $R = JA_i^T + A_iJ + B_iG + G^T B_i^T$, 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 $K = YQ^{-1}$. Fig. 2 depicts robust stabilizing controllers for the original uncertain set with $\rho = 0.5$ (points).

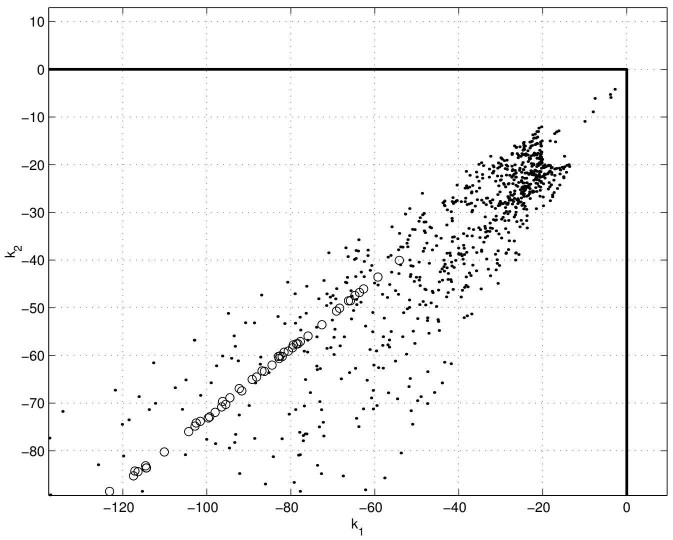

Fig. 2. Robust quadratic stabilizing controllers

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 "o" in Fig. 2. 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. $||K||_\infty \leq \gamma$ with starting point K^0 satisfying this condition. Another natural box-restriction is $||K - K^0||_\infty \leq \gamma$ as it was used in Examples 1,2. 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 $\bar{t}$ of the boundary oracle in HR algorithm.

6. CONCLUSIONS

Hit-and-Run is the promising tool for stabilization and optimization of linear systems. It allows solving high dimensional problems since the boundary oracle is available. The only requirement is the knowledge of one point belonging to the set but an auxiliary routine for obtaining this point is proposed. For the problems of polynomials and matrices stabilization, robust and quadratic stabilization general scheme of HR method is presented. Various specifications can be considered simultaneously as well as additional box-restriction for the controller parameters can be handled.

HR gives an opportunity to come up to hard problems. For instance, we dare to find row-vector $K : 1 \times 500$, stabilizing the continuous-time system with matrix $A : 500 \times 500$, $B : 500 \times 1$. It appears to be possible to generate 1000 stabilizing controllers less than for 30 minutes for the system in canonical form, this system matrix is extremely sensitive (condition number $1e{+}150$) to perturbations. For full matrices the procedure is much faster.

The proposed approach looks very promising for different design problems in control.

ACKNOWLEDGEMENT

The authors are thankful to Prof. R. Tempo from IEIIT (Italy). Support of bilateral joint project between RAS and CNR is also acknowledged.

REFERENCES

B. Barmish. Necessary and sufficient condition for quadratic stabilizability of an uncertain system. *Journ. of Optim. Theory and Appl.*, 46(4), 1985.

D. Bertsimas and S. Vempala. Solving convex programs by random walks. *Journ. of the ACM*, 51(4), 2004.

S. Bhattacharyya. Robust stabilization against structured perturbations. 1987.

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

E. Gryazina and B. Polyak. Stability regions in the parameter space: D-decomposition revisited. *Automatica*, 42(1), 2006.

O. Kiselev and B. Polyak. Design of low-order controllers with h_∞ or maximal robustness performance index. *Automation and Remote Control*, 60(3), 1999.

J. Lofberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://control.ee.ethz.ch/ joloef/yalmip.php`.

L. Lovasz. Hit-and-run mixes fast. *Mathematical Programming*, 86(3), 1999.

B. Polyak and P. Scherbakov. D-decomposition technique for solving linear matrix inequalities. *Automation and Remote Control*, 67(11), 2006.

E. Ferron S. Boyd, L. El Ghaoui and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, 1994.

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

R.L. Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6), 1984.

R. Tempo, G. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems.* Springer, London, 2004.

V. F. Turchin. On the computation of multidimensional integrals by the monte-carlo method. *Theory of Probability and its Applications*, 16(4), 1971.

Y. Oishi Y. Fujisaki and R. Tempo. Mixed deterministic/randomized methods for fixed order controller design. *IEEE Transactions on Automatic Control*, 2007.