The Generalized Fibonacci Grid as Low-Discrepancy Point Set for Optimal Deterministic Gaussian Sampling

DANIEL FRISCH
UWE D. HANEBECK

We propose a multivariate Gaussian sampling scheme. The samples exhibit an "optimal deterministic" configuration. This entails better quadrature or cubature results than with random or quasi-random samples. Our sampling is based on the generalized Fibonacci grid that makes the remarkable properties of the well-known two-dimensional Fibonacci grid applicable in higher dimensions. Two options for generating the multivariate generalized Fibonacci grid are presented, based on a rotated grid and a linear programming counter, respectively. Various options for covariance matching are explored to obtain an unscented transform.

I. INTRODUCTION

A. Context

In many practical applications, such as nonlinear filtering and control, moments of nonlinear functions of Gaussian random vectors must be approximated in real-time. Mathematically, this is a multidimensional integration, or cubature, that can be computationally very expensive. Nevertheless, filters and controllers often have to run under real-time constraints. A standard way to perform such integration is through Monte Carlo simulation using random samples. However, the convergence rate with independent samples is quite poor. Variance reduction techniques help to improve the efficiency of stochastic expectation value computations. After giving an overview of state-of-the-art variance reduction methods, we introduce novel Gaussian sampling schemes.

B. Considered Problem

We present a Gaussian sampling method for multivariate Gaussian densities based on a higher-dimensional generalization of the two-dimensional Fibonacci grid. See Fig. 1 for a visual comparison between random samples and proposed variance-reduced samples.

C. State-of-the-Art

Variance reduction techniques for expectation value calculations include antithetic variates [2], control variates [3], importance sampling [4], stratified sampling [5], low-discrepancy or quasi-random sampling [6], [7], moment matching [8]–[10], localized cumulative distribution (LCD)-based sampling [11], [12], and Projected Cumulative Distribution based sampling [13], [14]. These methods can also be combined, for example, LCD-based sampling with moment matching and antithetic variates [15]. In this work, we focus on Gaussian sampling and therefore present various state-of-the-art methods to obtain Gaussian samples in more detail. We comment on sampling techniques in common Gaussian estimators like the cubature Kalman filter (CKF), unscented Kalman filter (UKF), and Gaussian particle filter (GPF) and compare our proposed method against them.

The "standard" way of sampling from the normal distribution employs independent and identically distributed (iid) samples, e.g., by transforming iid uniform samples with the Box–Muller method [16]. This corresponds to standard Monte Carlo simulation. According to the central limit theorem (CLT) [17, p. 244], the standard deviation of the integration error equals the standard deviation of the integrand divided by the square root of the number of samples used [18, Sec. 2.1]. This slow convergence makes the computation inefficient.

Gauss–Hermite quadrature entails a finite set of predefined, weighted evaluation points for integration.
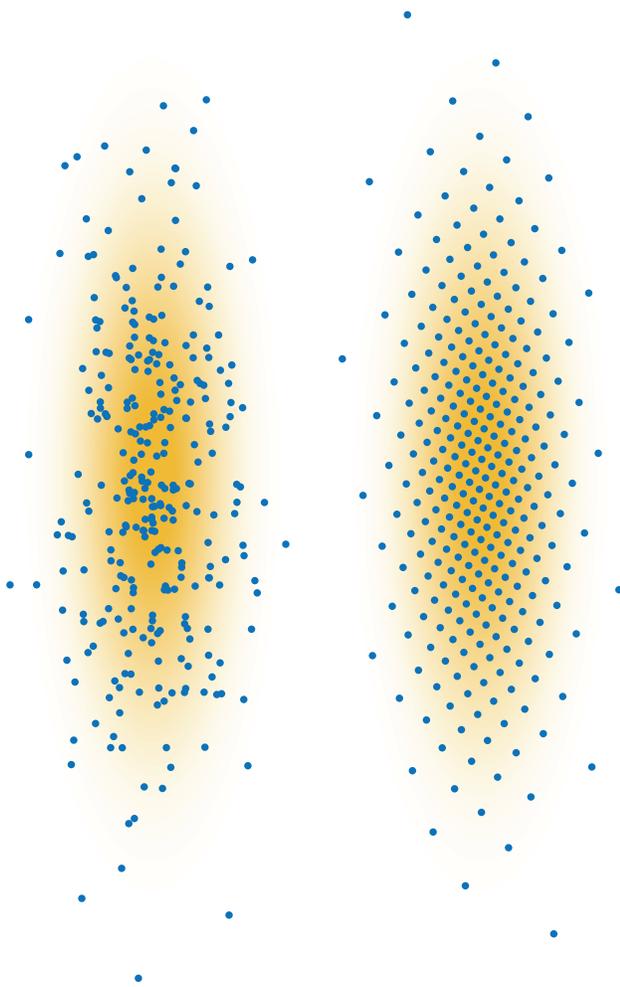
Fig. 1. Independent random samples (left) and Fibonacci-based samples (right) approximating a Gaussian density with covariance $\mathbf{C} = \mathrm{diag}([1, 0.3^2])$. A total of 300 samples are drawn.

It is ideally suited for scalar integrals of polynomial-like functions multiplied with a univariate Gaussian density function [19]. Extensions to higher dimensions require a Cartesian product of the evaluation points [20, Eq. (3.3)], [21], [22, Eq. (17)], so the number of required points increases exponentially with the number of dimensions. Kalman filters using this method for moment computation are called Gauss–Hermite quadrature filters (GHQF).

To avoid the "curse of dimensionality," one can place samples on the main axes only [23]. A more radical variant is the UKF, where only two samples, "sigma points," are placed on each coordinate axis [8], [24], plus one in the center, i.e., the number of samples is $L = 2D + 1$ for dimension D. The distances are chosen such that mean and covariance match. Very similarly, the third-order CKF places two samples on each coordinate axis, without the sample at the mode, hence $L = 2D$ [25]. The fifth-order CKF employs instead $L = 2D^2 + 1$ *weighted* samples [26], see also [27, Sec. 7], [28, Eqs. (48) and (49)]. The smallest possible sample set suitable to propagate mean and covariance has been explored in [29], [30]—it takes only $L = D+1$ or $L = D+2$ samples.

All these filters belong to the class of linear regression Kalman filters (LRKFs), introducing the second Gaussian assumption in the joint state and measurement space and thus performing an implicit linearization of the measurement equation. Particle filters avoid this and follow Bayes' theorem more directly. Thereby, the GPF [31], [32], its progressive variant [33], [34], and many other particle filters [35] need to draw medium to high numbers of samples from Gaussian priors, where our proposed Gaussian sampling technique could increase efficiency.

Now we focus on methods that allow the number of samples to be flexibly adapted to the problem and the desired accuracy. Given a suitable distance or optimality measure such as the LCD [11], optimal deterministic Gaussian sample sets can be computed using gradient optimization [36]. As this kind of sampling process is itself computationally expensive, for practical filtering it is necessary to compile a library of *standard* normally distributed samples beforehand, and transform them to the desired arbitrary Gaussian density online using the Cholesky factorization of its covariance matrix [12], [15], [37]. After such a transformation, however, the samples are usually no longer optimal as before [38, Figs. 4(a) and 5(a)].

Therefore, it is convenient to use low-discrepancy sequences. They are exactly made to achieve optimal convergence when used for numerical integration—better than the well-known convergence rate of $\frac{1}{\sqrt{L}}$ for L samples obtained with independent random samples according to the CLT. This is also referred to as quasi-Monte Carlo integration [7]—as opposed to Monte Carlo integration with independent random samples. Refer to Section IV-C2 for a formal definition of discrepancy (13) and its relation to approximate cubature (14).

Low-discrepancy sequences can, under some constraints, be transformed to densities other than uniform while preserving their low discrepancy. These point sets have already successfully been applied to nonlinear filtering problems [39], [40], yet not using a discrepancy-preserving transformation as we propose; see Fig. 8 for a visual comparison. In the one-dimensional case, equidistant samples make the best possible low-discrepancy point set. In higher dimensions, Frolov and Fibonacci grids are the only low-discrepancy sequences known to attain the theoretical optimum under very universal conditions [41]—they are, so to speak, the "lowest-discrepancy grids." Thus, we can expect them to provide results similar to the optimal deterministic samples based on the LCD with nonlinear optimization while being transformable without compromising quality.

In 2008, James Purser published *generalized* Fibonacci grids for certain higher dimensions [42]. He formulates the reason why two-dimensional grids are optimal in such a deep way that higher-dimensional generalizations become tangible. The generalized Fibonacci grid has already been applied to Gaussian sampling [1] and rejection sampling [43].

(a) Rescaled Fibonacci Grid.

(b) Rescaled Regular Grid.
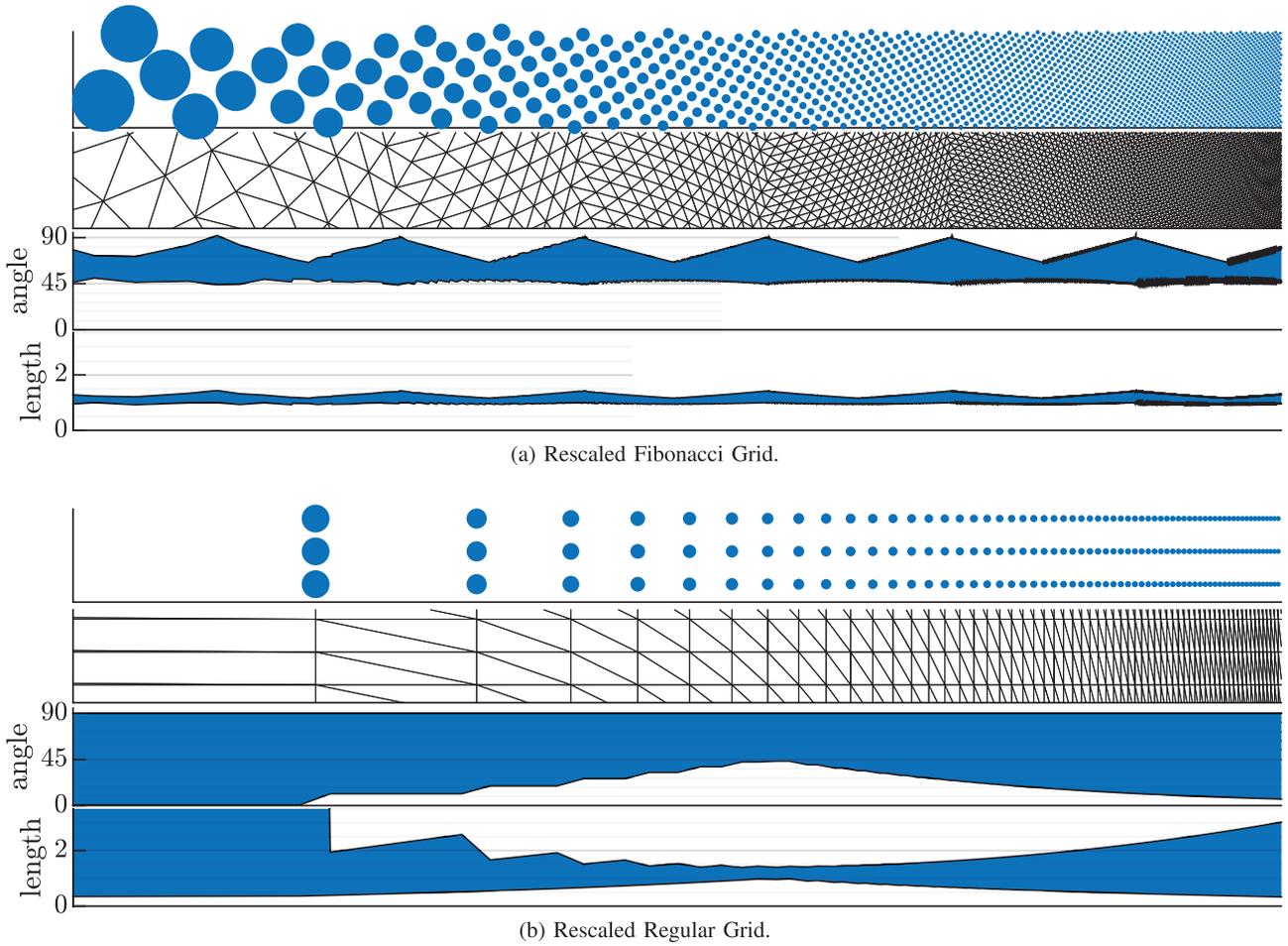
Fig. 2. Anisotropic scaling demonstration for Fibonacci grid (a) and regular grid (b). Scaling is altered exponentially along horizontal axis. Blue dots indicate grid points; black net the Delaunay triangulation of the same points. Note how in (a) the grid rearranges itself into square configurations of different sizes six times, evenly filling the space at any scaling, while in (b) there is only one square configuration (near the center) and away from that points clump together into horizontal or vertical lines, yielding bad space filling. For more quantitative assessment, maximum and minimum angle in Delaunay triangles, as well as maximum and minimum triangle side length (normalized to the side length of a square of appropriate size) are shown as well.

D. Key Idea

To produce univariate Gaussian samples, uniform samples can be transformed by the inverse Gaussian distribution. For multivariate Gaussians, this scalar transformation is applied along the directions of the eigenvectors of the covariance matrix, respectively. By doing so, the distribution of samples should stay locally homogeneous, i.e., without forming clumps or gaps. Therefore, we need a uniform point set being collision-avoiding under rescaling along certain axes.

An ideal candidate is the Fibonacci grid, as it can be anisotropically rescaled along the main axes while preserving the uniformity of points. Instead of colliding, Fibonacci grid points automatically get new neighbors, depending on the amount of rescaling. Refer to Fig. 2(a) for a visual demonstration of how the well-known two-dimensional Fibonacci grid remains uniform under inhomogeneous horizontal scaling, very much unlike the axis-aligned regular grid in Fig. 2(b). This remarkable property is what we take advantage

of in this work. Grids with equivalent properties also exist in higher dimensions—the *generalized* Fibonacci grids [42].

With a suitable mapping, these uniform samples can be transformed to an arbitrary density, similar to the well-known "inverse transform sampling" method. We introduce such a mapping for the Gaussian density. Refer to Fig. 3(a) for a visual demonstration of the mapping workflow using Fibonacci samples as compared to, e.g., a regular grid Fig. 3(b). In addition, we introduce some methods for moment matching so that the covariance of the samples is accurate to machine precision.

E. Overview

This paper is structured as follows: After explaining well-known and optimal two-dimensional uniform samples in Section II, we generalize to higher-dimensional uniform samples in Sections III and IV. Then in Section V, we explain how to obtain Gaussian instead

(a) Discrepancy-Preserving Transformation: Low-Discrepancy / Fibonacci Uniform $\rightarrow$ Arbitrary Gaussian.

(b) Same Transformation, but: Regular (*not* low-discrepancy) Uniform $\rightarrow$ Arbitrary Gaussian.

(c) Gaussian Copula Distribution (only for reference).
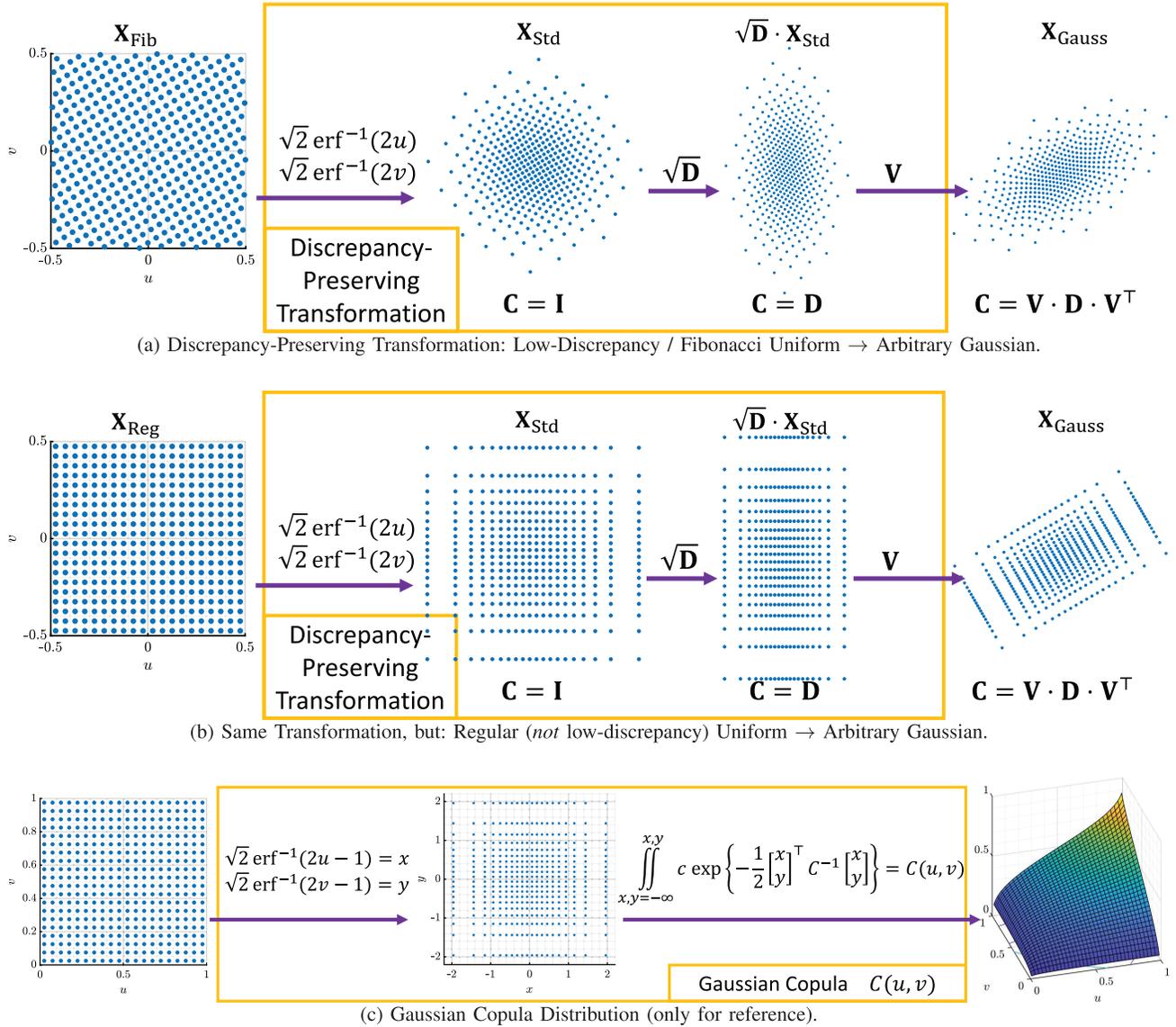
Fig. 3. (a) Transformation workflow from uniform distribution to arbitrary Gaussian. As opposed to the commonly used transformation via Cholesky factorization, see Fig. 8(a), our approach preserves the discrepancy of the input point set and is therefore better suited for low-discrepancy sequences. (b) Same transformation is applied to an axis-aligned regular grid instead of low discrepancy. Note how resulting Gaussian samples fill the space less homogeneously. (c) Gaussian copula, shown only for reference. Note that the first step in its transformation pipeline is related to what we discuss here.

of uniform samples and evaluate their optimality in Section VI.

The merits of our Gaussian samples lie in (i) providing superior coverage of the state space and (ii) free choice of the number of samples. Thus, they can improve the convergence and accuracy of algorithms that utilize Gaussian samples, e.g., sample-based Gaussian state estimation filters and controllers.

II. THE TWO-DIMENSIONAL FIBONACCI GRID

Two-dimensional Fibonacci grids have been known for a long time since they are ubiquitous in plant life. Seed heads are often arranged as a polar Fibonacci grid. Due to the size of its seeds, this is best seen in the sunflower, but a close look reveals similar structures in many other flower heads. It is well known that the two-dimensional Fibonacci grid is the best possible low-discrepancy point set [44, p. 186], [45, p. 61].

Arranging the seeds on a Fibonacci grid has two advantages. First, the space is well utilized and second, the arrangement is flexibly scalable along the radius. The former is important to utilize biological resources as efficiently as possible, and the latter is necessary because the whole thing is growing, with bigger seeds on the outside and younger, smaller seeds near the center. Although a hexagonal arrangement would make even better use of the space, this would require all seeds to always be of the same size.

In summary, the polar Fibonacci grid can be anisotropically rescaled along the radius—and the angle, for that matter, i.e., both main axes of the polar coordinate

(a) Lattice Rule – Polar.

(b) Lattice Rule – Cartesian.

(c) Lattice Rule – Gaussian.

(d) Frolov – Polar.

(e) Frolov – Cartesian.
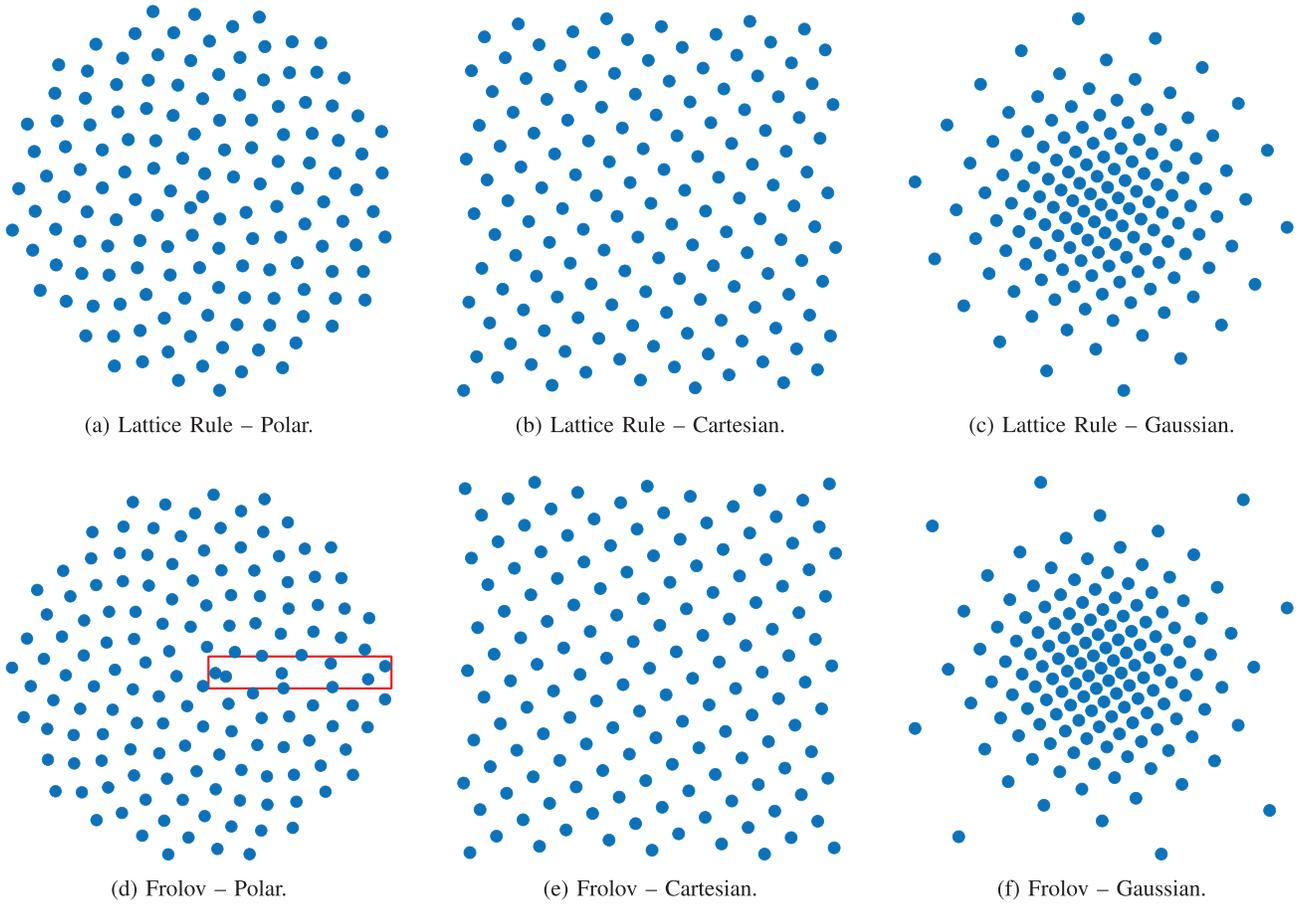
(f) Frolov – Gaussian.

Fig. 4. Fibonacci grids computed via the lattice rule (a)–(c) and Frolov method (d)–(e). Points are arranged uniformly in polar coordinates on the left, in Cartesian coordinates in the middle, and in Gaussian density on the right.
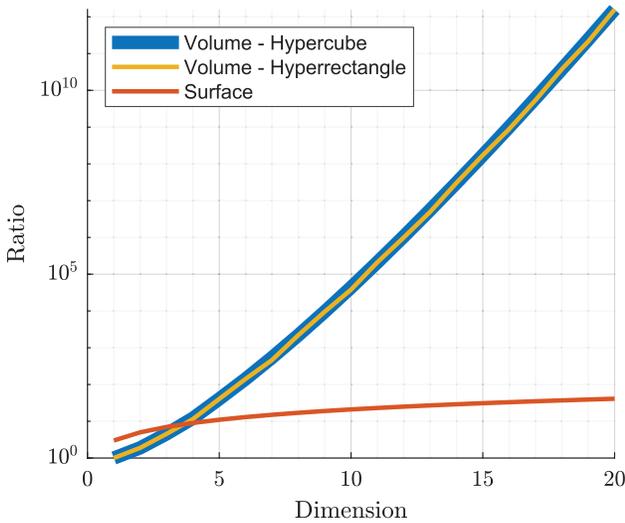
Fig. 5. Volume of the smallest hypercube (blue) and smallest hyperrectangle (yellow) that encloses the unit hypercube that is rotated by $\mathbf{V}^{\top}$. Note that there is no big difference between hypercube and hyperrectangle. Note also that the ratio for both increases exponentially with the dimension. For the linear programming method, the ratio between surface (red) and volume is more important. It increases much slower with the number of dimensions.

system. In Cartesian coordinates, the Fibonacci grid allows anisotropic scalings along the horizontal and vertical axes. This can be seen in Fig. 2(a), where the horizontal scaling of a Fibonacci grid is varied while the vertical scaling stays constant. Instead of colliding, the points change their neighborhood relationships periodically. Note how the Fibonacci grid repeatedly returns to a "regular grid" configuration, i.e., the Delaunay interior angles are $45°$ and $90°$, and the normalized side lengths are 1 and $\sqrt{2}$ over and over again, only at different scales. For comparison, the same anisotropic scaling performed on a regular axis-aligned grid does produce point collisions, Fig. 2(b).

A. Fibonacci Matrix

The Fibonacci numbers F_k are defined as [46, Sec. 6.6]

$$F_{k+1} = F_k + F_{k-1}, \qquad F_0 = 0, \qquad F_1 = 1.$$

This recurrence can be expressed with the Fibonacci matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \tag{1}$$

where Fibonacci numbers are then generated as

$$\begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} F_k \\ F_{k-1} \end{bmatrix} \ .$$

With the eigenvalue decomposition of the Fibonacci matrix

$$\mathbf{M} = \mathbf{V} \cdot \mathbf{D} \cdot \mathbf{V}^\top \ , \tag{2}$$

we define the orthogonal unitary matrix $\mathbf{V}$ containing the eigenvectors of $\mathbf{M}$, and diagonal matrix $\mathbf{D}$ containing the eigenvalues.

Note that $\mathbf{M}$ is unimodular, i.e., it consists of integers and has a unit absolute determinant. The former implies that $\mathbf{M}$ transforms integer vectors $\underline{z}$ into other integer vectors $\mathbf{M}\underline{z}$, and the latter implies that convex sets of integer vectors $\underline{z}$ stay convex after transformation, i.e., no empty holes would appear. When the entire integer lattice $\mathbb{Z}^2$ is transformed by $\mathbf{M}$, exactly the same integer lattice comes out, as lattice points are indistinguishable. Relaxing the unit determinant restriction and including intermediate configurations in between the square-lattice configurations, we can apply continuous scaling, as visualized in Fig. 2(a), where square lattice configurations are reached at six instances. But the fact that we run into square lattice configurations again and again guarantees that the points will always have a homogeneous microstructure and never collide, as in Fig. 2(b).

B. Rank-One Lattice

Mathematically, the Fibonacci grid is often represented as a rank-1 lattice rule. To produce lattice point $\underline{x}_i$, a generating vector is multiplied with an integer index i, and the result is taken modulo 1

$$\underline{x}_i = \frac{i}{F_{k+1}} \cdot \begin{bmatrix} 1 \\ F_k \end{bmatrix} \mod 1 \ , \tag{3}$$

$$i = 0, 1, \ldots, F_{k+1} - 1 \ ,$$

where F_k is the kth Fibonacci number [7, Ex. 2.8]. The result is a Cartesian Fibonacci grid with F_{k+1} samples in $[0, 1)^2$. An example with $F_k = 144$ is shown in Fig. 4(b). The grid can not only be anisotropically rescaled along its coordinate axes [as demonstrated in Fig. 2(a)] but also transformed to other coordinate systems while maintaining its packing efficiency. Transformation to polar coordinates (and proper scaling along the radius axis) yields the conspicuous sunflower pattern; see Fig. 4(a).

C. Frolov Lattice

A slightly different Fibonacci grid can be computed as a Frolov lattice. Here, the regular axis-aligned integer grid $\mathbb{Z}^2$ is rescaled with factor δ (to achieve the desired number of points L) and linearly transformed with matrix $\mathbf{T}$ (e.g., a rotation matrix). The result is then confined to the unit square $[0, 1]^2$

$$\{\underline{x}_i\}_{i=1}^L = \{\mathbf{T} \cdot \delta \cdot \underline{z} : \underline{z} \in \mathbb{Z}^2\} \cap [0, 1]^2 \ . \tag{4}$$

Now, we use the eigenvectors of the Fibonacci matrix (2) as the linear transformation $\mathbf{T}$

$$\mathbf{T} = \mathbf{V}^\top \ , \tag{5}$$

with $\mathbf{V}^\top$ meaning the transpose of $\mathbf{V}$. This again yields a two-dimensional Cartesian Fibonacci grid; see Fig. 4(e). As opposed to (3), this grid is not periodic; therefore, a transformation to polar coordinates is not smooth at the angular coordinate's transition between 0 and 2π; see the red box in Fig. 4(d).

In this work, we focus on the nonperiodic Fibonacci lattice that is computed via Frolov-like construction. The advantages of nonperiodic generalized Fibonacci grids are their visually appealing symmetry, and that they can be generated for arbitrary numbers of points. On the downside, their generation becomes more difficult in higher dimensions.

III. PURSER'S GENERALIZED FIBONACCI GRID

James Purser showed that higher-dimensional generalizations with optimality properties analogous to the two-dimensional Fibonacci grid do exist [42]. Purser's higher-dimensional Fibonacci grid is based on a new theory that captures the concept behind two-dimensional Fibonacci grids on a deep level. From that perspective, it is then easy to see how Fibonacci-type grids can be conceptualized in higher dimensions as well. The theory involves quasi-Fibonacci matrices that generalize (1) to higher dimensions. Specific constructions are stated for dimensions D with the restriction that $(2D+1)$ is a prime number.

A. The Quasi-Fibonacci Matrix

The D-dimensional quasi-Fibonacci matrix $\mathbf{M}$ according to Purser [42, Appendix A] is given by

$$[\mathbf{M}]_{i,j} = \begin{cases} 1, & i + j \leq D + 1 \\ 0, & i + j > D + 1 \end{cases} \ , \tag{6}$$

for example,

$$\mathbf{M}^{(D=2)} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \ , \quad \mathbf{M}^{(D=3)} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \ .$$

Note that $\mathbf{M}^{(D=2)}$, also (1), is known as "Fibonacci Q-Matrix" [47], but attempts to generalize to higher dimensions, e.g., in [47] are different from the concept in [42, Appendix A] that we pursue here. An eigenvalue decomposition

$$\mathbf{M} = \mathbf{V} \cdot \mathbf{D} \cdot \mathbf{V}^\top \ ,$$

again splits $\mathbf{M}$ into unitary $\mathbf{V}$ and diagonal $\mathbf{D}$. The eigenvectors $\mathbf{V}$ can also be obtained by properly normalizing the unnormalized eigenvector matrix $\mathbf{V}^u$, which is given

in closed form by [42, Eq. (A.4)]

$$[\mathbf{V}^{u}]_{i,j} = \cos\left(\frac{\pi}{2} \cdot \frac{(2i-1)\,(2j-1)}{2D+1}\right) \ ,$$

$$i, j \in \{1, 2, \ldots, D\} \ . \tag{7}$$

B. Dimensions With $(2D + 1)$ Prime

In dimensions where $(2D + 1)$ is prime, the eigenvector matrix $\mathbf{V}$ of the generalized Fibonacci matrix $\mathbf{M}$ is used in Frolov lattice creation (4) just as in the two-dimensional case (5).

C. Other Dimensions

In dimensions where $(2D + 1)$ is not prime, (6) is not well suited: One column of $\mathbf{V}$ has entries with identical fractional parts because in (7) the numerator $(2i-1)\,(2j-1)$ "interferes" with the denominator $2D+1$. However, it is possible to search for alternative matrices. An example for $D = 4$ is given in [42, Sec. 7] as

$$\mathbf{M}^{(D=4)} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \ .$$

It consists of block-diagonal replications of $\mathbf{M}^{(D=2)}$.

IV. OPTIMAL DETERMINISTIC UNIFORM FIBONACCI GRIDS

In this section, we describe how to enumerate the L samples $\underline{x}_i$ of the generalized Fibonacci grid using Frolov-like construction

$$\{\underline{x}_i\}_{i=1}^{L} = \left\{\mathbf{V}^{\top} \cdot \delta \cdot \underline{z} : \underline{z} \in \mathbb{Z}^{D}\right\} \cap \left[-\frac{1}{2}, \frac{1}{2}\right]^{D} \ ,$$

where δ specifies the number of points L approximately according to $\delta \approx L^{-1/D}$, see Section IV-C for more details. Note that to simplify notation, we changed the unit hypercube under consideration from $[0, 1]^D$ to $\left[-\frac{1}{2}, \frac{1}{2}\right]^D$. The computation can be done by (i) enumerating the grid points of a regular grid inside the rotated hypercube or (ii) enumerating the grid points of a rotated regular grid in an axis-aligned hypercube.

A. Enclosing Hypercube Counter

The most simple and obvious method works as follows: Find the smallest axis-aligned hyperrectangle or hypercube that encloses the desired rotated hypercube. Iterate through all the points of the axis-aligned hyperrectangle or hypercube, check if the point is inside or outside the rotated hypercube, and return all points that are inside [1, Alg. 1].

How large is the smallest axis-aligned unit hyperrectangle enclosing a rotated unit hypercube? To find out it
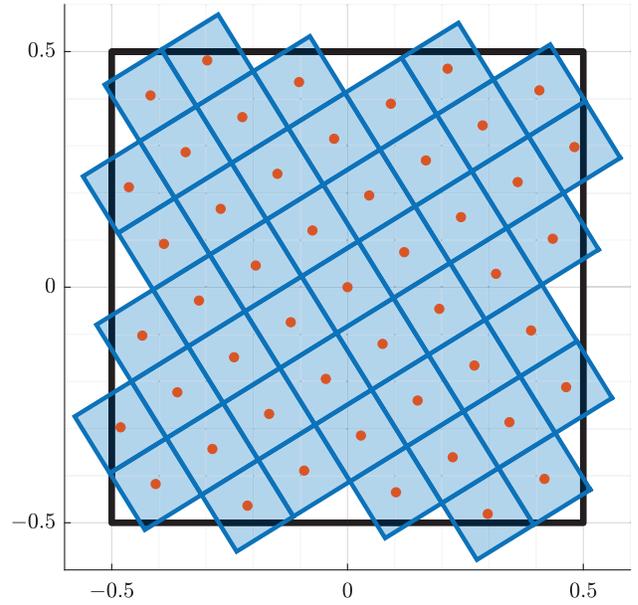
Fig. 6. Fibonacci grid with unit cells of side length $\delta = 50^{-1/2}$, calculated by $L_{\text{vol}} = 50$ according to (10). The actual number of grid points turns out to be $L_2 = 49$.

is sufficient to examine, the 2^D corners $\underline{x}_{\text{crn}}$ of the centered rotated hypercube. Their coordinates are

$$\underline{x}_{\text{crn,j}} = \mathbf{V} \cdot \underline{u}_j \ ,$$

$$\underline{u}_j \in \left\{-\frac{1}{2}, \frac{1}{2}\right\}^{D} \ ,$$

$$j \in \left\{1, 2, \ldots, 2^D\right\} \ .$$

Therefore, the side lengths β_d of the smallest enclosing hyperrectangle are

$$\beta_d = \sum_{j=1}^{D} \left|\mathbf{V}_{d,j}\right| \ , \qquad d \in \{1, 2, \ldots, D\} \ ,$$

and the side length β of the smallest enclosing hypercube is

$$\beta = \max_{d} \left\{\sum_{j=1}^{D} \left|\mathbf{V}_{d,j}\right|\right\} = \|\mathbf{V}\|_{\infty} \ .$$

For simplicity, we will focus on the hypercube instead of the hyperrectangle here, as for generalized Fibonacci matrices obtained by (6), the smallest hyperrectangle is very close to a hypercube; see Fig. 5.

We can now define a sampling vector $\underline{r} \in \mathbb{R}^{L_1}$ with centered, equidistant elements r_j that represent the grid coordinates along each dimension

$$r_j = \delta \cdot \left(j + \frac{1 - L_1}{2}\right) \ ,$$

$$j \in \{0, 1, \ldots, L_1 - 1\} \ . \tag{8}$$

After replicating $\underline{r}$, we obtain a regular grid with L_1^D elements and spacing δ, stored column-wise in the matrix $\mathbf{X}_{\text{reg}} \in \mathbb{R}^{D \times L_1^D}$. If L_1 is odd, then there will be a sample at

the origin, otherwise not. This grid is then transformed according to $\mathbf{V}^\top \mathbf{X}_{\text{reg}}$, followed by a rejection of points outside the centered unit hypercube $\left[-\frac{1}{2}, \frac{1}{2}\right]^D$. The result are L_2 Fibonacci grid points $\mathbf{X}_{\text{Fib}} \in \mathbb{R}^{D \times L_2}$ that uniformly cover the centered unit hypercube $\left[-\frac{1}{2}, \frac{1}{2}\right]^D$. By adding $\frac{1}{2}$, this can be transformed to cover the "standard" unit hypercube $[0, 1]^D$. See Fig. 6 for a visualization of the finally obtained samples inside the square for $D = 2$.

The volume ratio between the smallest enclosing hypercube β^D and a unit hypercube $1^D = 1$ increases exponentially with the dimension D; see Fig. 5. This means that the majority of samples are rejected in higher dimensions, and the method is applicable in dimensions smaller than 10 only, refer to Fig. 11(c) for more details.

B. Linear Programming Counter

In this section, we describe a method that avoids the excessive rejection of the enclosing hypercube counter. It is based on linear programming. The complexity is related more to the surface of the unit hypercube than to the volume of the enclosing hypercube; see Fig. 5. The generalized Fibonacci grid can be written as a system

of linear inequalities for integer vectors that describe a bounded polytope—so to speak, a "Diophantine inequality system," yet with real (instead of rational) coefficients. In this perspective, the first step is to find all integer vectors $\underline{z}$ such that

$$\mathbf{A}\,\underline{z} \leq \underline{b} , \qquad (9)$$

with coefficient matrix $\mathbf{A} \in \mathbb{R}^{N_c \times D}$, desired vectors $\underline{z} \in \mathbb{Z}^D$, and vector $\underline{b} \in \mathbb{R}^{N_c}$. Thereby, N_c is the number of linear inequality constraints that define the bounded polytope. For Fibonacci sampling, we define

$$\mathbf{A} = \delta \cdot \begin{bmatrix} \mathbf{V}^\top \\ -\mathbf{V}^\top \end{bmatrix} , \qquad \underline{b} = \frac{1}{2} \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} ,$$

and find all integer vectors $\underline{z}$ with $\mathbf{A}\,\underline{z} \leq \underline{b}$.

The method to obtain these points can best be described as a recursive procedure. Initially, we focus on the first coordinate z_1 and find its minimum and maximum values inside the polytope $\mathbf{A}\,\underline{z} \leq \underline{b}$ via linear programming or integer linear programming. Then, recursively for each integer value $\hat{z}_{1,k}$ between said minimum and maximum: fix $z_1 = \hat{z}_{1,k}$ as a constant temporally. In the
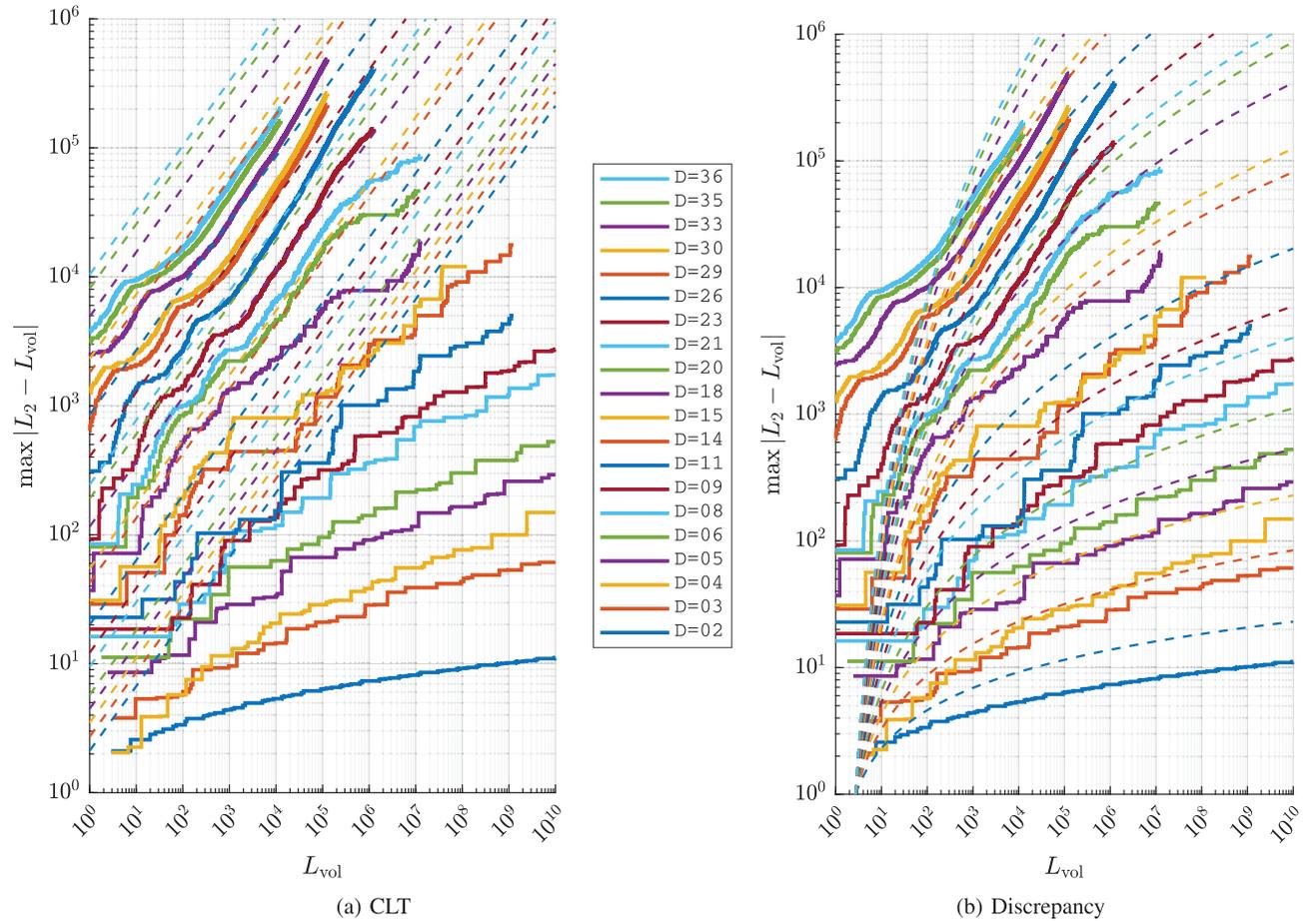
Fig. 7. Absolute difference between the expected number of points due to the volume and the obtained number of points in generalized Fibonacci grids (solid lines). Dotted lines show bounds based on the CLT (a) and the point discrepancy (b). For all sets of curves, the lowest dimension $D = 2$ appears at the bottom, with the higher dimensions following upwards according to the legend.

(a) Non-discrepancy-preserving transformation.

(b) Discrepancy-preserving transformation.
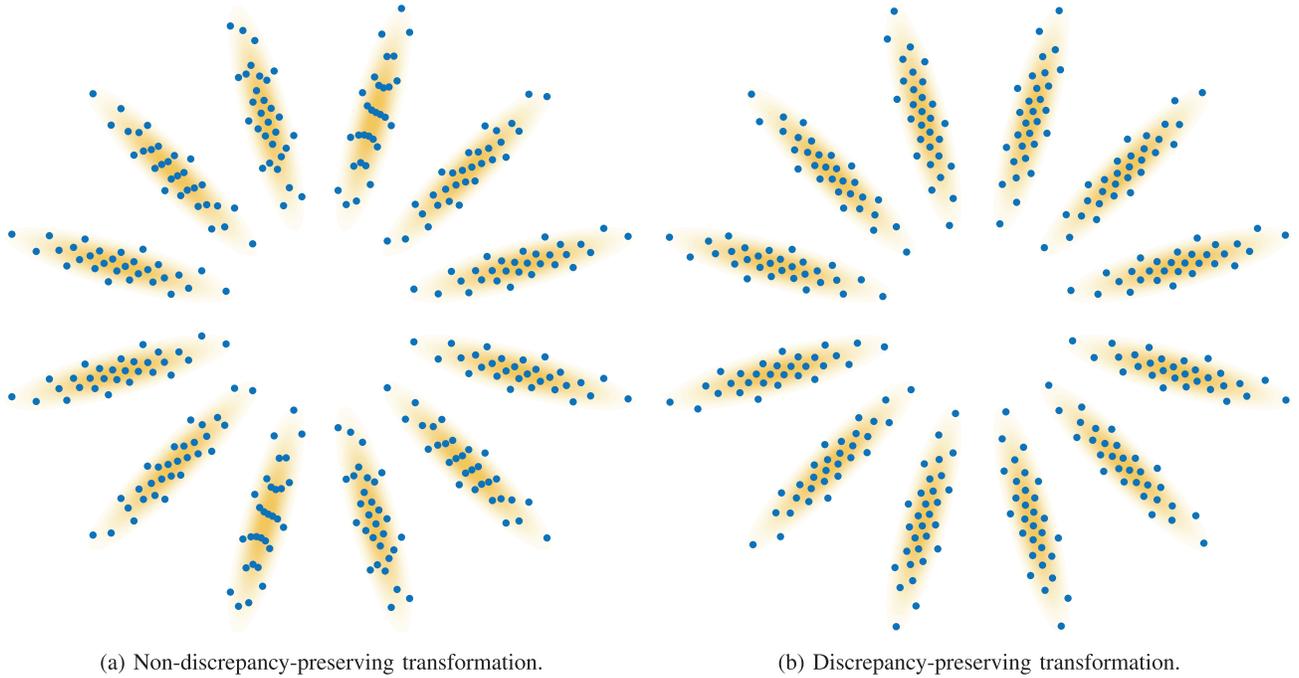
Fig. 8. Low-discrepancy point set transformed from uniform to Gaussian via nondiscrepancy-preserving transformations [39], [40] (a) and via discrepancy-preserving transformations as proposed here (b). While in (a), the sampling quality depends on the orientation of the Gaussian, in (b), it is always the same.

thus defined polytope

$$\mathbf{A} \cdot \underline{z} \le \underline{b} \ \cap \ z_1 = \hat{z}_{1,k} \ ,$$

find the minimum and maximum values of z_2. Repeat for all integer values $\hat{z}_{2,k}$ in that range. By doing so recursively for all dimensions $1 \ldots D$, all L_2 integer vectors that fulfill (9) are visited and collected in a matrix $\mathbf{Z} \in \mathbb{Z}^{D \times L_2}$. The Fibonacci point set $\mathbf{X}_{\text{Fib}} \in \mathbb{R}^{D \times L_2}$ that is uniform in $\left[-\frac{1}{2}, \frac{1}{2}\right]^D$ can then be derived from the obtained integral vectors $\mathbf{Z}$ via

$$\mathbf{X}_{\text{Fib}} = \mathbf{V}^\top \cdot \delta \cdot \mathbf{Z} \ .$$

An iterative version of this algorithm avoiding explicit recursion has been implemented in this work. For better efficiency it uses the GNU Linear Programming Kit GLPK [48].
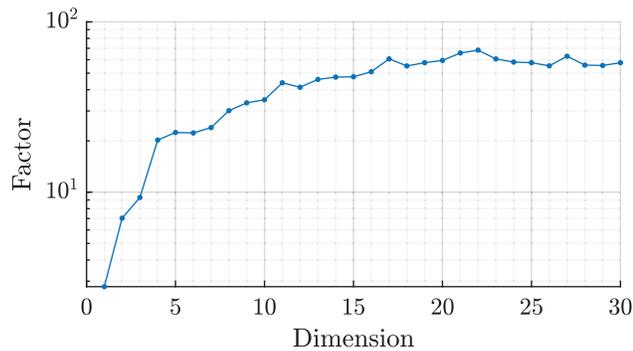
Fig. 9. Ratio of computation time between eigenvalue decomposition and the faster Cholesky factorization of random symmetric and positive definite matrices in Matlab.

Unfortunately, computational complexity increases exponentially with the dimension here as well, but slower than in the enclosing hypercube counter. This method is practical for dimensions up to about 20; see Fig. 11(c). Note that the samples, once created, can be stored and used henceforth to obtain sample sets of the same dimensionality.

C. Number of Points Obtained

In this section, we will elaborate on the number L_2 of grid points that can be expected to be inside a rotated hypercube, or, equivalently, how many rotated grid points can be expected inside an axis-aligned hypercube. Of course, a rough estimate L_{vol} is the ratio between the volume of a unit cell δ^D representing one sample and the volume of the rotated hypercube, which is one, therefore

$$L_2 \approx L_{\text{vol}} = \delta^{-D} \ ,$$

$$\delta = L_{\text{vol}}^{-1/D} \ . \tag{10}$$

However, due to the rotation between unit cells and the rotated hypercube, this estimate is not necessarily correct; see Fig. 6.

Therefore, we aim to quantify the worst case of how many "missing" or "surplus" points we can expect in L_2 compared to L_{vol}. This is related to the Gauss circle problem, where the number of two-dimensional integer lattice points inside a circle with a given radius is determined or approximated. It is even more related to the convergence rate of Monte Carlo and quasi-Monte Carlo methods. To quantify this, we define a "hypercube

(a) UKF, cubic measurement update

(b) GPF, cubic measurement update
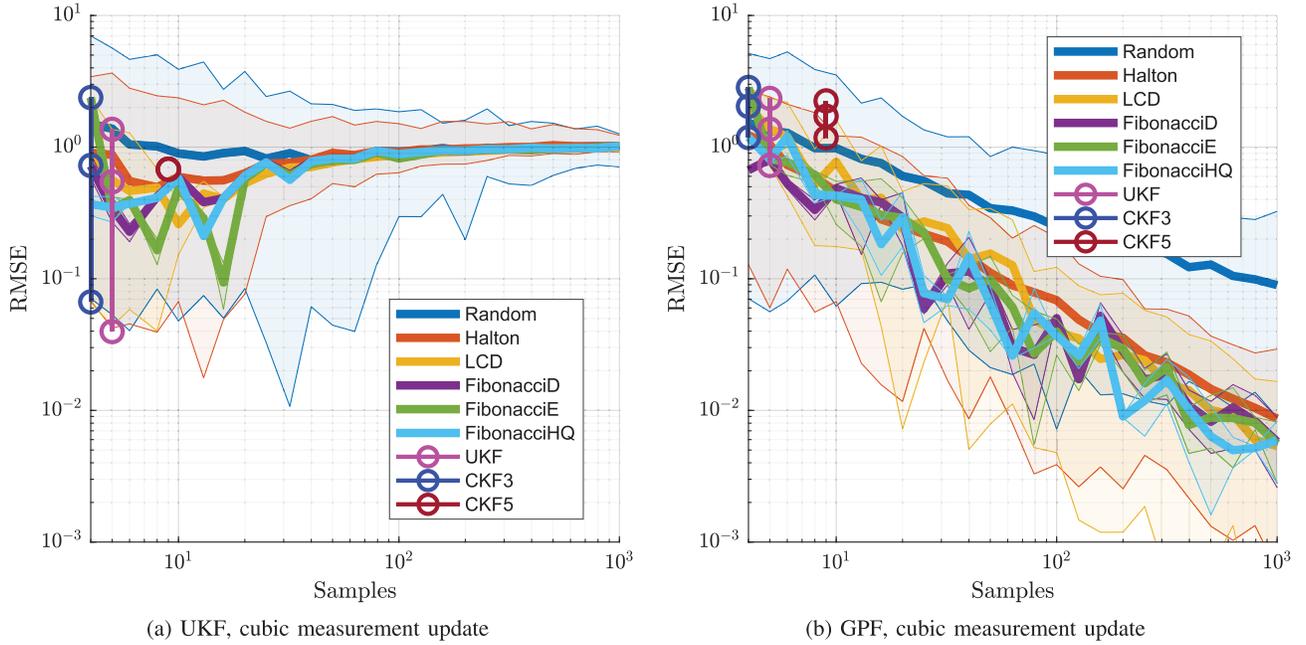
Fig. 10. Cubic measurement update, system model as explained in Section VI-A. Out of 100 trials, minimum, maximum, and mean RMSE between the estimated and true posterior mean are indicated, respectively. FibonacciD means diagonal variance matching according to (17); the same correction method has been applied to the quasi-random Halton [55] samples. FibonacciE means exact covariance matching by the eigenvalue method (19) that is however non discrepancy-preserving. FibonacciHQ means "high quality" covariance matching (21). Also included are Unscented Kalman Filter (UKF) samples [9] with scaling such that they have equal weights, CKF3 samples [25], and CKF5 samples [26].

function" $h_a(\cdot)$ centered around the origin

$$h_a(\underline{x}) = \begin{cases} 1, & |x_i| < \frac{a}{2} \quad \forall i \in [1, D] \\ 0 & \text{otherwise} \end{cases},$$

$$0 < a < 1 ,$$

and a centered unit-size hypercube

$$\mathcal{I} = \left[-\frac{1}{2}, \frac{1}{2}\right]^D .$$

1) *CLT:* The CLT states [18, Sec. 2.1] that the Monte Carlo integration error $\epsilon_{L,f}$ of an arbitrary function $f(x)$ over a unit cube $\mathcal{I}$

$$\epsilon_{L,f} = \left| \left(\frac{1}{L} \sum_{n=1}^{L} f(\underline{x}_n) \right) - \left(\int_{\mathcal{I}} f(\underline{x}) \, d\underline{x} \right) \right|$$

(with a large number L of i.i.d. uniform random samples $\underline{x}_n$ on $\mathcal{I}$) is normally distributed with standard deviation $\sigma_f \cdot L^{-1/2}$, where

$$\sigma_f = \sqrt{\int_{\mathcal{I}} \left(f(\underline{x}) - \left(\int_{\mathcal{I}} f(\underline{\tilde{x}}) \, d\underline{\tilde{x}} \right) \right)^2 \, dx} .$$

Thus, relying on a c-sigma-bound, we may assume

$$\epsilon_{L,f} \leq c \cdot \sigma_f \cdot L^{-1/2}$$

with high probability. Now we multiply both sides with L and insert h_a for f. This yields

$$\left| L_2 - L \cdot a^D \right| \leq c \cdot \sigma_h \cdot \sqrt{L} ,$$

$$\sigma_h = \sqrt{a^D - a^{2D}} ,$$

and with $L_{\text{vol}} = L \cdot a^D$, we write

$$|L_2 - L_{\text{vol}}| \leq c \cdot \sqrt{(1 - a^D) \cdot L_{\text{vol}}} .$$

For $a \to 0$, this finally becomes

$$|L_2 - L_{\text{vol}}| \leq c \cdot \sqrt{L_{\text{vol}}} ,$$

$$L_2 \geq L_{\text{vol}} - c \cdot \sqrt{L_{\text{vol}}} . \tag{11}$$

Solving this for L_{vol}, we see that we should select

$$L_{\text{vol}} \geq L_2 + \frac{c^2}{2} + \sqrt{L_2 \cdot c^2 + \frac{c^4}{4}} , \tag{12}$$

and then δ according to (10), to always obtain at least the desired number of samples L_2. Fig. 7(a) shows a numerical overview, where $c = \exp\{\frac{D+1}{4}\}$ has been selected heuristically as the y-intercept. Note that for large L and in dimensions lower than 10, the CLT is a rather conservative estimate because the generalized Fibonacci points have a lower discrepancy and therefore better convergence rate than iid samples that the CLT assumes.

2) *Discrepancy:* We give an intuitive definition of the discrepancy of a point set $\underline{\hat{x}}_i \in [0, 1]^D$, $i \in [1, 2, \ldots, L]$.

(a) Optimality, Standard Normal, $\tau = 0.5$

(b) Optimality, Gaussian with $\mathbf{C} = \mathrm{diag}([1, 0.1, 0.01])^2$, $\tau = 50$

(c) Computation Time

(d) Fastest Method
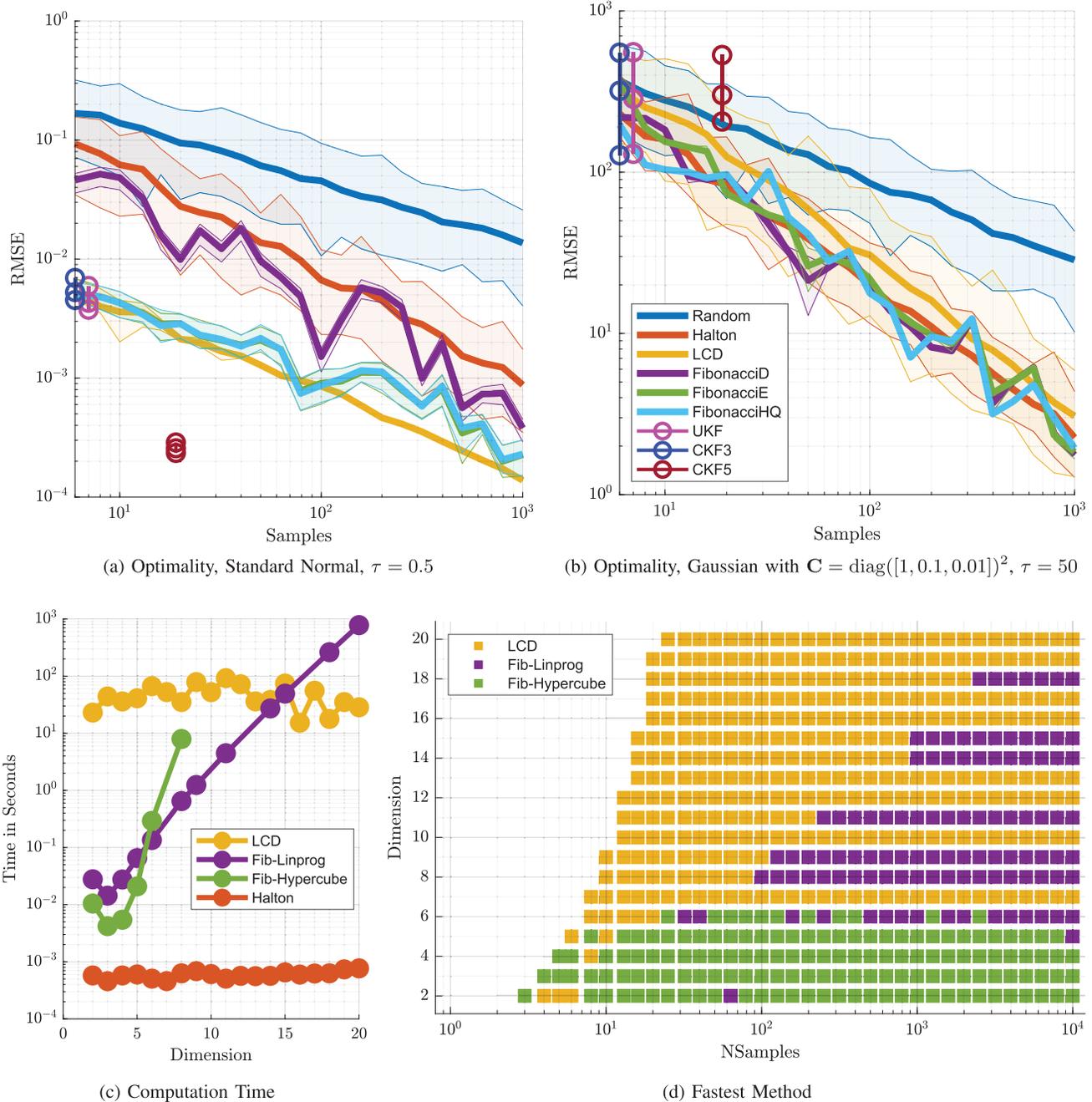
Fig. 11.   (a) Fourier-based optimality measure with isotropic Gaussian. (b) Fourier-based optimality measure with strongly nonisotropic Gaussian. The best, worst, and mean RMSE of 100 trials are shown, respectively. Note that for a given accuracy, far less Fibonacci samples than, e.g., random samples are needed. (c) Calculation times for 1 000 optimal deterministic samples. Note that samples can be generated offline and tabulated for given D and L, for subsequent real-time use. (d) The fastest optimal deterministic sampling method for given number of samples and dimension. Fibonacci samples are only given for dimensions where $2D + 1$ is prime and $D = 4$, i.e., where suitable unimodular matrices are currently known.

Consider the volume of a hyperrectangle spanned between the origin $\underline{0}$ and point $\underline{x} \in [0, 1]^D$. The proportion of points inside this hyperrectangle minus its volume yields the local discrepancy function $\Delta(\underline{x})$ of the point set. Aggregating all local discrepancies via a p-norm yields the discrepancy [7]

$$\mathrm{discr}_p = \left(\int_{\underline{x} \in [0,1]^D} |\Delta(\underline{x})|^p \right)^{\frac{1}{p}} . \quad (13)$$

According to the Koksma–Hlawka identity

$$\left| \frac{1}{L} \sum_{i=1}^{L} g(\hat{\underline{x}}_i) - \int_{[0,1]^D} g(\underline{x}) \, d\underline{x} \right| \leq \mathrm{discr}_p \cdot V(g) , \quad (14)$$

i.e., the integration error is bounded by the discrepancy of the point set times the variation $V(g)$ of the integrand, a constant that depends on the smoothness of $g(\cdot)$. Hence, the discrepancy, as a function of the number of

samples L, is a measure for integration error, similar to the $L^{-1/2}$ term in the CLT integration error estimate. A proven lower bound on the $\mathcal{L}^2$ discrepancy of any finite point set is [49, Sec. 2]

$$L \cdot \text{discr}_2 \geq c \cdot (\log L)^{(D-1)/2} \quad .$$

For $D = 2$, this is also known to be the best possible bound [49, Sec. 3]. Unfortunately, there is currently no known upper bound on the $\mathcal{L}^\infty$ discrepancy. Conjectures include

$$L \cdot \text{discr}_\infty \leq c \cdot (\log L)^{D-1} \quad ,$$

$$L \cdot \text{discr}_\infty \leq c \cdot (\log L)^{D/2} \quad ,$$

with respective unknown constants c that depend on the dimension [50, Sec. 4.2]. In Fig. 7(b), we plot the function

$$|L_2 - L_{\text{vol}}| \leq \log(L_{\text{vol}})^{\sqrt{D-1}} \quad . \tag{15}$$

It can be seen in Fig. 7(b) that (15) is an upper bound only for higher numbers of samples, but there it is a tighter bound than the CLT-based bound (11); see Fig. 7(a).

3) Removing Excess Samples: If too many samples have been generated, then the excess samples can easily be removed using the following strategy: Sort the samples with respect to the value of the first coordinate. Then, half the necessary number of points to remove are deleted at the beginning and end of that sorted list of samples, respectively. Finally, the first coordinate is stretched to fully cover the unit cube again. This is similar to the process depicted in [1, Fig. 3(c) and (d)].

Removing samples equally on both ends, instead of simply removing them at the end, preserves the symmetry of the point set and also the first moment, which is zero due to the symmetry. However, configurations with an odd number of samples can only be reduced to other odd configurations then. The same applies to even configurations. For the enclosed hypercube method, an even configuration is produced if the sampling vector (8) has an even number of entries, i.e., if L_1 is even and therefore has no entry at zero. For the linear programming counter, an odd grid is produced when the integer vectors $\underline{z} \in \mathbb{Z}$ are offset by $\frac{1}{2}$.

4) Sample Library: Since sample computation using Frolov-like methods as proposed here is quite expensive in higher dimensions, it makes sense to calculate and save the samples in advance for each dimension in an odd and an even configuration, respectively. The CLT formula can be solved in closed form (12) and hence allows direct computation of a scaling factor such that at least the wanted number of samples is generated in any case and therefore helps to avoid "trial and error" in producing the desired number of samples. In the real-time application, the respective number of samples can then be extracted from this library as described in Section IV-C3.

V. OPTIMAL DETERMINISTIC GAUSSIAN FIBONACCI GRIDS

At this point, it has been shown how to find a given number L of uniform Fibonacci samples $\mathbf{X}_{\text{Fib}} \in \mathbb{R}^{D \times L}$ in the unit hypercube $\mathcal{I}$. Now, we will see how these can be transformed into Gaussian samples with various levels of sophistication.

A. Types of Transformations

The Fibonacci grid has the unique property that point collisions are avoided under anisotropic rescaling along certain directions. We begin with reiterating two common conditions on transformations before quantifying the "discrepancy-preserving" property that we need here.

1) Rigid transformations preserve angles and distances. They are in the class of linear transformations, where the determinant of the transformation matrix is either 1 or -1. Rigid transformations are translations, rotations, and reflections.

2) Conformal maps preserve angles locally. Everywhere, the respective local Jacobian matrices are orthogonal matrices multiplied by a scalar.

3) Discrepancy-preserving transformations preserve the uniformity of Fibonacci grids. Therefore, their Jacobian matrices must be orthogonal matrices multiplied with a diagonal matrix from left. That is, the right angles between the main axes are preserved.

In the following, we describe a discrepancy-preserving mapping that transforms the uniform density on $\mathcal{I}$ to arbitrary Gaussian densities.

B. Standard Normal Samples

First of all, we apply the usual inverse transform to obtain standard normal samples

$$[\mathbf{X}_{\text{std}}]_{d,i} = \sqrt{2} \cdot \text{erf}^{-1}(2 \cdot [\mathbf{X}_{\text{Fib}}]_{d,i}) \quad ,$$

with the Gauss error function

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp\{-t^2\} \, \text{d}t \quad .$$

This transformation maps samples uniform in $\left[-\frac{1}{2}, \frac{1}{2}\right]^D$ to standard normal samples in $\mathbb{R}^D$. Refer to Fig. 4(f) for a visualization.

C. Variance Correction

Due to the discrete sample locations, the diagonal elements of the covariance matrix $\mathbf{C}_{\text{std}}$ of the resulting point set $\mathbf{X}_{\text{std}}$ are only approximately one, and the off-diagonal elements are only approximately zero. We can easily correct the former because rescaling along the coordinate axes, i.e., multiplication with a diagonal matrix, is a discrepancy-preserving transformation. So, we

calculate the variances

$$v_d = \frac{1}{L} \sum_{i=1}^{L} [\mathbf{X}_{\text{std}}]_{d,i}^2 \qquad (16)$$

and correct the variance of the samples accordingly by rescaling the coordinates individually

$$[\mathbf{X}_{\text{stdD}}]_{d,i} = \frac{[\mathbf{X}_{\text{std}}]_{d,i}}{\sqrt{v_d}} \quad . \qquad (17)$$

D. Arbitrary Gaussian Samples

Now we want to transform the standard normal samples $\mathbf{X}_{\text{stdD}}$ with a covariance of approximately $\mathbf{I}$ (the off-diagonals are not entirely zero) to a sample set with a given, arbitrary positive definite covariance $\mathbf{C} \in \mathbb{R}^{D \times D}$. Usually, this is done via the Cholesky decomposition $\mathbf{C} = \mathbf{L} \cdot \mathbf{L}^{\top}$ according to $\mathbf{X}_{\text{Gauss}} = \mathbf{L} \cdot \mathbf{X}_{\text{stdD}}$. However, this transformation is *not* discrepancy-preserving as the orthogonality of the D main axes is not being preserved; see Fig. 8(a). Therefore, we have to calculate the eigenvalue decomposition $\mathbf{V} \cdot \mathbf{D} \cdot \mathbf{V}^{\top} = \mathbf{C}$, with $\mathbf{V}$ orthogonal and $\mathbf{D}$ diagonal, and transform the samples according to

$$\mathbf{X}_{\text{Gauss1}} = \mathbf{V} \cdot \sqrt{\mathbf{D}} \cdot \mathbf{X}_{\text{stdD}} \quad . \qquad (18)$$

Refer to Fig. 3(a) for a visual description of the transformation workflow. While Cholesky decomposition requires $\frac{1}{3}D^3$ floating point operations, eigenvalue decomposition is an iterative procedure with computational complexity $\mathcal{O}(D^3)$ per iteration step. A quick test in Matlab showed that the eigenvalue decomposition takes up to 70 times longer than the Cholesky decomposition; see Fig. 9. This is a small disadvantage that we have to accept when working with Fibonacci grids or other low-discrepancy point sets.

E. Fast Cholesky Covariance Correction

Because the off-diagonals of $\mathbf{C}_{\text{stdD}}$ are not exactly zero, $\mathbf{C}_{\text{Gauss1}}$ in (18) will not mach $\mathbf{C}$ exactly. We can correct this by using a transformation that is not discrepancy-preserving. As the amount of correction is rather small, the downside that the transformation is not discrepancy-preserving should have a rather small effect. We determine $\mathbf{C}_{\text{StdD}}$ and its Cholesky decomposition $\mathbf{L}_{\text{StdD}}$

$$\mathbf{C}_{\text{StdD}} = \frac{1}{L} \cdot \mathbf{X}_{\text{StdD}} \cdot \mathbf{X}_{\text{StdD}}^{\top} = \mathbf{L}_{\text{StdD}} \cdot \mathbf{L}_{\text{StdD}}^{\top}$$

and perform the correction

$$\mathbf{X}_{\text{Gauss}} = \mathbf{V} \cdot \sqrt{\mathbf{D}} \cdot \mathbf{L}_{\text{StdD}}^{-1} \cdot \mathbf{X}_{\text{StdD}} \quad ,$$

resulting in unscented samples $\mathbf{X}_{\text{Gauss}}$ with matching covariance.

F. Fast Eigenvalue Covariance Correction

Instead of the Cholesky decomposition of $\mathbf{C}_{\text{stdD}}$, we can also use its eigenvalue decomposition

$$\mathbf{C}_{\text{StdD}} = \mathbf{V}_{\text{StdD}} \cdot \mathbf{D}_{\text{StdD}} \cdot \mathbf{V}_{\text{StdD}}^{\top}$$

and obtain unscented samples $\mathbf{X}_{\text{Gauss}}$ via

$$\mathbf{X}_{\text{Gauss}} = \mathbf{V} \cdot \sqrt{\mathbf{D}} \cdot \mathbf{V}_{\text{StdD}} \cdot \sqrt{\mathbf{D}_{\text{StdD}}^{-1}} \cdot \mathbf{V}_{\text{StdD}}^{\top} \cdot \mathbf{X}_{\text{StdD}} \quad . \tag{19}$$

Again, this transformation is not discrepancy-preserving, but due to the rather small amount of correction necessary, this should not cause real problems.

G. Discrepancy-Preserving Covariance Correction

Now we will derive a different covariance correction method that is discrepancy-preserving. Recall that the problem with $\mathbf{C}_{\text{stdD}}$ from (17) was that the off-diagonals are not exactly zero. That is, there are small correlations present that are expressed in inequality of eigenvalues, i.e., the Gaussian contour map looks like an ellipse (ellipsoid, hyperellipsoid) instead of a circle (sphere, hypersphere)—where the principal axes of the ellipse (ellipsoid, hyperellipsoid), i.e., the eigenvectors of the covariance matrix, do not coincide with the principal axes of the coordinate system. If we only manage to deform that covariance in a discrepancy-preserving way such that the eigenvalues match the wanted eigenvalues in $\mathbf{D}$, then we can easily find an appropriate rotation that puts the covariance in the right orientation subsequently.

Therefore, we have to find D deformations along the coordinate axes, collected in the vector $\underline{a} \in \mathbb{R}^D$, such that the eigenvalues of

$$\mathbf{C}_{\text{std},a} = \underline{a} \odot \mathbf{C}_{\text{std}} \odot \underline{a}^{\top} \quad , \qquad (20)$$

where $\odot$ is the pointwise product, i.e., the Hadamard product, match the targeted eigenvalues in $\mathbf{D}$.

Note that eigenvalues must be compared with a Wasserstein or earth mover's distance, i.e., the eigenvalues from both sets have to be associated appropriately. For real eigenvalues, this can be done by sorting both sets of eigenvalues [51, Sec. 3].

The search for $\underline{a}$ is now a gradient-based nonlinear optimization problem with D variables. Derivatives of eigenvalues can be calculated analytically [52, Eq. (11)], [53, Eq. (5)]

$$\frac{\partial \underline{\lambda}}{\partial a} = \mathbf{V}^{\top} \cdot \frac{\partial \mathbf{C}}{\partial a} \cdot \mathbf{V} \quad ,$$

where $\mathbf{V}$ is the normalized right eigenvector matrix of symmetric matrix $\mathbf{C}$, and $\underline{\lambda}$ the corresponding eigenvalues. With the optimal deformation vector $\underline{\hat{a}}$ obtained by nonlinear optimization, and the eigenvector matrix $\mathbf{V}_{\text{std},\hat{a}}$ of $\mathbf{C}_{\text{std},\hat{a}}$, we get the transformed Gaussian samples

$$\mathbf{X}_{\text{Gauss}} = \mathbf{V} \cdot \mathbf{V}_{\text{std},\hat{a}}^{\top} \cdot (\underline{\hat{a}} \odot \mathbf{X}_{\text{std}}) \quad ,$$

where $\mathbf{V}$ is again the eigenvector matrix of the wanted covariance $\mathbf{C}$.

The transformation (20) can only increase the ratio between eigenvalues. Therefore, if some eigenvalues of $\mathbf{C}$ are equal or nearly equal, the optimization cannot match the eigenvalues perfectly, and some residual distance to the desired eigenvalues remains. Again, this can be fully matched via a nondiscrepancy-preserving transformation

$$\mathbf{X}_{\text{Gauss}} = \mathbf{V} \cdot \left(\underline{b} \odot \mathbf{V}_{\text{std},\hat{a}}^{\top} \cdot (\underline{\hat{a}} \odot \mathbf{X}_{\text{std}}) \right) \quad , \qquad (21)$$

with nondiscrepancy-preserving deformation vector $\underline{b}$

$$\underline{b} = \sqrt{\text{diag}(\mathbf{D}) \oslash \text{diag}(\mathbf{V}_{\text{std},\hat{a}}^{\top} \mathbf{C}_{\text{std},\hat{a}} \mathbf{V}_{\text{std},\hat{a}})} \quad ,$$

where $\oslash$ denotes element-wise division. The amount of nondiscrepancy-preserving deformation encoded in $\underline{b}$ that is needed to match the covariance exactly is rather small, so point collisions will not occur.

VI. EVALUATION

The application we had in mind when developing this method are the "unscented transform" for nonlinear Gaussian filtering. Approximating first and second moments, i.e., means and covariances of nonlinear functions of Gaussian random variables, facilitates probabilistic Kalman filtering based on stochastic linearization, i.e., the various forms of LRKFs and also GPFs. Low-discrepancy sequences like the proposed Fibonacci grid and other deterministic sampling methods like LCD allow for better accuracy by using more samples than only $L = 2 \cdot D$ as in the standard unscented transform. In [1, Sec. IV], we used one specific nonlinear function to compare various deterministic sampling methods. After showing a similar, simple evaluation, we evaluate Gaussian sample sets based on a linear space of nonlinear functions.

A. Simple Evaluation

To evaluate different sampling methods, we define a nonlinear system model with additive noise

$$\boldsymbol{y} = \|\boldsymbol{x}\|_3^3 + \boldsymbol{v} \quad , \qquad \text{E}\{\boldsymbol{v}\} = 0 \quad , \qquad \text{E}\{\boldsymbol{v}^2\} = 30^2 \quad .$$

Given prior moments

$$\underline{x}^{\text{p}} = \begin{bmatrix} 2 \\ -2 \end{bmatrix} \quad , \qquad \mathbf{C}^{\text{p}} = \begin{bmatrix} 1^2 & 0 \\ 0 & 5^2 \end{bmatrix}$$

and the measurement

$$\hat{y} = 30 \quad ,$$

we compute a UKF as well as a Gaussian Filter update step with various methods for Gaussian sampling.

For the UKF case, we take Gaussian samples $\underline{\hat{x}}_i$, $i \in \{1, \ldots, L\}$ from the prior Gaussian $\mathcal{N}(x; \underline{x}^{\text{p}}, \mathbf{C}^{\text{p}})$ using the respective sampling method, insert them into the measurement equation, add samples from the

measurement noise, compute the empirical moments in the joint state and measurement space

$$\underline{z}^{\text{p}} = \begin{bmatrix} \underline{x}^{\text{p}} \\ \underline{y}^{\text{p}} \end{bmatrix} \quad , \qquad \mathbf{C}^z = \begin{bmatrix} \mathbf{C}^{\text{p}} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy} \end{bmatrix} \quad ,$$

and obtain the posterior estimate

$$\underline{x}^{\text{e}} = \underline{x}^{\text{p}} + \mathbf{C}_{xy} \mathbf{C}_{yy}^{-1} (\hat{y} - \underline{y}^{\text{p}}) \quad .$$

Refer to Fig. 10(a) for a quantitative evaluation of the root mean square error (RMSE) of the estimated posterior. Note that the real bottleneck here is the second Gaussian assumption, i.e., the statistical linearization between state space x and measurement y.

For GPF-style filtering, we take again L Gaussian samples $\underline{\hat{x}}_i$ from the prior Gaussian $\mathcal{N}(x; \underline{x}^{\text{p}}, \mathbf{C}^{\text{p}})$ using the respective sampling method and apply individual sample weights w_i according to the likelihood function value at the respective sample

$$w_i \propto \mathcal{N}\left(\hat{y}; \|\underline{\hat{x}}_i\|_3^3, \mathbf{C}^v\right) \quad , \qquad \sum_{i=1}^{L} w_i = 1.$$

The posterior mean is then approximated as the empirical average of the weighted samples. Refer to Fig. 10(b) for a quantitative evaluation of the RMSE of the estimated posterior.

B. Measure of Quality

In this section, we will define a general quality measure for Gaussian samples. It describes how well the expected values of Gaussian random variables are estimated using the respective sample sets. Thereby, instead of focusing on one specific nonlinear function, e.g., $\|\boldsymbol{x}\|_3^3$

$$\Theta = \left| \text{E}\left\{ \|\boldsymbol{x}\|_3^3 \right\} - \hat{\text{E}}\left\{ \|\boldsymbol{x}\|_3^3 \right\} \right| \quad ,$$

where $\text{E}\{\cdot\}$ is the true expected value, $\hat{\text{E}}\{\cdot\}$ is the sample approximation, and Θ is an optimality measure for the employed samples, we take a broad class of smooth nonlinear functions into account.

1) Harmonic Expectations: The expectation value of the function $g(\cdot)$ of a random variable $\boldsymbol{x}$ is

$$\text{E}\{g(\underline{x})\} = \int_{\mathbb{R}^D} g(\underline{x}) \cdot f(\underline{x}) \, d\underline{x},$$

where $\underline{x} \sim f(\underline{x})$, and $g(\cdot)$ is a smooth nonlinear function. Numerical approximation of this integral with unweighted samples $\underline{\hat{x}}_i$, $i \in \{1, \ldots, L\}$ goes like

$$\hat{\text{E}}\{g(\underline{x})\} = \int_{\mathbb{R}^D} g(\underline{x}) \cdot \hat{f}(\underline{x}) \, d\underline{x}$$

$$= \int_{\mathbb{R}^D} g(\underline{x}) \cdot \frac{1}{L} \sum_{i=1}^{L} \delta(\underline{x} - \underline{\hat{x}}_i) \, d\underline{x}$$

$$= \frac{1}{L} \sum_{i=1}^{L} g(\underline{\hat{x}}_i) \quad .$$

To obtain a representative comparison, we take into account all possible smooth functions $g(\underline{x})$ by using a Fourier basis

$$g_{\underline{t}}(\underline{x}) = \exp\{i\underline{t} \cdot \underline{x}\}$$
$$= \cos(\underline{t} \cdot \underline{x}) + i\sin(\underline{t} \cdot \underline{x}) \ . \tag{22}$$

The true expected value of a harmonic function of $\boldsymbol{x}$,

$$\mathrm{E}\{g_{\underline{t}}(\boldsymbol{x})\} = \int_{\mathbb{R}^D} g_{\underline{t}}(\underline{x}) \cdot f(\underline{x}) \, d\underline{x}$$
$$= \int_{\mathbb{R}^D} \exp\{i\underline{t} \cdot \underline{x}\} \cdot f(\underline{x}) \, d\underline{x}$$
$$= F(\underline{t}) \ ,$$

is the characteristic function $F(\underline{t})$ of the density $f(\underline{x})$. Furthermore, the approximated expectation

$$\hat{\mathrm{E}}\{g_{\underline{t}}(\boldsymbol{x})\} = \frac{1}{L} \sum_{i=1}^{L} g_{\underline{t}}(\hat{\underline{x}}_i)$$
$$= \frac{1}{L} \sum_{i=1}^{L} \exp\{i\underline{t} \cdot \hat{\underline{x}}_i\}$$
$$= \hat{F}(\underline{t})$$

is the characteristic function of $\hat{f}(\underline{x})$.

2) *Distance Measure:* Now by averaging over all spatial frequencies in domain $\mathcal{T}$

$$\Theta^2 = \int_{\mathcal{T}} \left| \mathrm{E}\{g_{\underline{t}}(\boldsymbol{x})\} - \hat{\mathrm{E}}\{g_{\underline{t}}(\boldsymbol{x})\} \right|^2 \, d\underline{t}$$
$$= \int_{\mathcal{T}} \left| F(\underline{t}) - \hat{F}(\underline{t}) \right|^2 \, d\underline{t} \ , \tag{23}$$

we obtain distance measure Θ that quantifies the approximation error of expectation values in the function space of band-limited nonlinear functions. In other words, Θ quantifies the average accuracy of an expectation value estimate of a sample set for band-limited functions. Note that (23) computes the square sum of the real and imaginary parts from (22), representing cosine and sine parts appropriately, according to

$$|a + ib|^2 = (a + ib)(a - ib) = a^2 + b^2 \ .$$

We consider spatial frequencies $\underline{t}$ in a certain domain $\mathcal{T}$,

$$\mathcal{T} = \left\{ \underline{t} \ \middle| \ \bigcap_{d=1}^{D} t^{(d)} \in [-\tau, \tau] \right\} \ , \tag{24}$$

around the origin, i.e., we focus on all sufficiently smooth functions $g(\underline{x})$, where the Fourier transform $G(\underline{x})$ does not have significant energy outside $\mathcal{T}$. From the Nyquist–Shannon sampling theorem, we can try to derive a suitable bound for τ. Given L optimal deterministic uniform samples on $[0, 1]$, their spacing is about L^{-1}. When transforming these to arbitrary densities by inverse transform sampling, the spacing of the transformed samples at the point of maximum density is $(L \cdot f_{\mathrm{mode}})^{-1}$, where f_{mode}

is the maximum derivative of the cumulative density, i.e., the density value at the mode. Therefore, the "sampling rate" is $L \cdot f_{\mathrm{mode}}$ at the mode, and lower elsewhere. Thus, according to Nyquist–Shannon, we may conclude that the maximum spatial frequency that can be represented by these samples is $\tau = L \cdot f_{\mathrm{mode}}/2$. In higher dimensions, the hypercubic cell representing one sample has volume L^{-1}. Using the inverse function theorem, the volume of this cell is transformed by the inverse of the Jacobian determinant $1/\det(\mathbf{J}(\underline{x}))$ of the mapping function that maps $f(\underline{x})$ to a uniform density. As a result, the new side length is $(L \cdot \det(\mathbf{J}(\underline{x})))^{-1/D}$, and we can choose $\tau = (L \cdot \det(\mathbf{J}(\underline{x})))^{1/D}/2$.

In our application, $f(\underline{x})$ is a Gaussian density

$$f(x) = \mathcal{N}\left(\underline{x}; \underline{\mu}, \mathbf{C}\right)$$
$$= \frac{1}{\sqrt{|2\pi\,\mathbf{C}|}} \exp\left\{-\frac{1}{2}\left(\underline{x} - \underline{\mu}\right)^\top \mathbf{C}^{-1} \left(\underline{x} - \underline{\mu}\right)\right\} \ ,$$

with $\underline{\mu} \in \mathbb{R}^D$, and positive definite component covariance matrix $\mathbf{C} \in \mathbb{R}^{D \times D}$. The characteristic function of $f(\underline{x})$ is

$$F(\underline{t}) = \exp\left\{i\,\underline{\mu} \cdot \underline{t} - \frac{1}{2}\underline{t}^\top \mathbf{C}\underline{t}\right\} \ .$$

To simplify the solution of (23), we choose

$$\underline{\mu} = \underline{0} \ ,$$
$$\mathbf{C}_k = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_D^2) \tag{25}$$

without restriction of generality, as the basis system and origin of the coordinate system can always be chosen appropriately. We obtain

$$\Theta^2 = \int_{\mathcal{T}} \left| \exp\left\{-\frac{1}{2}\underline{t}^\top \mathbf{C}\underline{t}\right\} - \frac{1}{L} \sum_{i=1}^{L} \exp\{i\underline{t} \cdot \hat{\underline{x}}_i\} \right|^2 \, d\underline{t}$$
$$= \Theta_{xx} - 2\Theta_{xy} + \Theta_{yy} \ ,$$

where

$$\Theta_{xx} = \int_{\mathcal{T}} \exp\{-\underline{t}^\top \mathbf{C}\underline{t}\} \, d\underline{t} \ ,$$
$$\Theta_{xy} = \frac{1}{L} \int_{\mathcal{T}} \exp\left\{-\frac{1}{2}\underline{t}^\top \mathbf{C}\underline{t}\right\} \cdot \sum_{i=1}^{L} \cos(\underline{t} \cdot \hat{\underline{x}}_i) \, d\underline{t} \ ,$$
$$\Theta_{yy} = \frac{1}{L^2} \int_{\mathcal{T}} \sum_{i=1}^{L} \sum_{j=1}^{L} \exp\{i\underline{t} \cdot (\hat{\underline{x}}_i - \hat{\underline{x}}_j)\} \, d\underline{t} \ .$$

For diagonal covariances (25) and $\mathcal{T}$ according to (24), this can be simplified to

$$\Theta_{xx} = \pi^{D/2} \prod_{d=1}^{D} \frac{\mathrm{erf}(\tau \cdot \sigma_d)}{\sigma_d} \ ,$$
$$\Theta_{xy} = \frac{(2\pi)^{D/2}}{L} \sum_{i=1}^{L} \prod_{d=1}^{D} \frac{1}{\sigma_d} \exp\left\{-\frac{1}{2}\frac{x_{i,d}^2}{\sigma_d^2}\right\}$$

$$\cdot \, \text{real}\left\{\text{erf}\left(\frac{\tau\sigma_d^2 + \text{i} \cdot x_{i,d}}{\sqrt{2}\,\sigma_d}\right)\right\} \quad,$$

$$\Theta_{yy} = \frac{2^D}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L}\prod_{d=1}^{D}\frac{\sin\big((x_{i,d}-x_{j,d})\cdot\tau\big)}{x_{i,d}-x_{j,d}} \quad.$$

The integration domain could be chosen as

$$\tau = \sqrt{\frac{\pi}{2}}\cdot\left(L\cdot\prod_{d=1}^{D}\sigma_d\right)^{1/D}$$

according to considerations relating to the sampling theorem. However, we choose τ constant to make the optimality measure better comparable over different L.

C. Gaussian Sampling Comparison

Now we will compare various Gaussian sampling methods for their suitability for numerical approximation of expectation values of band-limited nonlinear functions of Gaussian densities. The state-of-the-art we compare against includes LCD samples from the nonlinear estimation toolbox [54] and the Halton sequence [55], a well-known low-discrepancy sequence, that is transformed from uniform to Gaussian just as explained for the proposed Fibonacci grids in (16).

In Fig. 11(a), we show how well the different methods can approximate the three-dimensional standard normal density, i.e., the isotropic case. We can see there that the LCD method (yellow) generally provides the best results, closely followed by the Fibonacci methods with covariance matching (cyan, green). Note that $L = 10$ LCD or Fibonacci samples with moment correction are as effective as $L = 200$ samples from the Halton sequence, or more than 1 000 iid samples.

Fig. 11(b) shows the same for an anisotropic Gaussian with covariance $\mathbf{C} = \text{diag}([1, 0.1, 0.01])^2$. Here we see that the Fibonacci and Halton samples generally provide the best results, closely followed by LCD samples. This difference compared to the standard normally distributed case is because these LCD samples are produced by anisotropic transformation of standard normal ones, where some optimality is lost. Fibonacci grids, however, can be rescaled without any quality loss.

Fig. 11(c) visualizes the computational effort to compute $L = 1\,000$ LCD samples and Fibonacci samples computed via the enclosing hypercube enumeration from Section IV-A and linear programming enumeration from Section IV-B, respectively, for various dimensions. For dimensions $D < 6$, the enclosing hypercube method is fastest; for dimensions $6 \le D \le 15$, the linear programming counter; and for $D > 15$, the LCD samples. Note that for all three methods, samples can be computed beforehand and stored for later real-time use. LCD samples have to be generated for every desired dimension D and number of samples L, respectively, while Fibonacci grids have to be generated separately for every dimension D only, because subsets of Fibonacci grids can easily be used; see Section IV-C.

Fig. 11(d) shows the overall fastest sampling method out of LCD, Fibonacci linear programming, and Fibonacci hypercube for various dimensions and various numbers of samples. Again, we find that the Fibonacci enclosing hypercube method is fastest for smaller dimensions, and LCD for higher dimensions, and Fibonacci linear programming in between. Note that for $D = 7$ and $D = 10$, where $2D + 1$ is not prime, suitable generalized Fibonacci matrices are not yet known; therefore, LCD is available only. Note also that LCD with symmetric samples from the nonlinear estimation toolbox [54] requires $L \ge 2D$.

VII. CONCLUSION

We presented a new enumeration method for Fibonacci grids that is based on linear programming. It is faster than the existing enclosing hypercube method for dimensions $D \ge 6$. Furthermore, we introduced different methods for covariance correction, including a simple and fast Cholesky correction and a slower discrepancy-preserving method. We have also investigated the possible range of enumerated points given a certain scaling factor. The evaluation that was performed for a broad function class suggests that Fibonacci samples, together with the state-of-the-art LCD samples, yield the best approximations of nonlinear Gaussian expectations.

In the future, we will look for generalized Fibonacci matrices for dimensions where $(2D + 1)$ is prime. We will also look for lattice rule versions of the generalized Fibonacci grid, as they are faster to compute, and the number of resulting samples is exactly known beforehand.

REFERENCES

[1] D. Frisch and U. D. Hanebeck
    "Deterministic Gaussian sampling with generalized fibonacci grids,"
    in *Proc. 24th Int. Conf. Inf. Fusion*, 2021, pp. 1–8.
[2] J. M. Hammersley and K. W. Morton
    "A new Monte Carlo technique: Antithetic variates,"
    Math. Proc. Cambridge Philos. Soc., vol. 52, no. 3, pp. 449–475, 1956.
[3] R. Y. Rubinstein and R. Marcus
    "Efficiency of multivariate control variates in Monte Carlo simulation,"
    Operations Res., vol. 33, no. 3, pp. 661–677, 1985.
[4] S. T. Tokdar and R. E. Kass
    "Importance sampling: A review,"
    WIREs Comput. Statist., vol. 2, no. 1, pp. 54–60, 2010.
[5] V. L. Parsons
    Stratified Sampling. New York, NY: Wiley, 2017, pp. 1–11.
[6] I. H. Sloan
    "Lattice methods for multiple integration,"
    J. Comput. Appl. Math., vol. 12, pp. 131–143, 1985.

[7] J. Dick, F. Y. Kuo, and I. H. Sloan
"High-dimensional integration: The quasi-Monte Carlo way,"
Acta Numerica, vol. 22, pp. 133–288, 2013.

[8] S. J. Julier and J. K. Uhlmann
"New extension of the Kalman filter to nonlinear systems,"
in *Proc. Signal Process., Sensor Fusion, Target Recognit. VI*, 1997, pp. 182–193.

[9] S. Julier and J. Uhlmann
"Unscented filtering and nonlinear estimation,"
Proc. IEEE, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[10] M. Roth, G. Hendeby, and F. Gustafsson
"Nonlinear Kalman filters explained: A tutorial on moment computations and sigma point methods,"
J. Adv. Inf. Fusion, vol. 11, no. 1, pp. 47–70, 2016.

[11] U. D. Hanebeck and V. Klumpp
"Localized cumulative distributions and a multivariate generalization of the Cramér-von Mises distance,"
in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2008, pp. 33–39.

[12] J. Steinbring and U. D. Hanebeck
"LRKF revisited: The smart sampling Kalman filter (S2KF),"
J. Adv. Inf. Fusion, vol. 9, no. 2, pp. 106–123, Dec. 2014.
Available: https://confcats_isif.s3.amazonaws.com/web-files/journals/entries/441_1_art_11_17020.pdf

[13] U. D. Hanebeck
"Deterministic sampling of multivariate densities based on projected cumulative distributions,"
in *Proc. 54th Annu. Conf. Inf. Sci. Syst.*, 2020, pp. 1–6.

[14] D. Prossel and U. D. Hanebeck
"Dirac mixture reduction using Wasserstein distances on projected cumulative distributions,"
in *Proc. 25th Int. Conf. Inf. Fusion*, 2022, pp. 1–8.

[15] J. Steinbring, M. Pander, and U. D. Hanebeck
"The smart sampling Kalman filter with symmetric samples,"
J. Adv. Inf. Fusion, vol. 11, no. 1, pp. 71–90, Jun. 2016.

[16] G. E. P. Box and M. E. Muller
"A note on the generation of random normal deviates,"
Ann. Math. Statist., vol. 29, no. 2, pp. 610–611, 1958.

[17] W. Feller
An Introduction to Probability Theory and Its Applications, vol. 1, 3rd ed. New York, NY: Wiley. [Online].
Available: https://www.wiley.com/en-us/An+Introduction+to+Probability+Theory+and+Its+Applications%2C+Volume+1%2C+3rd+Edition-p-9780471257080

[18] R. E. Caflisch
"Monte Carlo and quasi-Monte Carlo methods,"
Acta Numerica, vol. 7, pp. 1–49, Jan. 1998, . Available: https://www.cambridge.org/core/journals/acta-numerica/article/monte-carlo-and-quasimonte-carlo-methods/FE7C779B350CFEA45DB2A4CCB2DA9B5C#

[19] N. Steen, G. Byrne, and E. Gelbard
"Gaussian quadratures for the integrals $\int_0^\infty \exp(-x^2) f(x)\, dx$ and $\int_0^b \exp(-x^2) f(x)\, dx$,"
Math. Computation, vol. 23, no. 107, pp. 661–671, 1969.

[20] K. Ito and K. Xiong
"Gaussian filters for nonlinear filtering problems,"
IEEE Trans. Autom. Control, vol. 45, no. 5, pp. 910–927, May 2000.

[21] P. Jäckel
A Note on Multivariate Gauss–Hermite Quadrature.
London: ABN-Amro. Re, 2005.

[22] I. Arasaratnam, S. Haykin, and R. J. Elliott
"Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature,"
Proc. IEEE, vol. 95, no. 5, pp. 953–977, May 2007.

[23] M. F. Huber and U. D. Hanebeck
"Gaussian filter based on deterministic sampling for high quality nonlinear estimation,"
in *Proc. 17th IFAC World Congr.*, vol. 41, no. 2, pp. 13527–13532, 2008.

[24] S. J. Julier
"The scaled unscented transformation,"
in *Proc. Amer. Control Conf.* 2002, pp. 4555–4559.

[25] I. Arasaratnam and S. Haykin
"Cubature Kalman filters,"
IEEE Trans. Autom. Control, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.

[26] B. Jia, M. Xin, and Y. Cheng
"High-degree cubature Kalman filter,"
Automatica, vol. 49, no. 2, pp. 510–518, 2013.

[27] R. Cools and P. Rabinowitz
"Monomial cubature rules since "stroud": A compilation,"
J. Comput. Appl. Math., vol. 48, no. 3, pp. 309–326, 1993. Available: https://www.sciencedirect.com/science/article/pii/0377042793900279

[28] G. Phillips
"A survey of one-dimensional and multidimensional numerical integration,"
Comput. Phys. Commun., vol. 20, no. 1, pp. 17–27, 1980. Available: https://www.sciencedirect.com/science/article/pii/0010465580901022

[29] S. Julier and J. Uhlmann
"Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations,"
in *Proc. Amer. Control Conf.*, 2002, pp. 887–892.

[30] K. G. Papakonstantinou, M. Amir, and G. P. Warn
"A scaled spherical simplex filter (S3F) with a decreased $n + 2$ sigma points set size and equivalent $2n + 1$ unscented Kalman filter (UKF) accuracy,"
Mech. Syst. Signal Process., vol. 163, 2022, Art. no. 107433. Available: https://www.sciencedirect.com/science/article/pii/S0888327020308190

[31] J. Kotecha and P. Djuric
"Gaussian particle filtering,"
IEEE Trans. Signal Process., vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[32] V. Elvira, L. Martino, and P. Closas
"Importance Gaussian quadrature,"
IEEE Trans. Signal Process., vol. 69, pp. 474–488, 2021.

[33] U. D. Hanebeck Closas
"PGF 42: Progressive Gaussian filtering with a twist,"
in *Proc. 16th Int. Conf. Inf. Fusion*, 2013, pp. 1103–1110.

[34] J. Steinbring and U. D. Hanebeck
"Progressive Gaussian filtering using explicit likelihoods,"
in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, 2014, pp. 1–8.

[35] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp
"A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,"
IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–188, Feb. 2002.

[36] U. D. Hanebeck, M. F. Huber, and V. Klumpp
"Dirac mixture approximation of multivariate Gaussian densities,"
in *Proc. IEEE Conf. Decis. Control*, 2009, pp. 3851–3858.

[37] J. Steinbring and U. D. Hanebeck
"S2KF: The smart sampling Kalman filter,"
in *Proc. 16th Int. Conf. Inf. Fusion*, 2013, pp. 2089–2096.

[38] D. Frisch and U. D. Hanebeck
"Efficient deterministic conditional sampling of multivariate gaussian densities,"
in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2020, pp. 75–81.

[39] D. Guo and X. Wang
"Quasi-Monte carlo filtering in nonlinear dynamic systems,"

IEEE Trans. Signal Process., vol. 54, no. 6, pp. 2087–2098, Jun. 2006.

[40] A. Rahimnejad, S. A. Gadsden, and M. Al-Shabi
"Lattice Kalman filters,"
IEEE Signal Process. Lett., vol. 28, pp. 1355–1359, 2021.

[41] C. Kacwin, J. Oettershagen, M. Ullrich, and T. Ullrich
"Numerical performance of optimized Frolov lattices in tensor product reproducing Kernel Sobolev spaces,"
Found. Comput. Math., vol. 21, no. 3, pp. 849–889, Jun. 2021.

[42] R. J. Purser
"Generalized Fibonacci grids; A new class of structured, smoothly adaptive multi-dimensional computational lattices" Office note (National Centers for Environmental Prediction (U.S.)), vol. 455, pp. 1–38, May 2008.

[43] D. Frisch and U. D. Hanebeck
"Rejection sampling from arbitrary multivariate distributions using generalized Fibonacci lattices,"
in Proc. 25th Int. Conf. Inf. Fusion, 2022, pp. 1–7.

[44] S. K. Zaremba
"A remarkable lattice generated by Fibonacci numbers,"
Fibonacci Quart., vol. 8, no. 2, pp. 185–198, 1970.

[45] H. Niederreiter
et al., "Integration of nonperiodic functions of two variables by Fibonacci lattice rules,"
J. Comput. Appl. Math., vol. 51, no. 1, pp. 57–70, 1994.

[46] R. L. Graham
Concrete Mathematics: A Foundation for Computer Science. Reading, MA, USA: Addison-Wesley, 1994.

[47] H. W. Gould
"A history of the Fibonacci Q-matrix and a higher-dimensional problem,"
Fibonacci Quart., vol. 19, no. 3, pp. 250–257, 1981.

[48] A. Makhorin
"GLPK (GNU linear programming kit)," [Online]. Available: http://www.gnu.org/software/glpk/glpk.html

[49] J. Dick and F. Pillichshammer
Explicit Constructions of Point Sets and Sequences with Low Discrepancy. Berlin, Germany: De Gruyter, 2014, pp. 63–86.

[50] D. Bilyk
"Discrepancy theory and harmonic analysis,"
Uniform Distribution and Quasi-Monte Carlo Methods: Discrepancy, Integration and Applications. Berlin, Boston: De Gruyter, 2014, pp. 45–62.

[51] E. Levina and P. Bickel
"The Earth Mover's distance is the mallows distance: Some insights from statistics,"
in Proc. Eighth IEEE Int. Conf. Comput. Vision, 2001, vol. 2, Jul. 2001, pp. 251–256.

[52] D. V. Murthy and R. T. Haftka
"Derivatives of eigenvalues and eigenvectors of a general complex matrix,"
Int. J. Numer. Methods Eng., vol. 26, no. 2, pp. 293–311, 1988, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620260202. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620260202

[53] P. Lancaster
"On eigenvalues of matrices dependent on a parameter,"
Numerische Mathematik, vol. 6, no. 1, pp. 377–387, Dec 1964. [Online]. Available: https://doi.org/10.1007/BF01386087

[54] J. Steinbring
"Nonlinear estimation toolbox," [Online]. Available: https://bitbucket.org/nonlinearestimation/toolbox

[55] J. H. Halton
"On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals,"
Numerische Mathematik, vol. 2, no. 1, pp. 84–90, 1960.

Daniel Frisch received the Master's degree in electrical engineering from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, in 2017. Afterwards, he was a Ph.D. student at the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Karlsruhe Institute of Technology (KIT), Germany. His research interests include deterministic sampling, nonlinear filtering, and multilateration.

Uwe D. Hanebeck received the Ph.D. and Habilitation degrees in electrical engineering from the Technical University in Munich, München, Germany, in 1997 and 2003, respectively. He is a Chaired Professor of Computer Science with the Karlsruhe Institute of Technology (KIT), Germany, and the Director of the Intelligent Sensor Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. His research interests are in the areas of information fusion, nonlinear state estimation, stochastic modeling, system identification, and control with a strong emphasis on theory-driven approaches based on stochastic system theory and uncertainty models. He is an author and coauthor of more than 550 publications in various high-ranking journals and conferences and is an IEEE Fellow.