Quasi-Monte Carlo Methods in Robust Control Design

Peter F. Hokayem†, Chaouki T. Abdallah†, Peter Dorato
EECE Department, MSC01 1100
1 University of New Mexico
Albuquerque, NM 87131-0001, USA
{hokayem,chaouki,peter}@eece.unm.edu

***Abstract*— Many practical control problems are so complex that traditional analysis and design methods fail to solve. Consequently, in recent years probabilistic methods that provide approximate solutions to such 'difficult' problems have emerged. Unfortunately, the uniform random sampling process usually used in such techniques unavoidably leads to clustering of the sampled points in higher dimensions. In this paper we adopt the quasi-Monte Carlo methods of sampling to generate deterministic samples adequately dispersed in the sample-space. Such approaches have shown to provide faster solutions than probabilistic methods in fields such as Financial Mathematics.**

I. Introduction

Many control problems are so complex in nature that analytic techniques fail to solve them. Furthermore, even if analytic solutions are available, they generally result in very high order compensators. It is for these reasons that we accept approximate answers to provide us with certain guarantees in such control problems. This is when sampling methods come into the picture to try and remedy the "cost of solution" problem by drawing samples from a sample space, and providing an approximate answer. For many years, random sampling has dominated the afore mentioned arena. Recently however, deterministic or quasi-Monte Carlo methods have proven superior to random methods in several applications.

In this paper we are interested in exploiting the quasi-Monte Carlo deterministic method of generating point samples from a sampling space in robust control problems. Quasi-Monte Carlo methods have been extensively used in financial mathematics in recent years, especially in calculating certain financial derivatives in very high dimensions. The controls community has so far relied heavily on generating random samples based on Monte Carlo theory for the evaluation of performance objectives for various problems in robust control. However, random sample generation, with a uniform underlying distribution, tends to cluster the samples on the boundary of the sample space in higher dimensions, unless we try to learn the underlying distribution. It is for the latter reason that we are interested in presenting a method that distributes the points *regularly* in the sample space while providing deterministic guarantees.

†The research of C.T. Abdallah and P.F. Hokayem is partially supported by NSF-0233205

The paper starts by formulating the robust stabilization problem in Section II. The we provide a brief summary of Monte Carlo methods and how they are used to solve the problem at hand in Section III. In Section IV, we present a fairly extensive exposure of quasi-Monte Carlo methods, with different methods for generating point sets of low discrepancy, hence low error bounds. Finally, in Section V, we simulate both random and quasi-random methods and compare them with respect to their ability to retain their level of accuracy as the number of points increases.

II. Problem Formulation

Consider the control problem shown in Fig 1:

Problem 1: Given a real rational plant model $G(s,p)$, with uncertain parameter vector $p = [p_1 \quad p_2 \quad \dots \quad p_n] \in \mathbb{I}_p^n$, does there exist a controller $C(s,q)$ that can stabilize the uncertain system, where $q = [q_1 \quad q_2 \quad \dots \quad q_m] \in \mathbb{I}_q^m$ is the admissible controller parameter vector.

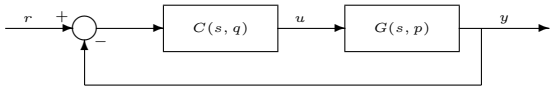

Fig. 1. Feedback Structure.

In Problem 1 above, $\mathbb{I}^i$ is the unit i-dimensional hypercube in $\mathbb{R}^i$. Without loss of generality the regions of plant uncertainty and design parameters have been scaled to the unit hypercubes $\mathbb{I}_p^n$ and $\mathbb{I}_q^m$, respectively. Let $T(s,p,q) = \frac{C(s,q)G(s,p)}{1+C(s,q)G(s,p)}$ be the closed-loop transfer function.

Problem 1 is the robust stabilization problem, and requires that the controller $C(s,q)$ stabilizes every plant inside the uncertainty interval ($\mathbb{I}_p^n$). This problem is inherently hard to solve in general, since we essentially have to check if all the plants inside the uncertainty set $\mathbb{I}_p^n$ are stabilizable, which is virtually impossible in a limited time span, due the continuity of the uncertainty interval. That is why we relax the problem into an approximate one through sampling. The method of solution is fairly simple using sampling and casting Problem 1 into an *empirical mean* (or integration) setting.

While Problem 1 requires an exact solution for the robust stabilization problem, the approximate solution requires the use of an *indicator function* (Ψ), which provides answers for

discrete points of the plant parameter uncertainty spectrum and admissible controller parameter space.

Definition 1: An indicator function Ψ is a decision type function that attains crisp values that belong to the discrete set $A = [0, 1, 2, \ldots, d]$ depending on the decision criteria used to evaluate the problem, at specific points of the sample space.

Definition 1 is a general one for indicator functions, but for our purposes we specialize it to fit our context as follows:

$$\Psi(P_i, Q_j) = \begin{cases} 1, & T(s,p,q) \text{ is stable} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where P_i and Q_j are sampled vectors from the plant parameter space and admissible controller parameter space, respectively, and $A = [0, 1]$.

Having defined the indicator function Ψ, we can easily cast Problem 1 into a sampling context as follows:

Problem 2: Consider Problem 1: Find vector $Q^* = [q_1^* \quad q_2^* \ldots q_m^*] \in \mathbb{I}_q^m$ which stabilizes the uncertain plant with a high level of confidence, that is, Q^* maximizes

$$f_{Q^*}(P_i) = f(P_i, Q^*) = \frac{1}{N} \sum_{i=1}^{N} \Psi(P_i, Q^*) \quad (2)$$

where f is called the counting function, and N is a large number.

Problem 2 gets rid of solving the problem over a continuous plant parameter space through sampling that space, and *counting* those samples that result in $\Psi = 1$, i.e. a stable combination of P_i and Q_j. The second step is to pick $Q^* = Q_j$ that produces the largest answer for $f_Q(P)$, the counting function. The function $f_Q(P)$ can be interpreted as the average performance of the uncertain system with respect to a certain controller Q_i.

III. Monte Carlo Method

In this section we define briefly the Monte Carlo method in general, and then specialize it to solve Problem 2.

The Monte Carlo method was first published in 1949 by Metropolis and Ulam at Los Alamos National laboratory. Since then it has been used extensively in various areas of science such as statistical and stochastic physical systems [4], derivative estimation [2], and integral evaluation [3].

Loosely defined, *Monte Carlo is a numerical method based upon random sampling of the parameters space*. There are many variations of sampling methods but the main idea is invariant throughout all methods. Given a function $g(x)$, it is required to find $\int_{\mathbb{I}^d} g(x)dx$ (the integration region has been scaled to the unit hypercube for consistency). Usually the dimension 'd' is high, and numerical solutions are computationally expensive. That is when Monte Carlo method comes into the picture, because it overcomes the dimensionality problem. The first step is to equip the integration region ($\mathbb{I}^d$) with a d-dimensional probability density Π, usually uniform if no prior knowledge of the distribution is available. The second step is to integrate with respect to the probabilistic distribution as follows:

$$\phi = \int_{\mathbb{I}^d} g(x)dx = \int_{\mathbb{I}^d} g(x)dx = \lambda_d(\mathbb{I}^d) \int_{\mathbb{I}^d} g(\eta)d\eta = \mathbb{E}\{g(\eta)\} \quad (3)$$

where λ_d is an d-dimensional Lebesgue measure and $\mathbb{I}^d$ is transformed into a probability space equipped with a probability measure $d\eta = \frac{dx}{\lambda_d(\mathbb{I}^d)}$ [1], [3]. As a result, the problem of evaluating the integral has been simply transformed into evaluating the expected value on the probability space, which provides an approximate answer. For an extensive overview on Monte Carlo methods in robust control problems see [5], [14], [17], [18].

The dimension 'd' could be extremely large in some applications, however the probabilistic results obtained using Monte Carlo methods are dimension-independent. Finally, and concerning the asymptotic behavior of Monte Carlo methods of sampling, the convergence error in (3) between expected value and the actual value of the integral is of order $\mathcal{O}(N^{-1/2})$, where N is the number of samples. The constant by which the order is multiplied is a function of the variance of the samples. That is why, different Monte Carlo methods are usually targeted at decreasing the variance of the samples (see [3]). See Figure 2 for illustration of uniform random sampling in the 2-dimensional unit plane[1]. It can be easily spotted that there are several clusters in the sample set, and huge gaps as a result.

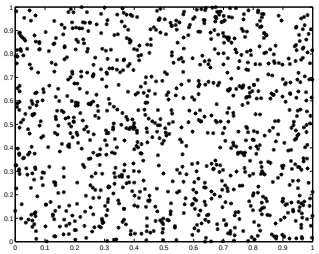

Fig. 2. Uniform random sampling in 2D for 1000 points

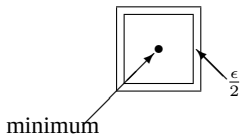

Fig. 3. Sampling in higher dimensions

A. Sampling in Higher Dimensions

Assume that we are trying to minimize a function over the unit cube ($[0, 1]^d$), and that the minimum (or maximum[2]) point is exactly at the center of the hypercube. Also consider a smaller hypercube of sides equal to $1-\epsilon/2$ inside the original unit cube. Now, imposing a uniform probability distribution on the unit hypercube, results in the probability of sampling inside the smaller cube (see Figure 3): $\mathbb{P}_{I_{inner}} = (1-\epsilon)^d$, where d is the dimension of the hypercube. Consequently, the probability of sampling inside I_{inner} tends to zero as $d \to \infty$, hence the clustering effect on the surface is observed as we go

[1]All the simulations in this paper were done using MATLAB software

[2]Converting between max and min problems is trivial

into higher dimensions. This phenomenon is typical in Monte Carlo simulations, if no prior knowledge of the distribution is known. One way to remedy the problem may be to learn the underlying distribution, which is usually computationally much more demanding.

IV. Quasi-Monte Carlo Methods

In this section we expose the reader to quasi-Monte Carlo methods. The main idea is to replace the random samples required for Monte Carlo simulation with deterministic samples that possess certain regularity conditions, i.e. they are regularly spread within the sampling space. This method is also independent of the dimension of the sampling space. It has shown its superiority over Monte Carlo methods in the calculation of certain integrals [11], financial derivatives [12] and motion planning in robotics [16]. Recently, quasi-Monte Carlo methods have been used for stability analysis of high speed networks [15]. Certain variations involving randomization of quasi-Monte Carlo methods were presented in [13]. However, in what follows we are going to present basic ideas in quasi-Monte Carlo methods due to the substantial diversity of the subject.

A. Preliminaries

We start off by introducing certain mathematical facts that will aid us in the evaluation of the error bounds for each of the different methods of generation of quasi-Monte Carlo points sets.

1) Discrepancy: The *discrepancy* is a measure of the 'regularity in distribution' of a set of points in the sample space. In order to define it mathematically, we need to define the following counting function: $A(B;P)=\sum_{i=1}^{N} I_B(X_i)$, where $B\subset \mathbb{I}^d$ is an arbitrary set, $P=(X_1,\ldots,X_N)$ is a point set, N is the number of points, and I_B is an indicator function.

Definition 2: The general formula for the evaluation of the discrepancy is given by

$$\mathcal{D}_N(\mathcal{B},P)=\sup_{B\subset\mathcal{B}}\left|\frac{A(B,P)}{N}-\lambda_d(B)\right| \tag{4}$$

where $\lambda_d(B)$ is the d-dimensional Lebegue measure of the arbitrary set B and $\mathcal{B}$ is the family of all lebesgue measurable subsets B of $\mathbb{I}^d$.

Definition 2 can be specialized into the following two cases:

- The *star discrepancy* $D_N^\star(X_1,\ldots,X_N)$ is obtained by letting $\mathcal{B}$ in (4) be defined as follows $\mathcal{B}^\star=\{\forall B: B=\prod_{i=1}^{d}[0,u_i)\}$ i.e. the set of all d-dimensional subsets of $\mathbb{I}^d$ that have a vertex at the origin, and u_i's being arbitrary points in the corresponding 1-dimensional space.
- The *extreme discrepancy* $D_N(X_1,\ldots,X_N)$ is obtained by letting $\mathcal{B}$ in (4) be defined as follows $\mathcal{B}=\{\forall B: B=\prod_{i=1}^{d}[v_i,u_i)\}$ where v_i's and u_i's are both arbitrary points in the corresponding 1-dimensional space.

The star discrepancy and extreme discrepancy are related through the following inequality $D_N^\star(P)\leq D_N(P)\leq 2^d D_N^\star(P)$.

2) Error in Quasi-Monte Carlo: [1]

The error in quasi-Monte Carlo methods integration over the unit hypercube for N samples is defined as follows,

$$e=\frac{1}{N}\sum_{n=1}^{N} f(X_n)-\int_{\mathbb{I}^d} f(\eta)d\eta \tag{5}$$

Define the total variation of the function f on $\mathbb{I}^d$ in the sense of *Vitali* as $V^{(d)}(f)=\sup_{\mathcal{P}}\sum_{J\in\mathcal{P}}|\Delta(f;J)|$, where $\mathcal{P}$ is the set of all partitions of $\mathbb{I}^d$, J is a subinterval of $\mathbb{I}^d$, and $\Delta(f,J)$ is an alternating sum of values of f at the vertices of J. The variation could also be equivalently defined as

$$V^{(d)}(f)=\int_0^1\ldots\int_0^1\left|\frac{\partial^d f}{\partial\eta_1\ldots\partial\eta_d}\right|d\eta_1\ldots d\eta_d \tag{6}$$

whenever the indicated partial derivative is continuous on $\mathbb{I}^d$. This variation is redefined on $\mathbb{I}^d$ in the sense of *Hardy and Krause* as

$$V(f)=\sum_{k=1}^{d}\sum_{1\leq i_1<i_2<\cdots<i_k\leq d} V^{(k)}(f;i_1,i_2,\ldots,i_k) \tag{7}$$

where $V^{(k)}(f;i_1,i_2,\ldots,i_k)$ is the variation of f in the sense of *Vitali* rerstricted to a k-dimensional surface. f has bounded variation if $V(f)$ in (7) is finite. Now we are ready to state the error formula for quasi-Monte Carlo methods.

Theorem 1: [1] If f has bounded variations in the sense of Hardy and Krause on $\mathbb{I}^d$, then for any point set $\{X_i\}_{i=1}^N$ we have

$$|e|\leq V(f)D_n^\star(X_1,\ldots,X_N) \tag{8}$$

Basically, the magnitude of the error depends on the variation of the function and the star discrepancy of the point set chosen. That is why we are always after low star discrepancy point sets in quasi-Monte Carlo methods. It is also worth mentioning that the error bound in (8) is conservative, i.e. if the variation of the function is large, we get a large bound on the error, although the actual error might be small. This error bound is obtained for multi-dimensional integrals of functions, however another error bound could be obtained for a 1-dimensional integral in terms of the modulus[3] of continuity.

In subsequent sections we are going to present the error bounds for each of the methods used in generating the low discrepancy point sets. The values given are for the star discrepancy of the sequence, which is then reflected in the error bound given in (8).

B. Point Sets Generation

In this section we briefly describe how to generate quasi-Monte Carlo low discrepancy points in an s-dimensional sample space. Since the points result from a deterministic method of generation, they possess a certain regularity property of distribution in the sample space described by their discrepancy. This gives the method leverage over Monte Carlo methods in the sense that the guarantees over the error

[3] $|e| \leq \omega(f;D_N^\star(X_1,\ldots,X_N)) = \sup_{u,v\in[0,1]\&|u-v|\leq D_N^\star(X_1,\ldots,X_N)}|f(u)-f(v)|$

magnitude are deterministic and are given by (8).

1) Van Der Corput: [1]

The van der Corput sequence in base b, where $b \geq 2 \in \mathbb{N}$, is a one dimensional sequence of points that possesses the property of having a low discrepancy in the unit interval $\mathbb{I} = [0,1] \subset \mathbb{R}$. *The main idea is to express every integer* $n \in \mathbb{N}$ *in base* b *and then reflect the expansion into the unit interval* $\mathbb{I}$. This is done as follows:

1) Let $R_b = \{0, 1, \ldots, b-1\}$ be the residue set modulo b
2) Any integer $n \geq 0$ can be expanded in base b as: $n = \sum_{k=0}^{\infty} a_k(n) b^k$, where $a_k(n) \in R_b, \forall k$.
3) Finally, we get the sequence $\{X_n\}$ as $X_n = \phi_b(n) = \sum_{k=0}^{\infty} a_k(n) b^{-j-1}$.

As will be seen, the van der Corput sequence will be used to generate higher dimensional vector samples, with the variation of the expansion base b. Finally, the star discrepancy of the van der Corput sequence is given by: $D_N^{\star}(X_1, \ldots, X_N) = \mathcal{O}(N^{-1}\log(N))$, with a constant depending on the base of expansion.

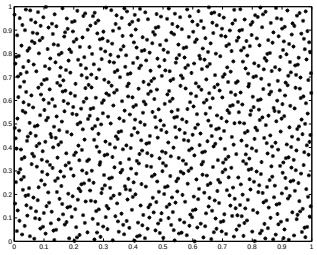

Fig. 4. Halton sequence in 2D for 1000 points

2) Halton Sequence: [1]

The Halton sequence is a generalization of the van der Corput sequence in Section IV-B.1 to span an d-dimensional sample space. The main idea is to generate d 1-dimensional sequences and form the corresponding d-dimensional vector sample points. Let $b_1, b_2, \ldots, b_d$ be the corresponding expansion bases for each dimension, preferably relatively prime[4]. Let $\phi_{b_1}, \phi_{b_2}, \ldots, \phi_{b_d}$ be the corresponding reflected expansions according to the corresponding bases. Then the d-dimensional sequences $\{X_n^d\}$ are formed as follows: $X_n = (\phi_{b_1}, \phi_{b_2}, \ldots, \phi_{b_d}) \in \mathbb{I}^d$ A figure of a 2-dimensional Halton sequence is shown in Figure 4. Assume that the bases for the expansion are relatively prime, then the star discrepancy is given by (see [1])

$$D_N^{\star}(X_1, \ldots, X_N) < \frac{d}{N} + \frac{1}{N} \prod_{i=1}^{d} \frac{b_i - 1}{2 \log b_i} \log N + \frac{b_i + 1}{2} \quad . \quad (9)$$

3) Hammersley Sequence: [1]

The Hammersley sequence is generated as Halton sequences in Section IV-B.2. However, there is slight difference is that we form $d-1$, ϕ-sequences and then the d-dimensional vector samples are generated as follows: $X_n = (\frac{n}{N}, \phi_{b_1}, \ldots, \phi_{b_{d-1}})$. One extra difference between the Halton and Hammersley

[4]Choosing the expansion bases relatively prime reduces the discrepancy, hence the error bound

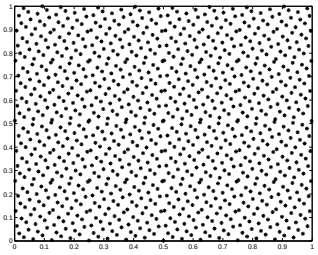

Fig. 5. Hammersley sequence in 2D for 1000 points

sequences is that, the first can virtually be expanded for infinite N while the second requires an upper bound on the number of samples. See Figure 5 for a 2-dimensional Hammersley sequence. The bound on the star discrepancy of the Hammersley sequence is given by (9) except that the product is terminated at $(d-1)$, because we only have $(d-1)$ van der Corput sequences in the construction.

4) (t,s)-Sequences: [1], [6], [7]

One of the most successful low discrepancy sequences are (t,s)-sequences[5] which depend on the theory of (t,m,s)-nets. The construction of such sequences is much more involved than those introduced earlier. Consequently, our discussion will be concise and the interested reader is referred to [1], [6], [7]. First, we will start with some preliminary definitions.

Definition 3: Let $E \subset \mathbb{I}^s$ be an s-dimensional subinterval defined as follows: $E = \prod_{i=1}^{s} [a_i b^{-d_i}, (a_i + 1) b^{-d_i})$, where $a_i, d_i \in \mathbb{Z}$ such that $b \geq 2$, $d_i \geq 0$ and $0 \leq a_i \leq b^{d_i}$.

Definition 4: Let E be as in Definition 3, then a (t,m,s)-net in base b is a point set P; such that, $card(P) = b^m$ and each E interval contains $N\lambda_s(E)$ points, where $\lambda_s(E) = b^{t-m}$ is the s-dimensional Lebegue measure of E.

Basically, Definition 4 guarantees that the samples are distributed evenly inside smaller hypercubes $E \subset \mathbb{I}^s$. This property decreases the discrepancy value of the point sequence P, hence the error bound.

Next we define (t,s)-sequences starting with one dimensional sequence and generalizing to s-dimensional. Let $x \in [0,1] \subset \mathbb{R}$ then the reflected expansion in base b is defined as follows:

$$x = \sum_{j=1}^{\infty} a_j b^{-j}, \qquad a_i \in R_b = \{0, 1, \ldots, b-1\} \qquad (10)$$

Given an integer $m \geq 0$ define the truncated sequence of (10) as, $[x]_{b,m} = \sum_{j=1}^{m} a_j b^{-j}, \qquad a_i \in R_b = \{0, 1, \ldots, b-1\}$, which is a one-dimensional truncated sequence. Then, we expand the one-dimensional truncated sequence into an s-dimensional one; $[X]_{b,m} = \left([x^{(1)}]_{b,m}, \ldots, [x^{(s)}]_{b,m}\right)$, and now we are ready to state the main definition of (t,s)-sequences.

Definition 5: For $b \geq 2$ and $t \geq 0$ being integers; a sequence $X_0, X_1, \ldots$ in $\mathbb{I}^s$ is a (t,s)-sequence in base b, if

[5]In this section we let 's' be the dimension of the space instead of 'd', in order to preserve the nomenclature of the sequence '(t,s)', as given in the cited references.

$[X_n]_{b,m}$ and $kb^m \leq n \leq (k+1)b^m$ form a (t,m,s)-net in base b, for $k \geq 0$ and $m > t$.

Note 1: The van der Corput introduced in Section IV-B.1 is an example of a (0,1)-sequence in base b.

Note 2: As might have been suspected, a smaller value of t in Definition 5 would result in stronger regularity in the sample space.

The discussion on (t,s)-sequences presented sofar is descriptive and the actual construction of such sequences is relatively complicated with several available methods and variations thereof, see [9] for an abridged presentation of available methods of construction. Finally, there are several available bounds on the star discrepancy of this method which we will not list here, however the interested reader may consult [1].

5) Lattice Points: The construction of lattice structured points is fairly simple. In [1] the general method is stated as follows:

- Let $1, \alpha_1, \alpha_2, \ldots, \alpha_d$ be linearly independent rational numbers.
- N is the number of sampled points.
- Then the lattice point set is constructed as follows $X_n = \left\{\frac{n}{N}(\alpha_1, \alpha_2, \ldots, \alpha_d)\right\} \in \mathbb{I}^d, \quad \forall n = 0, 1, \ldots, N-1$, and $\{.\}$ denotes the fractional part of the real number.

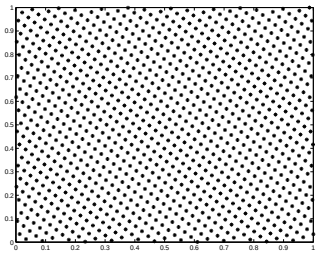

Fig. 6. Lattice point set in 2D for 1000 points, $a = \frac{\sqrt{5}+1}{2}$

It was reported in [10] that a good choice for lattice point sets would be those of the *Korobov type*, which are a special case of the choice above where $\alpha_i = \alpha^{i-1}$, i.e. $X_n = \left\{\frac{n}{N}(1, \alpha, \alpha^2, \ldots, \alpha^{d-1})\right\} \in \mathbb{I}^d, \quad \forall n = 0, 1, \ldots, N-1$, where $1 < \alpha < N, \alpha \in \mathbb{N}$. See Figure 6 for illustration in 2-dimensional space. As observed in Figure 6, the lattice point set has the best regularity of distribution of the points in the 2-dimensional unit plane. The derivation of the bound on the error in the case of lattice construction is fairly more involved, and depends on periodic functions and Fourier coefficients, and consequently will be omitted here.

V. Robust Control Problem Simulation

In this section we consider an old problem first introduced by Truxal in [8]. The main idea is having a hypercube-like parameter space ($\mathbb{I}^n$) with a hypersphere-like region ($\mathcal{B}^n(0, \rho)$) of instability. The problem becomes challenging when the instability radius becomes close to the boundary of the sampling space. Refer to Figure 1 with the plant transfer function $G(s, p, r) = \frac{s^2+s+(3+2p_1+2p_2)}{s^3+(1+p_1+p_2)s^2+(1+p_1+p_2)s+(0.25+\rho^2+3p_1+3p_2+2p_1p_2)}$ and the simple gain controller $C(s, q) = q$, with $q \in [0, 1]$, $p_1 \in [0, 1]$ and $p_2 \in [0, 1]$. The resulting closed-loop characteristic polynomial is

$$\begin{aligned} p(s) &= s^3 + (1 + p_1 + p_2 + q)s^2 \\ &+ (1 + p_1 + p_2 + q)s \\ &+ (0.25 + \rho^2 + 3p_1 + 3p_2 + 3q + 2p_1p_2 + 2p_1q + 2p_2q). \end{aligned} \tag{11}$$

Using Maxwell's criterion for 3^{rd}-order polynomials, we obtain the following multivariate-polynomial inequalities (MPIs) that guarantee the stability of (11),

$$\begin{aligned} v_1(p_1, p_2, q) &= 1 + p_1 + p_2 + q > 0 \\ v_2(p_1, p_2, q) &= 0.25 + \rho^2 \\ &+ 3p_1 + 3p_2 + 3q + 2p_1p_2 + 2p_1q + 2p_2q > 0 \\ v_3(p_1, p_2, q) &= p_1^2 + p_2^2 + q^2 - p_1 - p_2 - q + 0.75 - \rho^2 > 0 \end{aligned} \tag{12}$$

It is easily seen that the first and second inequalities in (13) are always positive for the ranges of uncertainties and design regions given. However, the 3^{rd} inequality requires a closer look to establish the stability regions for the closed-loop system. Through completing the squares, the 3^{rd} inequality could be written as

$$v_3(p_1, p_2, q) = (p_1 - 0.5)^2 + (p_2 - 0.5)^2 + (q - 0.5)^2 - \rho^2 > 0 \tag{13}$$

It is easily seen that (13) equated to zero results in the equation of a sphere centered at $(0.5, 0.5, 0.5)$ and radius ρ. Therefore, our instability region is defined by the intersection of the unit 3-dimensional hypercube and the spherical region given in (13). Consequently, the problem is restated as follows

$$\begin{aligned} Q_{sol} = \{q \in [0, 1] : \quad &\forall \quad p \in [0, 1], r \in [0, 1], p_1(p, q, r) > 0 \\ &\wedge \quad p_2(p, q, r) > 0 \wedge p_3(p, q, r) > 0\} \end{aligned} \tag{14}$$

Usually solution regions for problems such as the one presented in (14) are hard to obtain analytically. However, in our case the solution is fairly simple; $Q_{sol} = \{[0, 0.5 - \rho) \cup (0.5 + \rho, 1]\}$. For $\rho = 0.499$ we have $Q_{sol} = \{[0, 0.001) \cup (0.999, 1]\}$.

In what follows, we address the same the problem using sampling methods, random and quasi-random. The indicator function is the one mentioned in (1), where

- $P_i = [p_i, r_i]$ is the sample vector from the plant parameter space
- $Q_i = [q_i]$ is the scalar sample from the controller parameter space.

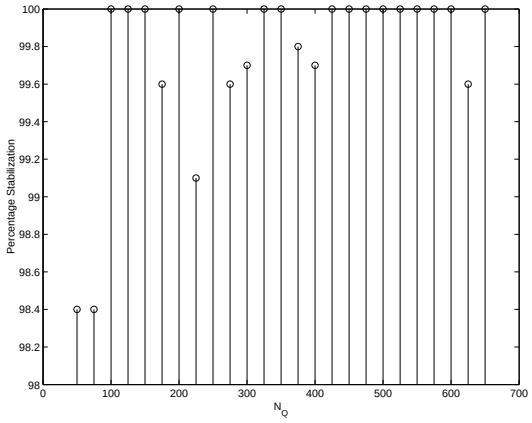

Fig. 7. Percentage stabilization with random uniform sampling

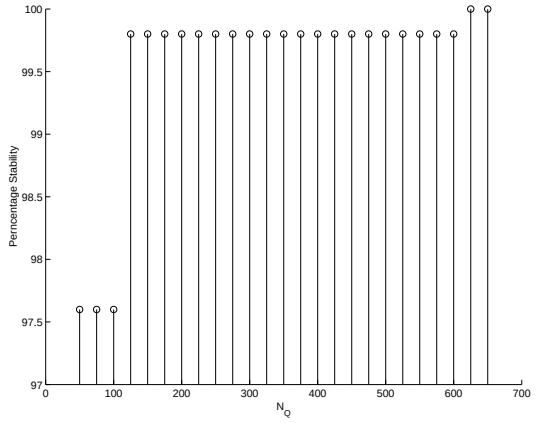

Fig. 8. Percentage stabilization with deterministic Halton sequence

A. Using Random Samples

Let $N_P = 1000$ be the number of samples taken from the parameter space $\mathbb{I}_p^2$ and $N_Q = \{50, 75, \ldots, 625, 650\}$ be the number of samples taken from the controller admissible space $\mathbb{I}_q^1$. In Figure 7, we present the number of samples vs. the best percentage stabilization achieved. Since the sampling is random, there are no deterministic guarantees that the plant can be stabilized even with a high number of controller samples. That is the reason why we can achieve 100% stability with 100 samples, on the other hand we may not reach 100% stability at 625 samples.

Note 3: When taking N_Q controller parameter samples for the simulation, we disregard previous samples for smaller N_Q.

B. Using Quasi-Random Samples

In this section we are going to explore the performance of deterministic quasi-Monte Carlo sampling. We follow the same presentation as in Section V-A, using the Halton sequence presented in Section IV-B.2. The result is seen in Figure 4. The only 100% stabilizing controller parameter for $N_Q = \{625, 650\}$ is $Q^\star = 0.00032 \in Q_{sol}$. As suspected the deterministic sequence retains its ability to stabilize the uncertain plant once it reaches the 100% stabilization mark. That is due to the fact that the points are not selected randomly, but are instead chosen to fill the sampling space in a regular fashion.

Note 4: In the simulations of Sections V-A and V-B, achieving 100% stability is only with respect to the samples (P_i) taken over the plant parameter space $(\mathbb{I}_p^2)$, and therefore our answer is approximate. There may be intervals between samples for which the closed-loop system is unstable.

VI. Conclusion

In this paper we have presented the robust stabilization problem and tackled it from a sampling point of view. A fairly self-contained presentation of Quasi-Monte Carlo point generation was presented. Then random and deterministic point generation were used in order to solve the robust stabilization problem. Both methods of sample generation were compared through simulation according to their ability to solve the problem at hand. Although random methods might converge to the solution at a lower number of samples, they might lose convergence at higher number of samples. However, deterministic quasi-Monte Carlo point generation retains its ability to find the solution once it converges. Future work aims at investigating the performance of quasi-Monte Carlo methods in high dimensional robust control problems and deriving analytic bounds for the error when dealing with MPI problems.

References

[1] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, 1992.

[2] G.A. Mikhailov, *New Monte Carlo Methods with Estimating Derivatives*, VSP, Netherlands, 1995.

[3] J.E. Gentle, *Random Number Generation and Monte Carlo Methods*, Springer-Verlag, NY, 1998.

[4] M.H Kalos and P.A. Whitlock, *Monte Carlo Methods*, John Wiley & Sons, 1986.

[5] R. Tempo and F. Dabbene, "Randomized Algorithms for Analysis and Control of Uncertain Systems: An Overview", *Perspectives in Robust Control - Lecture Notes in Control and Information Science*, (ed. S.O. Moheimani), pp. 347-362, Springer-Verlag, London, 2001.

[6] H. Niederreiter and C. Xing, "Nets, (t, s)-Sequences, and Algebraic Geometry", in *Random and Quasi-Random Point Sets*, (Eds. P. Hellekalek and G. Larcher), Springer-Verlag, NY, 1998.

[7] S. Tezuka, "Financial Applications of Monte Carlo and Quasi-Monte Carlo Methods", in *Random and Quasi-Random Point Sets*, (Eds. P. Hellekalek and G. Larcher), Springer-Verlag, NY, 1998.

[8] J.G. Truxal, "Control Systems-Some Unusual Design Problems", in *Adaptive Control Systems*, (Eds. E. Mishkin and L. Braun), McGraw-Hill, NY, 1961.

[9] H. Faure, "Monte-Carlo and quasi-Monte-Carlo methods for numerical integration", Combinatorial & Computational Mathematics (Pohang, 2000), 1–12, World Sci. Publishing, River Edge, NJ, 2001

[10] P. Hellekalek, "On the Assessment of Random and Quasi-Random Point Sets", in *Random and Quasi-Random Point Sets*, (Eds. P. Hellekalek and G. Larcher), Springer-Verlag, NY, 1998.

[11] A. Papageorgiou and J.G. Traub, "Faster Evaluation of Multidimesional Integrals", *Computers in Physics*, pp. 574-578, Nov., 1997.

[12] S. Paskov and J.G. Traub, "Faster Valuation of Financial Derivatives", *Journal of Portfolio Management*, Vol. 22:1, pp. 113-120, Fall, 1995.

[13] A.B. Owen, "Monte Carlo Extension of Quasi-Monte Carlo", *1998 Winter Simulation Conference Proceedings*, (D. J. Medieiros, E.F. Watson, M. Manivannan, and J. Carson, Eds.), pp. 571–577.

[14] V. Koltchinskii, C.T. Abdallah, M. Ariola, P. Dorato and D. Panchenko, "Improved Sample Complexity Estimates for Statistical Learning Control of Uncertain Systems", *IEEE Trans. Automatic Control*, vol.45, no.12, pp.2383-2388, Dec. 2000.

[15] T. Alpcan, T. Basar, and R. Tempo, "Randomized Algorithms for Stability and Robustness Analysis of High Speed Communication Networks", *IEEE Trans. on Control Systems Technology*, submitted, July 2002.

[16] M.S. Branicky, S.M. LaValle, K. Olson, L. Yang, "Deterministic vs. Probabilistic Roadmaps", *IEEE Trans. on Robotics and Automation*, submitted, Jan. 2002.

[17] V. Koltchinskii, C.T. Abdallah, M. Ariola, P. Dorato and D. Panchenko, "Statistical learning control of uncertain systems: It is better than it seems", Tech. Rep.EECE 99-001, EECE Dept., The University of New Mexico, Feb. 1999.

[18] F. Dabbene, *Randomized Algorithms for Probabilistic Robustness Analysis and Design*, PhD Thesis, Politecnico Di Torino, Italy, 1999.