Mahalanobis Shell Sampling (MSS) method for collision probability computation

Ulises E. Núñez Garzón* and E. Glenn Lightsey†
Georgia Institute of Technology, Atlanta, GA, 30332

Motivated by desire for collision avoidance in spacecraft formations, and by the need for accurately computing low kinematic probabilities of collision (KPC) in spacecraft collision risk analysis, this work introduces an algorithm for sampling from non-degenerate, multidimensional normal random variables. In this algorithm, the analytical relationship between certain probability density integrals of such random variables and the chi-square distribution is leveraged in order to provide weights to sample points. In so doing, this algorithm allows direct sampling from probability density "tails" without unduly penalizing sample size, as would occur with Monte Carlo-based methods. The primary motivation for the development of this algorithm is to help in the efficient computation of collision probability measures for relative dynamic systems. Performance of this method in approximating KPC waveforms is examined for a low-dimensionality dynamic example. However, this method could be applied to other dynamic systems and for probability density integrals other than collision probability measures, allowing for efficient computation of such integrals for problems where analytical results do not exist. Therefore, this method is suggested as an alternative to random sampling algorithms such as Monte Carlo methods or the Unscented Transform.

I. Nomenclature

Mathematical symbols:

$\mathbf{0}_{n\times m}$	=	zero-valued matrix in $\mathbb{R}^{n\times m}$
$\dot{[]}$	=	First time-derivative operator
$\bar{(\cdot)}$	=	"mean" value of random variable $(\cdot)$
$\|\|\cdot\|\|$	=	Euclidean norm operator
$B\left(\,\cdot\,; a, b\right)$	=	incomplete beta function with arguments a and b
$\mathbb{B}_r\left(x\right)$	=	ball of radius r centered at x
cdf_X	=	cumulative distribution function (cdf) of random variable X
$\mathrm{cdf}_{X\|Y}$	=	conditional cdf of random variable X given outcome of random variable Y
$\mathrm{Cov}\left(\cdot\right)$	=	covariance operator
D	=	dimension of points in the unit $(D-1)$-sphere
$d_0\left(\tilde{S}\left(N\right)\right)$	=	minimum arc length between any pair of points in the sample $\tilde{S}\left(N\right)$
$d_{\max}$	=	cutoff Mahalanobis distance (MSS sample parameter)
$d'_{\max}$	=	"transition" cutoff Mahalanobis distance
$d_{\mathbf{R}}$	=	dimensionality of position states (1, 2 or 3)
$D_X\left(x\right)$	=	Mahalanobis distance of instance x of random variable X
$\mathbb{E}\left[\cdot\right]$	=	expectation operator
F_X	=	"alternative", Mahalanobis distance-based cdf for normally distributed random variable X
$h_m\left(\cdot\right)$	=	regularized incomplete beta function (with parameter m)
$\mathbb{I}_n$	=	$n\times n$ identity matrix
$\mathrm{KPC}_{i,j}\left(t\|t_0\right)$	=	KPC between agents i and j at time t, given initial conditions at time t_0
$L_d\left(X\right)$	=	d-Mahalanobis contour of normally distributed random variable X
l_i	=	characteristic length of agent i
$l_{i,j}$	=	i-j joint hard-body radius

*Graduate Research Assistant, Guggenheim School of Aerospace Engineering; ueng3@gatech.edu. AIAA Student Member.

†Professor, Guggenheim School of Aerospace Engineering; glenn.lightsey@gatech.edu. AIAA Fellow.

$\mathcal{N}(\mu, \Sigma)$	=	normal distribution with mean μ and covariance Σ
N_{shells}	=	number of shells (MSS sample parameter)
$N_{\text{samples/shell}}$	=	number of samples per shell (MSS sample parameter)
$n_{\mathbf{X}}$	=	dimensionality of dynamic state
$p(\cdot)$	=	probability of event $(\cdot)$
∂A	=	boundary of set A
pdf_X	=	probability density function (pdf) of random variable X
$\text{pdf}_{X\|Y}$	=	conditional pdf of random variable X given outcome of random variable Y
ϕ_n	=	generalized golden ratio numbers of order n
$\Phi_{i,j}(t, t_0), \Phi_{\mathbf{X}_{i,j}}(t, t_0)$	=	i-j relative state transition matrix (from time t_0 to time t)
$\mathbf{R}_i$	=	position of the center of mass of agent i
$\mathbf{R}_{i,j}$	=	i-j relative position
$\mathbb{R}^n$	=	set of all $n \times 1$ real-valued matrices
$\mathbb{R}^{n\times m}$	=	set of all $n \times m$ real-valued matrices
$\tilde{S}(N)$	=	sample of points on the unit hypersphere with size N
$\Sigma_{\mathbf{R}_{i,j}}$	=	i-j relative position covariance
$\Sigma_{\mathbf{X}_{i,j}}$	=	i-j relative state covariance
$\mathbb{S}^n_r(x)$	=	n-sphere of radius r centered at $x \in \mathbb{R}^{n+1}$
$\sup(\cdot)$	=	supremum operator
$V_d(X)$	=	d-Mahalanobis volume of normally distributed random variable X
$V^{d_2}_{d_1}(X)$	=	d_1, d_2-Mahalanobis volume of normally distributed random variable X
$V_{i,j}$	=	i-j intersection volume
x	=	instance (written in lower case) of random variable X (written in uppercase)
$\mathbf{X}_i$	=	dynamic state of agent i
$\mathbf{X}_{i,j}$	=	i-j relative dynamic state
χ^2_r	=	chi-square distribution with r degrees of freedom

Acronyms:

CARA	=	"Conjunction Analysis and Risk Assessment" group at NASA GSFC
GSFC	=	NASA Goddard Space Flight Center
KPC	=	kinematic probability of collision
MSS	=	Mahalanobis Shell Sampling algorithm
Pc	=	"probability of collision", as defined by NASA CARA (equivalent to KPC)
SFF	=	spacecraft formation flying
SSA	=	space situational awareness
SVD	=	singular value decomposition
TPc	=	"total probability of collision", as defined by NASA CARA
WPC	=	window probability of collision (equivalent in meaning to TPc)

II. Introduction

Spacecraft formation flying (SFF), which has been considered extensively since the beginning of the space age, has potential to allow for missions that require high instrumentation precision. For example, synthetic aperture radiometry, which could provide increased precision in weather forecasting and climate monitoring, may be enabled by interferometric arrays of spacecraft that synthesize large instrument apertures (specifically, by distributing a formation over regions larger than those spanned by large, monolithic spacecraft) and by using sensor fusion.[1–4] Likewise, astrophysics missions such as ASTROD-GW (" 'Astrodynamical Space Test of Relativity using Optical Devices' optimized for 'Gravitational Wave' detection"), NGO ("New Gravitational wave Observatory") and LISA ("Laser Interferometer Space Antenna") have been proposed as space-based observatories which would implement SFF to detect gravitational waves, as opposed to ground-based observatories such as LIGO ("Laser Interferometer Gravitational-wave Observatory").[5, 6]

Moreover, SFF is also particularly attractive from a space mission design perspective. In a particular, SFF can enable missions with increased system robustness, as the impact of deterioration (or even failure) of and agent in a spacecraft formation may cause performance degradation in such a mission, as opposed to causing the end of the mission.[7]

Spacecraft formation missions may also have performance improvements over their mission lifetimes by replacing failed agents or by simply adding new ones, which adds a new layer to space mission architecture options beyond traditional, monolithic spacecraft missions.[8] Additionally, missions that implement SFF have an opportunity for enhanced system flexibility through improved "adaptability, scalability, evolvability, and maintainability".[9]

Having established the desirability of spacecraft formation missions, it is essential to examine some of the challenges with realizing these missions. Specifically, many problems in formation flight arise from guidance, navigation and control (GN&C) considerations. [10, 11] According to Scharf, Hadaegh and Ploen, spacecraft formation flying is defined as a "set of more than one spacecraft whose dynamic states are coupled through a common control law".[10] Under this commonly accepted definition, for any spacecraft formation, there exists at least one agent of such formation which tracks a relative state with respect to some other agent of the formation, and the control law of the former must be a function of its relative state with respect to the latter. Hence, this definition allows for an important distinction among groups of spacecraft whose members are interrelated in some manner. For instance, spacecraft in constellations, though related operationally to each other, are not linked from the perspective of control laws, whereas spacecraft that undergo certain operations that require relative control laws (such as rendezvous, docking, or relative station-keeping, to name a few examples) are instances of spacecraft formation flying (SFF).

Generally, the focus on spacecraft formation guidance in planetary orbital environments is not to prescribe arbitrary relative trajectories, but to design "passive" relative orbits whose periodicity enables certain observation properties.[10] However, errors in state knowledge, in dynamics modeling, or in maneuver timing or pointing (none of which can be fully eliminated) create a need for constant station-keeping, without which a mission cannot maintain passive periodicity (often, not even passive stability).[12] Thus, a formation flight mission whose agents run out of propellant is eventually rendered unable to continue.[7]

Therefore, in order to extend the lifetime of a spacecraft formation, it is imperative to minimize fuel consumption by means of optimally designing and executing trajectories and station-keeping schemes. However, this in itself creates several challenges. Generally, optimal control schemes presuppose full dynamic state knowledge.[13, 14] This creates issues concerning both control coordination and state knowledge transfer. Control coordination in spacecraft formations presents many challenging architectural trades. First, executing arbitrary optimal control maneuvers would require a central spacecraft (with full dynamic state knowledge of every spacecraft in the formation) to compute control commands, and to distribute them (along with timing information) to individual spacecraft. Central spacecraft implementation is key, unless every spacecraft in the formation has full-formation knowledge and is able to synthesize control signals that are mutually consistent, just as a central spacecraft would. Thus, control coordination would be difficult to accomplish without centralized processing, which, on the other hand, raises open questions, such as deciding which spacecraft should do the following tasks in such a scheme: collect formation-level state knowledge, compute control signals, schedule maneuvers, and issue details of control signals to all the other spacecraft. Additionally, time delays in signal transmission from central spacecraft to the rest may induce instabilities in the formation.[15] Furthermore, failure of such a central spacecraft could be catastrophic to a formation flight mission.

In order to mitigate the issues of formation-wide control coordination, decentralized control schemes have been proposed for active control in spacecraft formations implementing consensus protocols based on graph theoretic methods.[16–19] In the case of a spacecraft formation, its full dynamic state comprises the full dynamic state of every spacecraft in the formation. This implies that, in general, in order to execute optimal control maneuvers, each spacecraft must have knowledge of at least itself and at least one more agent (if not several agents) in the formation. This is to be accomplished either by having each spacecraft estimate its own state and that of the spacecraft in its neighborhood (i.e. the set of other spacecraft with respect to which the present spacecraft must track a relative state), or by having each spacecraft communicate its own state estimate to the agents in its neighborhood. Requirements for controllability of distributed agent groups (based on the underlying relative state knowledge communication network structure) have been presented, as well as protocols for agent consensus, based on applications of graph-theoretic methods.[20] However, the implementation of these communication networks for spacecraft state knowledge transfer is made difficult by the presence of inherent sensor and estimator inaccuracies, and by general communication "cost", which refers to the power, attitude control and computation overhead required by each spacecraft to both know its own dynamic state and receive (or directly estimate) state knowledge of other spacecraft. All of these challenges are exacerbated for small satellites, which have limited control authority, computational power and sensor accuracy.

The challenges aforementioned showcase the need for advances in autonomy in spacecraft formations, which is a trend observed more generally across most aerospace systems.[21] Specifically, the problem of missions implementing SFF requires improvements in autonomous system technologies for the following reasons. First, ground station-centered control of spacecraft formations is not only infeasible, but undesirable. On the other hand, it would be difficult to command individual spacecraft among a group of spacecraft in close proximity. On one hand, even if a spacecraft formation has no agents in close proximity, it would still be difficult to command from the ground all agents at the same time without significant investments in ground station infrastructure development. It is difficult, logistically and financially, to have dedicated ground operations for each spacecraft in a formation. Second, in a spacecraft formation there is need for real time control, not only because maneuvers in a spacecraft formation must be very precise, but because reactivity is needed for capabilities such as collision avoidance and fault management. According to the 2015 NASA Technology Roadmaps for Robotics and Autonomous Sytems [22], the need for increased system level autonomy is clear whenever any of several criteria are met (all of which are applicable for spacecraft formations):

- when system decision making must be faster than time between inter-agent communications and much faster than time between system-human operator communications;
- when mission-critical decisions must be made on board of the system, not by human operators; and
- when decisions are better informed by data on board than decisions made exogenously with summarized data.

A system with autonomy entails that such system should be able to sense, actuate and reconfigure itself from different perspectives (e.g. mission specification, planning, executive and functional perspectives) in the absence of (or in the presence of limited) human guidance or input.[21, 23] To accomplish this, the system should carry out tasks without intervention, it should modify tasks depending on goals and execution context, and it should need to instantiate and refine tasks at execution time. Autonomy implies that the system is self-directed, since it requires the system to generate and monitor its own tasks. Therefore, a vision for advances in autonomy for SFF entails treating spacecraft formations as units, thus simplifying operations through allowing the ability to command high-level behavior of formations and then letting formations accomplish high-level commands autonomously. Consequently, spacecraft formations would attain a high degree of autonomous operations, including knowledge gathering, formation corrections and reconfigurations.

The first requirement of autonomous spacecraft formations (and, in general, of autonomous multi-agent systems) is collision avoidance, which simply holds that each agent in the formation must not be in a trajectory that would have it occupy portions of the same volume of physical space as any other agent at the same time. Collisions should be avoided in the presence of deterministic (yet possibly unknown) and stochastic perturbations to nominal dynamics. Similarly, a spacecraft formation should be able to successfully avoid collisions despite limited and imperfect sensing of the external environment. Space is a complex, partially known dynamic environment, yet spacecraft formations, as multi-agent systems, are expected to function in this environment.[24] Therefore, basic functions of agents in spacecraft formations entail characterizing the likelihood of collision with other agents in the formation, and acting accordingly so as to avoid collisions while minimizing impact to nominal trajectories.

The purpose of this work is twofold. One contribution of this work is to present an alternative way of computing, between arbitrary agents, the kinematic probability of collision, abbreviated in this work as KPC, and known to NASA as Pc [25], restricted to cases for which knowledge of the relative dynamic states are normally distributed. In the specific context of spacecraft collision probability, this subject has been investigated extensively, mostly with regards to how to compute KPC accurately and efficiently, either through direct numerical computation or through pseudo-analytical approximations, and/or by attempting to include certain nuances of the dynamics.[26–31] For the Conjunction Assessment and Risk Analysis (CARA) group at NASA Goddard Space Flight Center (GSFC), potential conjunctions with KPCs as low as 4.4E-4 are deemed “Operational Red”, which are considered high-risk events, and therefore prompt extensive analysis and briefings.[25, 32] Contemplated mitigation for such high risk events include intervening actions up to collision avoidance maneuver planning and execution.[25] In CARA’s practice, one in 1000 potential conjunctions with KPC as low as 1E-7 could become a high-risk event at some point before closest approach.[25] Further, should a collision avoidance maneuver be required, current CARA best practices recommend that the maximum KPC after such maneuver be 1E-10 or less, in order to avoid the need for follow-up maneuvers.[33] This approach demonstrates the need for high resolution when computing KPC. Thus, implementing Monte Carlo methods would require a prohibitively high number of samples in order to obtain the necessary accuracy, so it is imperative to avoid computationally costly, slow Monte Carlo methods. Schemes to compute KPC using the Unscented Transform (UT) have been proposed.[34] However, since the Unscented Transform was created to reproduce moments of a probability distribution (which are integrals taken over the entire sample space of a random variable) [35], as opposed to arbitrary integrals of the distribution (which the KPC is, as shown in this work), the Unscented Transform is not ideal as a sampling method for KPC computation. The weighted sampling method presented in this work aims to directly sample from “tails” (i.e. probabilistic outliers) of normal, nondegenerate probability distributions by leveraging 1) samples in the unit hypersphere, and 2) analytical relationships between certain integrals of normal probability distributions and the chi-square distribution. By doing so, samples can be designed so as to be capable of reproducing low-valued integrals over arbitrary regions of the original distribution.

The second purpose of this work is to enable, in future work, the efficient computation of a new collision likelihood measure that could be more operationally useful than KPC, and which has been motivated and referred to as "Total Probability of Collision (TPc)" by Frigm et. al.[36] This measure is the probability of the event that two agents may collide "at any time within an arbitrary, compact time window". This "window probability of collision" measure has a simple, yet useful physical interpretation: it is the overall risk of collision between two objects in proximity within a finite time horizon, given initial conditions of the relative state between such two objects. By contrast, the kinematic probability of collision (KPC) at a given time is the probability of the event that two agents are colliding "at that specific time". However, the "window probability of collision" measure cannot be computed directly from instantaneous relative state distribution information, since a probability density function does not retain information about which regions in its domain have been within the collision region at any one time, especially not after translation, rotation or scaling transformations. Therefore, a sampling method must be used for the computation of this probability measure. However, implementing Monte Carlo methods for this purpose would entail the same difficulties as previously described.

It is helpful to clarify that the primary outcomes of this work do not directly contribute to the space situational awareness (SSA) field. First, in this work, the objective is to compute collision probability specifically between agents in a formation; thus, focus is restricted to those interactions, irrespective of potential interactions between agents of the formation with objects (or agents) outside the formation (including external debris). This objective is different from that of NASA CARA, which is concerned with evaluating the risk (and sometimes actively avoiding) collisions between members of a small set of primary assets and members of a much larger, secondary set of objects with whom the primary assets might collide (including debris). Additionally, this work is not concerned with how the initial relative state distribution between pairs of agents is obtained, but it presupposes that such information is available, implying the outcome of an underlying SSA process.

This work is organized as follows. First, the Background section (Section III) expounds the notation used in this work, defines collision events topologically, defines the kinematic probability of collision (KPC), and lists explicit dynamic system assumptions that hold for the remainder of this work. Second, the Theory section (Section IV) relates certain integrals of normal distributions to the chi-square distribution, develops a sampling method of non-degenerate normal distributions (called the Mahalanobis Shell Sampling algorithm, or MSS) using this relationship, and applies this algorithm to develop a KPC computation method. Third, the Results and Discussion section (Section V) applies the KPC computation method to a simple, one-dimensional relative position, two-dimensional relative state example in order to the behavior of this method, and obtains insights about MSS sample parameters. Fourth, the Conclusion (see Section VI) summarizes findings and makes recommendations for further examination of this topic. Fifth, since sampling methods from the unit hypersphere are presupposed as an input to the MSS sampling algorithm, the Appendix (Section VII) examines the performance of a set of such algorithms in order to motivate the choice of algorithm when extending applications of these methods to higher dimensions.

III. Background

Before presenting the Mahalanobis Shell Sampling (MSS) method, it is appropriate to introduce some preliminaries. This section begins by elaborating on the notation used in this work. Then, collision events are formally, topologically defined. Next, the kinematic probability of collision (KPC) is characterized and discussed. Finally, general dynamic system assumptions are made that hold throughout this work.

A. Notation

1. Notation - absolute states

Let $\mathbf{X}_i \in \mathbb{R}^{n_\mathbf{X}}$ denote the dynamic state of agent i, i.e. a minimum set of variables required to fully describe the behavior of agent i for all time. Here, $n_\mathbf{X}$ denotes the dimension of the dynamic state. Let $\bar{\mathbf{X}}_i(t) \doteq \mathbb{E}[\mathbf{X}_i(t)]$ denote the expected (or "mean") value of $\mathbf{X}_i$.

Let $\mathbf{R}_i \in \mathbb{R}^{d_\mathbf{R}}$ denote the position of the center of mass of agent i in $d_\mathbf{R}$-dimensional space, and let $\bar{\mathbf{R}}_i(t) \doteq \mathbb{E}[\mathbf{R}_i(t)]$ denote the expected (or "mean") position of the center of mass of agent i. For the purposes of this work, $d_\mathbf{R}$-dimensional space is assumed to be physical, Euclidean space, i.e. $d_\mathbf{R} \in \{1, 2, 3\}$ (note: in this work, norm operations refer to the Euclidean norm [37]). Additionally, the position of the center of mass of agent i is often referred to as the "the position of agent i".

It is assumed that $\mathbf{R}_i$ is a linear combination of the components of $\mathbf{X}_i$, i.e. that there exists a mapping $L : \mathbb{R}^{n_\mathbf{X}} \longrightarrow \mathbb{R}^{d_\mathbf{R}}$ defined by the rule $\mathbf{R}_i = L(\mathbf{X}_i) = \mathbf{M}\mathbf{X}_i$ for some matrix $\mathbf{M} \in \mathbb{R}^{d_\mathbf{R} \times n_\mathbf{X}}$. In particular, if $\mathbf{X}_i$ is partitioned as

$\mathbf{X}_i^T = \left[\mathbf{R}_i^T\ ,\ \mathbf{U}_i^T\right]$ (where $\mathbf{U}_i \in \mathbb{R}^{n_\mathbf{X}-d_\mathbf{R}}$ is a vector whose components are the components of $\mathbf{X}_i$ different from those of $\mathbf{R}_i$), then $\mathbf{M}$ is given by $\mathbf{M} = \left[\mathbb{I}_{d_\mathbf{R}}\ ,\ \mathbf{0}_{d_\mathbf{R}\times(n_\mathbf{X}-d_\mathbf{R})}\right]$.

In this formulation, the state $\mathbf{X}_i\left(t\right)$ and position $\mathbf{R}_i\left(t\right)$ are random variables for every instance of time t. Therefore, the state $\mathbf{X}_i$ and position $\mathbf{R}_i$ are random processes under a continuous time formulation, and they are random sequences under a discrete time formulation.[38]

Additionally, in order to provide phenomenological clarity, whenever the state $\mathbf{X}_i$ and position $\mathbf{R}_i$ are written with a lowercase (i.e. as $\mathbf{x}_i$ and $\mathbf{r}_i$), they are meant to denote a specific, deterministic "instance" or value that the state $\mathbf{X}_i$ and position $\mathbf{R}_i$ may take on. This is particularly useful in Subsection III.B when defining collision events. Even though the positions of colliding agents are generally random variables at the time of collision, collision events are topologically defined without requiring this notion.

2. Notation - relative states

Let $\mathbf{X}_{i,j} \in \mathbb{R}^{n_\mathbf{X}}$ be defined as the difference between the dynamic states of agents i and j, i.e. $\mathbf{X}_{i,j} \doteq \mathbf{X}_i - \mathbf{X}_j$. Often, $\mathbf{X}_{i,j}$ is referred to as the "relative state $\mathbf{X}_{i,j}$". Let $\bar{\mathbf{X}}_{i,j}\left(t\right) \doteq \mathbb{E}\left[\mathbf{X}_{i,j}\left(t\right)\right]$ denote the expected (or "mean") value of $\mathbf{X}_{i,j}\left(t\right)$.

Let the relative state covariance $\Sigma_{\mathbf{X}_{i,j}} \in \mathbb{R}^{n_\mathbf{X}\times n_\mathbf{X}}$ be defined as

$$\Sigma_{\mathbf{X}_{i,j}}\left(t\right) \doteq \mathrm{Cov}\left(\mathbf{X}_{i,j}\left(t\right)\right) = \mathbb{E}\left[\left(\mathbf{X}_{i,j}\left(t\right) - \bar{\mathbf{X}}_{i,j}\left(t\right)\right)\left(\mathbf{X}_{i,j}\left(t\right) - \bar{\mathbf{X}}_{i,j}\left(t\right)\right)^T\right] \tag{III.1}$$

Let $\mathbf{R}_{i,j} \in \mathbb{R}^{d_\mathbf{R}}$ be defined as the difference between the position states of agents i and j, i.e. $\mathbf{R}_{i,j} \doteq \mathbf{R}_i - \mathbf{R}_j$. This is understood to mean that $\mathbf{R}_{i,j}$ represents the position of the center of mass of agent i relative to that of agent j. Often throughout this work, $\mathbf{R}_{i,j}$ is referred to as the "relative position $\mathbf{R}_{i,j}$". Let $\bar{\mathbf{R}}_{i,j}\left(t\right) \doteq \mathbb{E}\left[\mathbf{R}_{i,j}\left(t\right)\right]$ denote the expected (or "mean") value of $\mathbf{R}_{i,j}$.

By the assumptions of subsection III.A.1, the relative position $\mathbf{R}_{i,j}$ can be computed as a function of the relative state $\mathbf{X}_{i,j}$ as given by $\mathbf{R}_{i,j} = L\left(\mathbf{X}_{i,j}\right) = \mathbf{M}\mathbf{X}_{i,j}$, with $\mathbf{M}$ as defined in subsection III.A.1.

Let the relative position covariance $\Sigma_{\mathbf{R}_{i,j}} \in \mathbb{R}^{d_\mathbf{R}\times d_\mathbf{R}}$ be defined as

$$\Sigma_{\mathbf{R}_{i,j}}\left(t\right) \doteq \mathrm{Cov}\left(\mathbf{R}_{i,j}\left(t\right)\right) = \mathbb{E}\left[\left(\mathbf{R}_{i,j}\left(t\right) - \bar{\mathbf{R}}_{i,j}\left(t\right)\right)\left(\mathbf{R}_{i,j}\left(t\right) - \bar{\mathbf{R}}_{i,j}\left(t\right)\right)^T\right] \tag{III.2}$$

3. Notation - Conditional dependencies of relative state and its pdf

Notation III.1 (Conditional dependencies of relative state and its pdf)**.** Suppose that, for every instance of time t, the relative state $\mathbf{X}_{i,j}\left(t\right)$ has a continuous probability distribution. Suppose t belongs to some set $\mathcal{T}$ such that $\mathcal{T} \subseteq \left[t_0, t_f\right]$, where t_0 and t_f represent the lower and upper bounds, respectively, of a period of time (or horizon) of interest.

The notation $\mathbf{X}_{i,j}\left(t|t_0\right)$ entails that, for every instance $\mathbf{x}_{i,j}$ of $\mathbf{X}_{i,j}$, then

- the instance $\mathbf{x}_{i,j}\left(t\right)$ is conditionally dependent on its initial condition $\mathbf{x}_{i,j}\left(t_0\right)$, for every $t \in \mathcal{T}$, i.e.

$$\mathbf{x}_{i,j}\left(t|t_0\right) \doteq \mathbf{x}_{i,j}\left(t\middle|\mathbf{x}_{i,j}\left(t_0\right)\right) \tag{III.3}$$

- the instance $\mathbf{x}_{i,j}\left(t\right)$ is conditionally independent of the horizon end time t_f, given its initial condition $\mathbf{x}_{i,j}\left(t_0\right)$, i.e.

$$\mathbf{x}_{i,j}\left(t\middle|\mathbf{x}_{i,j}\left(t_0\right), t_f\right) = \mathbf{x}_{i,j}\left(t\middle|\mathbf{x}_{i,j}\left(t_0\right)\right) = \mathbf{x}_{i,j}\left(t|t_0\right) \tag{III.4}$$

Let $\mathrm{pdf}_{\mathbf{X}_{i,j}}$ denote the probability density function (pdf) of the relative state. Similarly, the notation $\mathrm{pdf}_{\mathbf{X}_{i,j}|t,t_0}$ entails that

- the pdf of the relative state $\mathrm{pdf}_{\mathbf{X}_{i,j}}$ is conditionally dependent on its initial value, at all instances of time, i.e. for every $\tau \in \mathcal{T}$,

$$\mathrm{pdf}_{\mathbf{X}_{i,j}|t,t_0}\left(\mathbf{X}|\tau, t_0\right) \doteq \mathrm{pdf}_{\mathbf{X}_{i,j}|t}\left(\mathbf{X}\middle|\tau, \mathrm{pdf}_{\mathbf{X}_{i,j}|t}\left(\mathbf{X}|t_0\right)\right) \tag{III.5}$$

- the pdf of the relative state $\mathrm{pdf}_{\mathbf{X}_{i,j}}$ is conditionally independent of the horizon end time t_f, given the initial pdf $\mathrm{pdf}_{\mathbf{X}_{i,j}|t}\left(\mathbf{X}|t_0\right)$, i.e. for every $\tau \in \mathcal{T}$,

$$\mathrm{pdf}_{\mathbf{X}_{i,j}|t,t_0,t_f}\left(\mathbf{X}\middle|\tau, t_0, t_f\right) = \mathrm{pdf}_{\mathbf{X}_{i,j}|t,t_0}\left(\mathbf{X}|\tau, t_0\right) \tag{III.6}$$

Note: this notation applies to the relative position $\mathbf{R}_{i,j}$ as well. ◊

Remark III.2. Throughout this work, sometimes the dependencies of the relative state $\mathbf{X}_{i,j}$ and relative position $\mathbf{R}_{i,j}$ on their initial probability distributions are not expressed, i.e. sometimes $\mathbf{X}_{i,j}\left(t|t_0\right)$ is expressed as $\mathbf{X}_{i,j}\left(t\right)$, and $\mathbf{R}_{i,j}\left(t|t_0\right)$ is expressed as $\mathbf{R}_{i,j}\left(t\right)$ be done for the sake of simplicity. However, albeit implicitly, Notation III.1 would hold in such cases.

Note: this remark also applies to moments of the relative state $\mathbf{X}_{i,j}$ and relative position $\mathbf{R}_{i,j}$, and to their means and covariances. ◊

B. Definition of a collision event

A collision event (between two agents) occurs whenever their respective physical, nonempty "volumes" in $d_{\mathbf{R}}$-dimensional space have a nonempty intersection. In other words, a collision event is understood to mean that two agents may occupy portions of the same "volume" of space at the same time. The notion of a hard-body radius (or characteristic length) is used to simplify the definition of collision events, and consequently, the computation of the kinematic probability of collision (KPC).

Definition III.3 (n-ball [39, 40])**.** The n-ball of radius r, centered at $x \in \mathbb{R}^n$, denoted by $\mathbb{B}_r\left(x\right)$, is defined as the set

$$\mathbb{B}_r^n\left(x\right) \doteq \left\{y \in \mathbb{R}^n : \|x - y\| < r, r > 0\right\} \tag{III.7}$$

Note: when the dimensionality of elements in $\mathbb{B}_r^n\left(x\right)$ is implicit, it will be referred to as $\mathbb{B}_r\left(x\right)$ for simplicity. ◊

Definition III.4 (n-sphere [41])**.** The $(n-1)$-sphere of radius r, centered at $x \in \mathbb{R}^n$, denoted by $\mathbb{S}_r^{(n-1)}\left(x\right)$, is defined as the set

$$\mathbb{S}_r^{(n-1)}\left(x\right) \doteq \left\{y \in \mathbb{R}^n : \|x - y\| = r, r > 0\right\} \tag{III.8}$$

Note: the symbol "$\mathbb{S}^{(n-1)}$" denotes $\mathbb{S}_1^{(n-1)}\left(0\right)$, i.e. the unit $(n-1)$-sphere centered at the origin. ◊

Notation III.5 (Characteristic length)**.** Let the "body of agent i" B_i be defined as the set

$$B_i \doteq \left\{x \in \mathbb{R}^{d_{\mathbf{R}}} : x \text{ is in the body of agent } i\right\} \tag{III.9}$$

Let $\mathbf{r}_i \in \mathbb{R}^{d_{\mathbf{R}}}$ denote the position of the center of mass of agent i. Then, the i^{th} characteristic length l_i is defined as

$$l_i \doteq \sup_{x \in B_i} \|x - \mathbf{r}_i\| \tag{III.10}$$

Note: $B_i \subseteq \mathbb{B}_{l_i}\left(\mathbf{r}_i\right)$, where "the ball of radius ε centered at μ", denoted by $\mathbb{B}_\varepsilon\left(\mu\right)$ (for $\mu \in d_{\mathbf{R}}$), is defined as the set

$$\mathbb{B}_\varepsilon\left(\mu\right) \doteq \left\{\mathbf{r} \in \mathbb{R}^{d_{\mathbf{R}}} : \|\mu - \mathbf{r}\| < \varepsilon, \varepsilon > 0\right\} \qquad \diamond \tag{III.11}$$

Definition III.6 (Hard-body radius simplification)**.** The body of agent i is circumscribed within the ball of radius equal to the i^{th} characteristic length l_i centered at the i^{th} center of mass $\mathbf{r}_i$, i.e. agent i is circumscribed within $\mathbb{B}_{l_i}\left(\mathbf{r}_i\right)$. Furthermore, the body of agent i is assumed to be equal to $\mathbb{B}_{l_i}\left(\mathbf{r}_i\right)$, i.e. $B_i = \mathbb{B}_{l_i}\left(\mathbf{r}_i\right)$. ◊

The hard-body radius simplification presently described is illustrated in Figure 1.

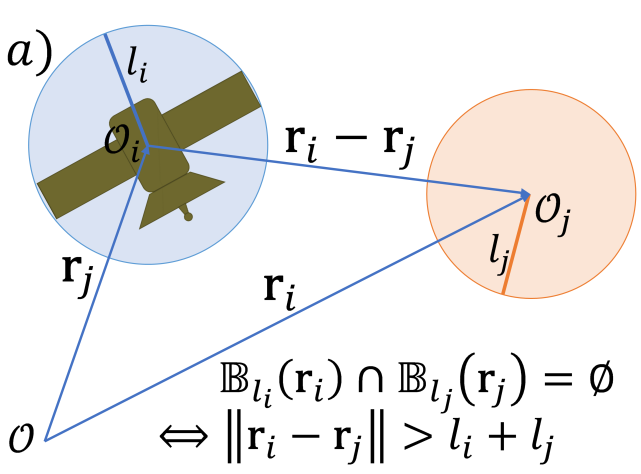
Suppose there exist two agents i and j in proximity. Through the hard-body radius simplification, the i^{th} characteristic length l_i is meant to represent a no-contact zone, meaning that the i^{th} agent does not collide with any other j^{th} agent ($i \neq j$) if no point belonging to the body of agent j^{th} becomes closer to the i^{th} center of mass of than the i^{th} characteristic length l_i. Thus, in order to avoid a collision with agent i, it is sufficient for agent j to be at least a distance l_i away from agent i, and vice versa. Using this intuition, the i-j collision event is now formally defined.

Definition III.7 (Collision event)**.** Let $\mathbf{r}_i$ and $\mathbf{r}_j$ denote the centers of mass of agents i and j. Assume the hard-body radius simplification holds (see Def. III.6). Then, a collision event between agents i and j occurs when there is a nonempty intersection between the "volumes spanned" by agent i ($\mathbb{B}_{l_i}\left(\mathbf{r}_i\right)$) and agent j ($\mathbb{B}_{l_j}\left(\mathbf{r}_j\right)$), i.e. a collision occurs whenever

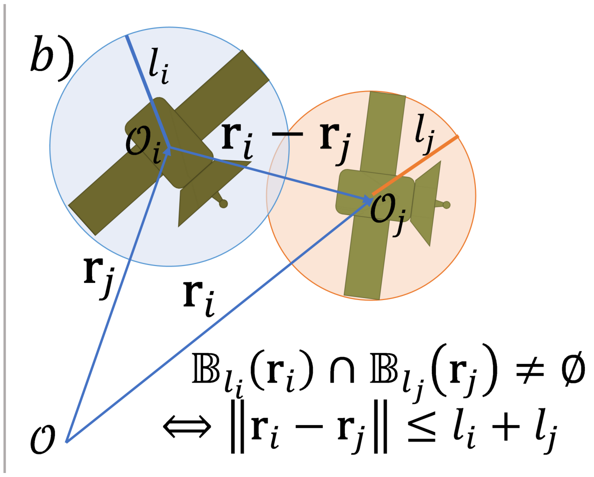
$$\mathbb{B}_{l_i}\left(\mathbf{r}_i\right) \cap \mathbb{B}_{l_j}\left(\mathbf{r}_j\right) \neq \emptyset \qquad \diamond \tag{III.12}$$

A simpler way to infer that a collision is occurring is by observing that, whenever the i^{th} and j^{th} (ball-) volumes intersect, the distance between the respective centers of mass is less than the sum of their respective hard-body radii, as seen in Figure 1. That is,

$$\mathbb{B}_{l_i}\left(\mathbf{r}_i\right) \cap \mathbb{B}_{l_j}\left(\mathbf{r}_j\right) \neq \emptyset \iff \left\|\mathbf{r}_i - \mathbf{r}_j\right\| < l_i + l_j \tag{III.13}$$



a)
l_i
$\mathcal{O}_i$
$\mathbf{r}_i - \mathbf{r}_j$
$\mathcal{O}_j$
l_j
$\mathbf{r}_j$
$\mathbf{r}_i$
$\mathcal{O}$
$\mathbb{B}_{l_i}(\mathbf{r}_i) \cap \mathbb{B}_{l_j}(\mathbf{r}_j) = \emptyset$
$\Leftrightarrow \|\mathbf{r}_i - \mathbf{r}_j\| > l_i + l_j$



b)
l_i
$\mathcal{O}_i$
$\mathbf{r}_i - \mathbf{r}_j$
l_j
$\mathcal{O}_j$
$\mathbf{r}_j$
$\mathbf{r}_i$
$\mathcal{O}$
$\mathbb{B}_{l_i}(\mathbf{r}_i) \cap \mathbb{B}_{l_j}(\mathbf{r}_j) \neq \emptyset$
$\Leftrightarrow \|\mathbf{r}_i - \mathbf{r}_j\| \leq l_i + l_j$

Fig. 1 Agents i and j under the hard-body radius simplification; a) no-collision condition, b) collision condition. (Note: the circles drawn represent d_{R}-dimensional spheres that circumscribe the bodies of agents i and j, not position pdf's.)

Notation III.8 (Joint hard-body radii)**.** Let l_i and l_j denote the characteristic lengths of agents i and j, respectively (see Not. III.5). Then, the i-j joint hard-body radius, denoted by $l_{i,j}$, is defined as

$$l_{i,j} \doteq l_i + l_j \qquad \diamond \tag{III.14}$$

Notation III.9 (Intersection volumes)**.** Let $l_{i,j}$ denote the i-j joint hard-body radius (see Not. III.8). Then, the i-j intersection volume (denoted by $V_{i,j}$) is defined as the set

$$V_{i,j} \doteq \mathbb{B}_{l_{i,j}}\left(\mathbf{0}_{d_{\mathrm{R}}}\right) = \left\{\mathbf{r} \in \mathbb{R}^{d_{\mathrm{R}}} : \|\mathbf{r}\| < l_{i,j}\right\} \qquad \diamond \tag{III.15}$$

Therefore, a collision occurs whenever any of the following equivalent statements hold:

$$\mathbb{B}_{l_i}\left(\mathbf{r}_i\right) \cap \mathbb{B}_{l_j}\left(\mathbf{r}_j\right) \neq \emptyset \iff \left\|\mathbf{r}_i - \mathbf{r}_j\right\| < l_{i,j} \iff \mathbf{r}_{i,j} \in V_{i,j} \tag{III.16}$$

Proposition III.10 (Collision condition under HBR simplification)**.** Suppose there exist two agents i and j for which the hard-body radius simplification holds (see Def. III.6), let l_i and l_j and denote their respective hard-body radii. Suppose a collision event between two agents i and j is occurring (see Def. III.7). Let $\mathbf{r}_i$ and $\mathbf{r}_j$ denote the present positions of agents i and j by, respectively. Then, the following statements are equivalent:

1) $\mathbb{B}_{l_i}\left(\mathbf{r}_i\right) \cap \mathbb{B}_{l_j}\left(\mathbf{r}_j\right) \neq \emptyset$
2) $\left\|\mathbf{r}_i - \mathbf{r}_j\right\| < l_{i,j}$
3) $\left(\mathbf{r}_i - \mathbf{r}_j\right) \in V_{i,j}$

$\diamond$

C. Definition of Kinematic Probability of Collision (KPC)

As noted previously, collision events are topologically defined without any notion of the positions of colliding agents being random variables at the time of collision. Thus, if the relative positions of agents are known deterministically, the question of whether or not agents are colliding (in the sense that the conditions in Proposition III.10 are met, which may or may not imply not a physical collision) can be answered as either true or false, but not both.

However, the primary aim of this work is to examine collision events when the relative position between agents is not deterministically known. In such cases, whether or not an object is colliding at any given time with another object is a question that can only be strictly answered in a probabilistic sense.

Additionally, as can be seen in Definition III.7, the way that a collision event is defined implies that it is an instantaneous event, since it is a function of the instantaneous relative position between agents. This motivates the definition of a "kinematic probability of collision" to reflect this physical interpretation of the event of interest.

Definition III.11 (Kinematic probability of collision)**.** The notation $C_{i,j}(t)$ is shorthand for the "event that agents i and j are colliding at time t". Then, the kinematic probability of collision of agents i and j (abbreviated as $\mathrm{KPC}_{i,j}$) is defined as the probability of the event $C_{i,j}(t)$, conditioned on the relative state (and uncertainty) at t_0, as given by

$$\mathrm{KPC}_{i,j}(t|t_0) \doteq p\left(C_{i,j}(t)\middle|t_0\right) \qquad \diamond \tag{III.17}$$

Remark III.12. Assume the hard-body radius simplification (see Def. III.6), and the conditioning of the relative position state (i.e. $\mathbf{R}_{i,j} = \mathbf{R}_{i,j}(t|t_0)$, as seen in Notation III.1). Then, $\mathrm{KPC}_{i,j}$ can be expressed as

$$\mathrm{KPC}_{i,j}(t|t_0) = p\left(\left\|\mathbf{R}_{i,j}(t|t_0)\right\| < l_{i,j}\middle|t_0\right) = p\left(\mathbf{R}_{i,j}(t|t_0) \in V_{i,j}\middle|t_0\right) \tag{III.18}$$

Furthermore, suppose the relative position probability density function $\mathrm{pdf}_{\mathbf{R}_{i,j}|t,t_0}$ is known and is continuous (over the relative position states and over time). Then, $\mathrm{KPC}_{i,j}$ can be computed as

$$\mathrm{KPC}_{i,j}(t|t_0) = \int_{\mathbf{R}\in V_{i,j}} \mathrm{pdf}_{\mathbf{R}_{i,j}|t,t_0}(\mathbf{R}|t,t_0)\, d\mathbf{R} \qquad \diamond \tag{III.19}$$

Remark III.12 is illustrated with an example where the (one-dimensional) relative position is normally distributed, as shown in Figure 2.

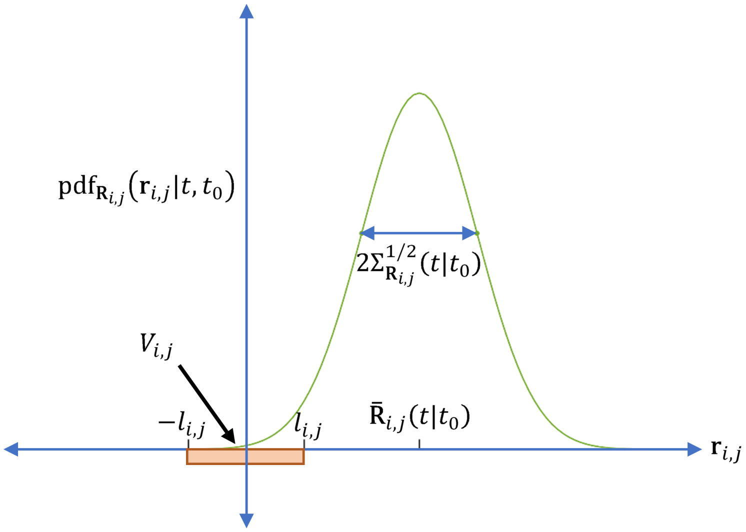


Fig. 2 Computing kinematic probability of collision (KPC) for a system with normally distributed (one-dimensional) relative position (computing directly from relative position pdf).

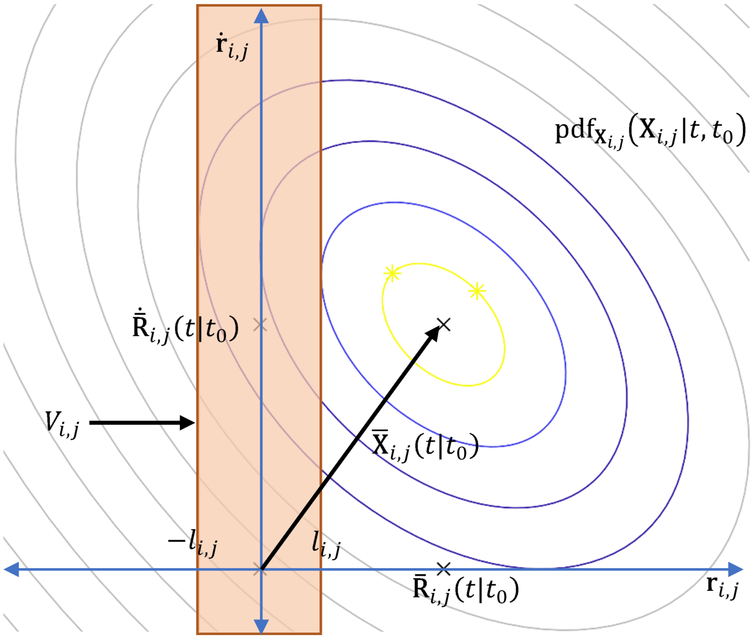
Even though the collision event $C_{i,j}(t)$ is defined as a condition in the relative position states $\mathbf{R}_{i,j}$ at time t, these relative position states (and their uncertainty descriptions) are, in general, dynamically coupled with other states.

According to the notation of subsubsection III.A.2, and assuming that the absolute state of agent i is partitioned as $\mathbf{X}_i^T = \left[\mathbf{R}_i^T, \mathbf{U}_i^T\right]$ (with a corresponding partition for agent j), the relative state $\mathbf{X}_{i,j}$ is partitioned as $\mathbf{X}_{i,j}^T = \left[\mathbf{R}_{i,j}^T, \mathbf{U}_{i,j}^T\right]$.

Suppose that the probability distribution of the entire relative state $\mathrm{pdf}_{\mathbf{X}_{i,j}|t,t_0}$ is known. Then, the probability distribution of the relative position states $\mathrm{pdf}_{\mathbf{R}_{i,j}|t,t_0}$ can be found as the marginal pdf of the entire relative state $\mathbf{X}_{i,j}$ integrated over all the possible "other relative states" $\mathbf{U}_{i,j}$ as given by

$$\mathrm{pdf}_{\mathbf{R}_{i,j}|t,t_0}(\mathbf{R}|t,t_0) = \int_{\mathbf{U}\in\mathbb{R}^{n_\mathbf{X}-d_\mathbf{R}}} \mathrm{pdf}_{\mathbf{X}_{i,j}|t,t_0}\left(\begin{bmatrix}\mathbf{R}\\\mathbf{U}\end{bmatrix}\middle|t,t_0\right) d\mathbf{U} \tag{III.20}$$

Thus, if the pdf of the complete relative state $\mathbf{X}_{i,j}$ is known, $\mathrm{KPC}_{i,j}(t|t_0)$ can be computed accordingly. This is illustrated with an example where the relative position is one-dimensional and the (two-dimensional) relative state is normally distributed, as shown in Figure 3, where $\mathbf{U}_{i,j}$ is the relative velocity $\dot{\mathbf{R}}_{i,j}$.



$\dot{\mathrm{r}}_{i,j}$
$\mathrm{pdf}_{\mathrm{X}_{i,j}}(\mathrm{X}_{i,j}|t,t_0)$
$\dot{\bar{\mathrm{R}}}_{i,j}(t|t_0)$
$V_{i,j}$
$\bar{\mathrm{X}}_{i,j}(t|t_0)$
$-l_{i,j}$
$l_{i,j}$
$\bar{\mathrm{R}}_{i,j}(t|t_0)$
$\mathrm{r}_{i,j}$

Fig. 3 Computing kinematic probability of collision (KPC) for a system with one-dimensional relative position and normally distributed (two-dimensional) relative state (computing from full-state pdf).

It is helpful to note that these other relative states $\mathbf{U}_{i,j}$ do not have to be (or include) velocity, but they must complete the relative dynamic state (i.e. provide enough information for the relative state history to be propagated forward, given inputs). For example, in 2D and 3D Clohessy-Wiltshire (CW) relative orbital dynamics, $\mathbf{U}_{i,j}$ would be the relative position rate state, which is different from a relative velocity state because the CW frame (also known as Hill frame) is not inertial.[42, 43]

D. Dynamic system assumptions

Suppose that the initial mean relative state $\bar{\mathbf{X}}_{i,j}(t_0)$ is known. Suppose that the initial relative state covariance $\Sigma_{\mathbf{X}_{i,j}}(t_0) > 0$ (i.e. $\Sigma_{\mathbf{X}_{i,j}}(t_0)$ is a symmetric, positive definite matrix) and is known.

Finally, suppose that the probability distribution of the initial relative state $\mathbf{X}_{i,j}(t_0)$ is normal, i.e. it is given by

$$\mathbf{X}_{i,j}(t_0) \sim \mathcal{N}\left(\bar{\mathbf{X}}_{i,j}(t_0), \Sigma_{\mathbf{X}_{i,j}}(t_0)\right) \tag{III.21}$$

Let $\mathbf{x}_{i,j}(t)$ be any instance of the relative state $\mathbf{X}_{i,j}(t)$. In some examples, the dynamics of the relative state $\mathbf{X}_{i,j}(t)$ are assumed as linear and uncontrolled, i.e. that there exists a matrix $A(t) \in \mathbb{R}^{n_\mathbf{X} \times n_\mathbf{X}}$ such that

$$\dot{\mathbf{x}}_{i,j}(t) = A(t)\,\mathbf{x}_{i,j}(t) \tag{III.22}$$

for every instance $\mathbf{x}_{i,j}(t)$ of $\mathbf{X}_{i,j}(t)$, and the statement "$\dot{\mathbf{X}}_{i,j}(t) = A(t)\,\mathbf{X}_{i,j}(t)$" carries this interpretation. Therefore, in such examples, the instance $\mathbf{x}_{i,j}(t)$ can be computed as

$$\mathbf{x}_{i,j}(t) = \Phi_{\mathbf{X}_{i,j}}(t,t_0)\,\mathbf{x}_{i,j}(t_0) \tag{III.23}$$

where $\Phi_{\mathbf{X}_{i,j}}(t,t_0)$ is the state transition matrix for the instance $\mathbf{x}_{i,j}$ of the relative state $\mathbf{X}_{i,j}$ from t_0 to t, which satisfies

$$\frac{d}{dt}\Phi_{\mathbf{X}_{i,j}}(t,t_0) = A(t)\,\Phi_{\mathbf{X}_{i,j}}(t,t_0) \tag{III.24}$$

$$\Phi_{\mathbf{X}_{i,j}}(t_2,t_1) = \mathbb{I}_{n_\mathbf{X}} \text{ whenever } t_2 = t_1, t_1 \in \mathbb{R} \tag{III.25}$$

and the statement "$\mathbf{X}_{i,j}(t) = \Phi_{\mathbf{X}_{i,j}}(t,t_0)\,\mathbf{X}_{i,j}(t_0)$" carries this interpretation. Further, $\Phi_{i,j}(t,t_0)$ is used as an abbreviated notation for the relative state transition matrix $\Phi_{\mathbf{X}_{i,j}}(t,t_0)$.

IV. Theory

In this section, integrals of multi-dimensional, normally distributed random variables bounded by certain hypersurfaces are linked to the chi-square distribution. Then, the relationship between these two distributions is leveraged in order to develop the Mahalanobis Shell Sampling (MSS) algorithm for weighted sampling from normal distributions. This section ends with an application of the MSS algorithm to develop a sample-based method for computing the kinematic probability of collision (KPC) between two agents in a relative dynamic system: first, individual sample points are propagated using the same dynamics as the original process; then, particles are flagged whenever they enter the collision region; finally, probabilities of collision can be computed based on the weights of sample points in the collision region.

A. Relating the normal distribution to the chi-squared distribution

Definition IV.1 (Mahalanobis distance)**.** Let $X \sim \mathcal{N}\left(\bar{X}, \Sigma\right)$, where $\bar{X} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n \times n}, \Sigma > 0$. Let $x \in \mathbb{R}^n$ be an instance of X. Then, mapping $D_X^2 : \mathbb{R}^n \to [0, \infty)$ is defined as a nonlinear transformation of instances x of the random variable X, given by the rule

$$D_X^2\left(x\right) \doteq \left[x - \bar{X}\right]^T \Sigma^{-1} \left[x - \bar{X}\right] \tag{IV.1}$$

Thus, $D_X\left(x\right) \doteq \sqrt{D_X^2\left(x\right)}$ is the Mahalanobis distance of x.[44–46]

Note: it is meaningful to define the Mahalanobis distance for general distributions, and this is often done to evaluate the presence of outlier points for empirical, sample distributions of arbitrary phenomena.[45–48] However, use of the Mahalanobis distance in this work is restricted to continuous, normal distributions. ◊

Now, the notions of the d-Mahalanobis contour and volume are introduced.

Notation IV.2 (Mahalanobis contour and volume)**.** Let $X \sim \mathcal{N}\left(\bar{X}, \Sigma\right)$, where $\bar{X} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n \times n}, \Sigma > 0$. Let $x \in \mathbb{R}^n$ be an instance of X. Let the Mahalanobis distance of x, $D_X\left(x\right)$, be as defined in Def. IV.1. Then, the d-Mahalanobis volume of X, denoted by $V_d\left(X\right)$, is defined as the set

$$V_d\left(X\right) \doteq \left\{x \in \mathbb{R}^n : D_X^2\left(x\right) \le d^2\right\} \tag{IV.2}$$

Similarly, the d-Mahalanobis contour of X, denoted by $L_d\left(X\right)$, is defined as the set

$$L_d\left(X\right) \doteq \left\{x \in \mathbb{R}^n : D_X^2\left(x\right) = d^2\right\} \tag{IV.3}$$

Note: the d-Mahalanobis contour is the boundary of the d-Mahalanobis volume, i.e. $L_d\left(X\right) = \partial V_d\left(X\right)$. ◊

Figure 4 illustrates the notions of Mahalanobis contour and volume for an arbitrary normally distributed random variable X, which are true for any finite-dimensionality. The d-Mahalanobis volumes $V_d\left(X\right)$ are hypervolumes (specifically, hyperellipsoids) in n-dimensions. Similarly, the d-Mahalanobis contours $L_d\left(X\right)$ are hypersurfaces (specifically, hyperellipses) in n-dimensions.

Theorem IV.3. Let $X \sim \mathcal{N}\left(\mathbf{0}_{n\times 1}, \mathbb{I}_n\right)$, and let $\mathbf{A} \in \mathbb{R}^{n \times n} : \mathbf{A}^T = \mathbf{A}$. Then,

$$X^T \mathbf{A} X \sim \chi_r^2 \iff \mathbf{A} \text{ is idempotent, and } 1 \le \text{rank}\left(\mathbf{A}\right) = r \le n \tag{IV.4}$$

Note:

- An idempotent matrix $\mathbf{A}$ satisfies $\mathbf{A}^2 = \mathbf{A}$.
- The symbol "χ_r^2" denotes a chi-square distribution with r degrees of freedom, and the notation "$X^T \mathbf{A} X \sim \chi_r^2$" implies that

$$p\left(X^T \mathbf{A} X \le d^2\right) = \text{cdf}_{\chi_r^2}\left(d^2\right) \qquad \Diamond \tag{IV.5}$$

Proof. Shown by Mathai and Provost (1992) [49]. □

Lemma IV.4 gives an analytical expression for integrals of normal probability distributions that are bounded by contours of constant Mahalanobis distance. This result has been shown by Bhattacharya et. al. (See Thm 12.3.2).[46] An alternate, intuitive proof of Lemma IV.4 is presented here.

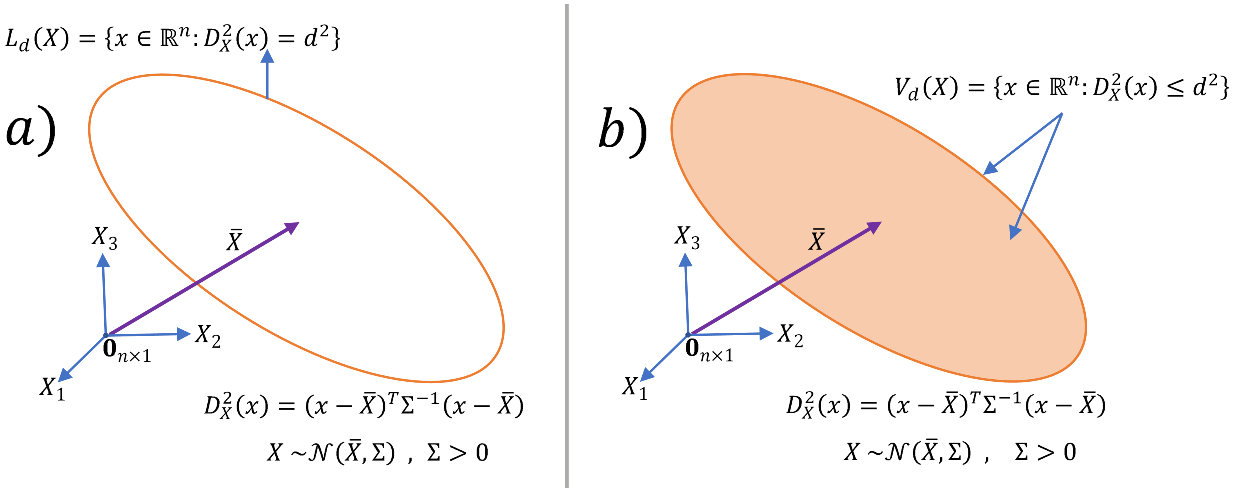

$L_d(X) = \{x \in \mathbb{R}^n: D_X^2(x) = d^2\}$
$a)$
X_3
$\bar{X}$
X_2
$\mathbf{0}_{n\times1}$
X_1
$D_X^2(x) = (x-\bar{X})^T\Sigma^{-1}(x-\bar{X})$
$X \sim \mathcal{N}(\bar{X},\Sigma)$, $\Sigma > 0$
$b)$
$V_d(X) = \{x \in \mathbb{R}^n: D_X^2(x) \le d^2\}$
X_3
$\bar{X}$
X_2
$\mathbf{0}_{n\times1}$
X_1
$D_X^2(x) = (x-\bar{X})^T\Sigma^{-1}(x-\bar{X})$
$X \sim \mathcal{N}(\bar{X},\Sigma)$, $\Sigma > 0$

Fig. 4 d**-Mahalanobis contour and volume, a)** L_d **and b)** V_d**, respectively.**

Lemma IV.4 (Cumulative distributions bounded by contours of constant Mahalanobis distance)**.** Let $X \sim \mathcal{N}\left(\bar{X},\Sigma\right)$, where $\bar{X} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n\times n}, \Sigma > 0$. Then,

$$p\left(\left[X-\bar{X}\right]^T \Sigma^{-1}\left[X-\bar{X}\right] \le d^2\right) = \mathrm{cdf}_{\chi_n^2}\left(d^2\right) \tag{IV.6}$$

Let $x \in \mathbb{R}^n$ be an instance of X. Let $D_X(x)$ be the Mahalanobis distance of x (see Def. IV.1). Let the function $F_X: \mathbb{R} \to [0,1]$ be defined by the rule

$$F_X(d) \doteq \begin{cases} p\left(D_X^2(X) \le d^2\right) & \text{if } d \ge 0 \\ 0 & \text{if } d < 0 \end{cases} \tag{IV.7}$$

Then, F_X is a cumulative distribution function for X. ◇

Proof. Define $Z \doteq \Sigma^{-1/2}\left(X-\bar{X}\right)$. Via Kroese et. al., it follows that $Z \sim \mathcal{N}\left(\mathbf{0}_{n\times1}, \mathbb{I}_n\right)$.[50]

Define $B^2 \doteq Z^T\mathbb{I}_nZ$. Then, via Theorem IV.3, $B^2 \sim \chi_n^2$, i.e.

$$p\left(B^2 \le d^2\right) = \mathrm{cdf}_{\chi_n^2}\left(d^2\right) \tag{IV.8}$$

However, $B^2 = (X-\bar{x})^T\left(\Sigma^{-1/2}\right)^T\Sigma^{-1/2}(X-\bar{x})$. Via Bernstein (see Ch 8.5), since $\Sigma > 0$, it follows that $\Sigma^{-1/2} > 0$.[51] This fact implies that $\Sigma^{-1/2}$ is symmetric. Therefore,

$$\left(\Sigma^{-1/2}\right)^T\Sigma^{-1/2} = \Sigma^{-1/2}\Sigma^{-1/2} = \Sigma^{-1} \tag{IV.9}$$

Thus,

$$B^2 = \left(X-\bar{X}\right)^T\Sigma^{-1}\left(X-\bar{X}\right) = D_X^2(X) \tag{IV.10}$$

It follows that

$$p\left(D_X^2(X) \le d^2\right) = p\left(\left[X-\bar{X}\right]^T\Sigma^{-1}\left[X-\bar{X}\right] \le d^2\right) = \mathrm{cdf}_{\chi_n^2}\left(d^2\right) \tag{IV.11}$$

The function $F_X(d)$, as previously defined, satisfies the following axioms:

1) $0 \le F_X(d)$

   Subproof. In the case that $d < 0$, it follows that $F_X(d) = 0$, by definition.
   In the case that $d \ge 0$, it follows that $F_X(d) = \mathrm{cdf}_{\chi_n^2}\left(d^2\right) \ge 0$. ■

2) $d_1 \le d_2 \implies F_X(d_1) \le F_X(d_2)$

Subproof. Suppose $d_1 \leq d_2$.
In the case that $d_1 < 0$ and $d_2 < 0$, it follows that $F_X(d_1) = F_X(d_2) = 0$, so $F_X(d_1) \leq F_X(d_2)$.
In the case that $d_1 < 0$ and $d_2 \geq 0$, it follows that $F_X(d_1) = 0$ and $F_X(d_2) = \mathrm{cdf}_{\chi_n^2}\left(d_2^2\right) \geq 0$, so $F_X(d_1) \leq F_X(d_2)$.
In the case that $d_1 \geq 0$ and $d_2 \geq 0$,

$$\begin{aligned} F_X(d_2) = p\left(D_X^2(X) \leq d_2^2\right) \\ = p\left(\left[D_X^2(X) \leq d_1^2\right] \cup \left[d_1^2 \leq D_X^2(X) \leq d_2^2\right]\right) = p\left(D_X^2(X) \leq d_1^2\right) + p\left(d_1^2 \leq D_X^2(X) \leq d_2^2\right) \\ \geq p\left(D_X^2(X) \leq d_1^2\right) = F_X(d_1) \end{aligned} \qquad \text{(IV.12)}$$

■

3) $F_X(d) \leq 1$

Subproof. In the case that $d < 0$, it follows that $F_X(d) = 0 \leq 1$. In the case that $d \geq 0$, it follows that $F_X(d) \doteq p\left(D_X^2(X) \leq d^2\right)$ is a probability measure, which means that it is axiomatically bounded above by 1. Therefore, $F_X\left(d^2\right) \leq 1$. ■

4) $\lim_{d \to \infty} F_X(d) = 1$

Subproof. The function $F_X(d)$ is monotonically increasing and is bounded above by 1. Therefore, its limit exists, and is given by

$$\lim_{d \to \infty} F_X(d) = \sup_{d \in \mathbb{R}} F_X(d) = \sup_{d \in [0,\infty)} p\left(D_X^2(X) \leq d^2\right) = 1 \qquad \text{(IV.13)}$$

■

Given that the function F_X satisfies all the axioms for a cumulative distribution function, it follows that F_X is a cumulative distribution function for X.[52] □

Remark IV.5. Since it is trivial that $F_X(d) = 0$ for every $d < 0$, without loss of generality, only cases when $d \geq 0$ are considered in this work. ◊

Figure 5 illustrates the “alternative” cumulative distribuition of a normally distributed random variable X, F_X, for representative dimensionalities of X.

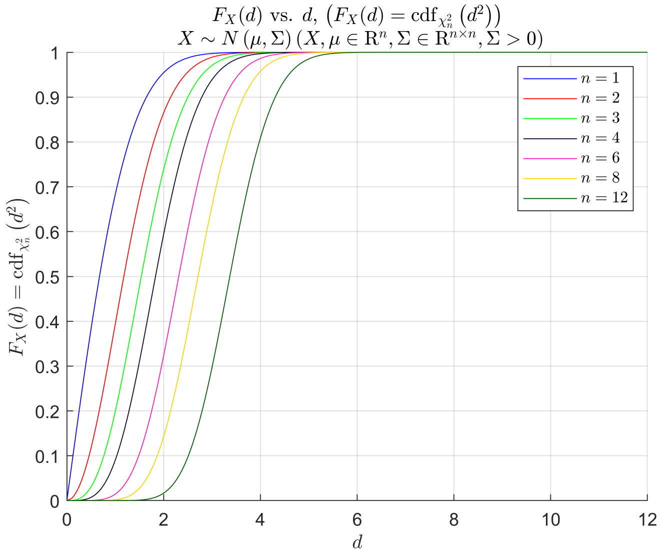


Fig. 5 Chi-square cdf, and connection to normal probability distributions (for representative dimensionalities).

Corollary IV.6 (Probability mass within Mahalanobis volume)**.** Let $X \sim \mathcal{N}\left(\bar{X}, \Sigma\right)$, where $\bar{X} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n \times n}, \Sigma > 0$. Let the d-Mahalanobis volume of X, $V_d\left(X\right)$, be as defined in Not. IV.2. Then,

$$p\left(X \in V_d\left(X\right)\right) = \mathrm{cdf}_{\chi_n^2}\left(d^2\right) \qquad \diamond \tag{IV.14}$$

Now, the notion of the d_1, d_2-Mahalanobis shell is introduced.

Notation IV.7 (Mahalanobis shell)**.** Let $X \sim \mathcal{N}\left(\bar{X}, \Sigma\right)$, where $\bar{X} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n \times n}$, $\Sigma > 0$. Let $x \in \mathbb{R}^n$ be an instance of X. Let the Mahalanobis distance of x, $D_X\left(x\right)$, be as defined in Def. IV.1. Let the d-Mahalanobis volume of X, $V_d\left(X\right)$, and the d-Mahalanobis contour of X, $L_d\left(X\right)$, be as defined in Not. IV.2. Then, the d_1, d_2-Mahalanobis shell of X, denoted by $V_{d_1}^{d_2}\left(X\right)$, is defined as the set

$$V_{d_1}^{d_2}\left(X\right) \doteq \left\{x \in \mathbb{R}^n : 0 \leq d_1 \leq D_X\left(x\right) \leq d_2\right\} \tag{IV.15}$$

Note: the union of the d_1- and d_2-Mahalanobis contours is the boundary of the d_1,d_2-Mahalanobis shell, i.e. $\partial V_{d_1}^{d_2} = L_{d_1}\left(X\right) \cup L_{d_2}\left(X\right)$. $\diamond$

Corollary IV.8 (Probability mass within Mahalanobis shell)**.** Let $X \sim \mathcal{N}\left(\bar{X}, \Sigma\right)$, where $\bar{X} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n \times n}, \Sigma > 0$. Let the d_1, d_2-Mahalanobis shell of X, $V_{d_1}^{d_2}\left(X\right)$, be as defined in Not. IV.7. Then,

$$p\left(X \in V_{d_1}^{d_2}\left(X\right)\right) = \mathrm{cdf}_{\chi_n^2}\left(d_2^2\right) - \mathrm{cdf}_{\chi_n^2}\left(d_1^2\right) \qquad \diamond \tag{IV.16}$$

The implications of Lemma IV.4 and Corollaries IV.6 and IV.8 on normally distributed random variables (with positive definite covariances) are illustrated in Fig. 6. Specifically, for such random variables, probability measures over n-hypervolumes that are bounded by n-hypersurfaces of constant Mahalanobis distances (say, d_1 and d_2) can be found analytically as functions that depend only on d_1 and d_2 (through chi-square cdfs), regardless of the properties of the mean and covariance of the random variable, and regardless of its dimensionality.

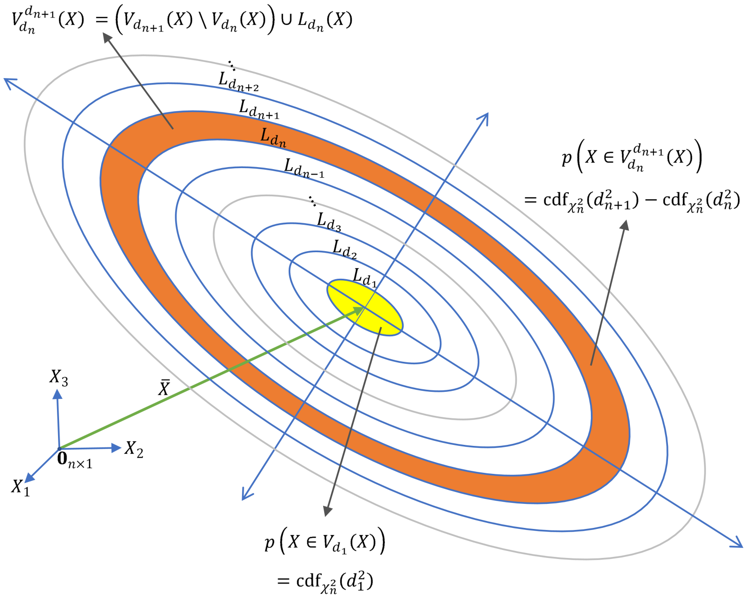

$V_{d_n}^{d_{n+1}}(X) = \left(V_{d_{n+1}}(X) \setminus V_{d_n}(X)\right) \cup L_{d_n}(X)$
⋰
$L_{d_{n+2}}$
$L_{d_{n+1}}$
L_{d_n}
$L_{d_{n-1}}$
⋰
L_{d_3}
L_{d_2}
L_{d_1}
$p\left(X \in V_{d_n}^{d_{n+1}}(X)\right)$
$= \mathrm{cdf}_{\chi_n^2}(d_{n+1}^2) - \mathrm{cdf}_{\chi_n^2}(d_n^2)$
X_3
$\bar{X}$
X_2
$\mathbf{0}_{n \times 1}$
X_1
$p\left(X \in V_{d_1}(X)\right)$
$= \mathrm{cdf}_{\chi_n^2}(d_1^2)$

Fig. 6 Probability measures of normally distributed random variables over n-hypervolumes bounded by n-hypersurfaces of constant d-Mahalanobis distance.)

B. Introducing the Mahalanobis Shell Sampling (MSS) method and applications to KPC computation

The sampling method presented in this work, Algorithm 1 is referred to as the "Mahalanobis Shell Sampling" algorithm (MSS, for short) throughout this work. The MSS algorithm aims to do the following. First, a canonical sample is made along high-dimensional unit sphere surfaces (namely, on the unit $(n-1)$-sphere, where n is the dimension of the points in the normal distribution to be reproduced). This unit hypersphere sample is uniformly distributed in a geometric sense. Subsection VII.A in the Appendix is dedicated to exploring uniform sequences on the unit hypersphere in detail.

Second, the result from Corollary IV.8 (namely, an analytical solution for the probability mass inside a Mahalanobis shell) is leveraged in order to both create MSS sample points and to provide weights for such points. Specifically, the original distribution is truncated until a maximum Mahalanobis distance $d_{\max}$, and it is divided into a certain number of Mahalanobis shells (N_{shells}). Then, for every Mahalanobis shell in the sample, points on the unit hypersphere are transformed to points in their respective Mahalanobis shell. (This transformation employs the statistics of the original normal random variable, as well as some Mahalanobis distance within said Mahalanobis shell). Finally, each point in the MSS sample is given a weight proportional to the probability mass in its respective shell and inversely proportional to the number of points in the shell. In other words, the collective weight of points in each sample Mahalanobis shell is the probability mass in the shell, and the collective weight of the shell is divided evenly among points in the shell.

Thus, by transforming canonical unit hypersphere sample points into sample points in Mahalanobis shells, the MSS algorithm generates a sample that is representative of some normally distributed random variable.

```
Input: $N_{\text{shells}}$, $N_{\text{samples/shell}}$, $d_{\max}$ (max. sample Mahalanobis distance), dimension of elements of the sequence
       $n_{\mathbf{X}}$ ($n_{\mathbf{X}} \in \mathbb{N}, n_{\mathbf{X}} \geq 2$)
Output: MSS sample $\{\tilde{\mathbf{x}}_k(t_0)\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$, sample weights $\{w_k\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$
Data: Initial conditions $\bar{\mathbf{X}}_{i,j}(t_0), \Sigma_{\mathbf{X}_{i,j}}(t_0)$; prior state distribution $\mathbf{X}_{i,j}(t_0) \sim \mathcal{N}\left(\bar{\mathbf{X}}_{i,j}(t_0), \Sigma_{\mathbf{X}_{i,j}}(t_0)\right)$
$\{\mathbf{z}_p\}_{p\in\{1,\cdots,N_{\text{samples/shell}}\}}$ ← output of some unit $(n_{\mathbf{X}}-1)$-sphere sampling algorithm
    $(N_{\text{sequence}} = N_{\text{samples/shell}}, D = n_{\mathbf{X}})$        // see Appendix, subsection VII.A for a list of algorithms
$\delta_d \leftarrow d_{\max}/N_{\text{shells}}$
$\mathbf{U}, \mathbf{S}, \mathbf{V} \leftarrow$ such that $\Sigma_{\mathbf{X}_{i,j}}(t_0) = \mathbf{U}\mathbf{S}\mathbf{V}^T$        // output of SVD
$\Sigma^{1/2}_{\mathbf{X}_{i,j}}(t_0) \leftarrow \mathbf{U}\mathbf{S}^{1/2}\mathbf{V}^T$
$k \leftarrow 0$
$N_{\text{samples}} \leftarrow N_{\text{shells}} N_{\text{samples/shell}}$
for $l \leftarrow 1$ to $N_{\text{shells}}$ do
    $d_l \leftarrow (l - \frac{1}{2})\delta_d$
    $W_l \leftarrow \text{cdf}_{\chi^2_D}\left((d_l + \frac{1}{2}\delta_d)^2\right) - \text{cdf}_{\chi^2_D}\left((d_l - \frac{1}{2}\delta_d)^2\right)$
        // probability mass in in $l^{th}$ ($n_{\mathbf{X}}$-dimensional) Mahalanobis shell
    for $p \leftarrow 1$ to $N_{\text{samples/shell}}$ do
        $k \leftarrow k + 1$
        $\tilde{\mathbf{x}}_k(t_0) \leftarrow \bar{\mathbf{X}}_{i,j}(t_0) + d_l \left[\Sigma^{1/2}_{\mathbf{X}_{i,j}}(t_0)\right] \mathbf{z}_p$        // point in $l^{th}$ Mahalanobis shell
        $w_k \leftarrow W_l/N_{\text{samples/shell}}$        // same weight for pts in the same Mahalanobis shell
return $\{\tilde{\mathbf{x}}_k(t_0)\}_{k\in\{1,\cdots,N_{\text{samples}}\}}, \{w_k\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$
```

Algorithm 1: Generation of MSS sample, general $n_{\mathbf{X}}$-dimensional state, normal prior state distribution.

Corollary IV.8 has interesting implications for sampling from multidimensional, non-degenerate normal distributions. Algorithm 1 is a direct application of Corollary IV.8, whereby using an uniform sample of the unit sphere, a maximum Mahalanobis distance $d_{\max} < \infty$, and a discretization of the Mahalanobis distances $d \in [0, d_{\max}]$, a sample is generated for some normal distribution.

The MSS KPC and TPc/WPC computation method, listed as Algorithm 2, aims to do the following. First, a MSS sample is generated that replicates the initial distribution of a normally distributed relative dynamic state. Then, individual particles in such sample are propagated using arbitrary dynamics (without requiring the propagated distribution to retain normality). Finally, the weights of each particle in the sample are used to compute the probabilities of collision. Thus, through application of the present sampling method to collision probability computation, integration is sidestepped by using weights that reflect integration of regions bounded by certain surfaces, and the difficulties of Monte Carlo methods are sidestepped by using canonical samples that are uniformly distributed in a geometric sense.

In principle, MSS samples can be used for arbitrary purposes since, within some truncation and discretization, they represent the original distribution of some normal random variable. In this work, however, Algorithm 2 is an application

```
Input: Initial time $t_0$, final time $t_f$, $N_t$
Output: Discrete time sample $\mathcal{T}$; kinematic PC $\mathrm{KPC}_{i,j}(t|t_0)$, sample kinematic PC $\widetilde{\mathrm{KPC}}_{i,j}(t|t_0)$, sample
        window PC $\widetilde{\mathrm{WPC}}_{i,j}(t_0,t)$, $t \in \mathcal{T}$; KPC error RMS
Data: Initial conditions $\bar{\mathbf{X}}_{i,j}(t_0), \Sigma_{\mathbf{X}_{i,j}}(t_0)$; prior state distribution $\mathbf{X}_{i,j}(t_0) \sim \mathcal{N}\left(\bar{\mathbf{X}}_{i,j}(t_0), \Sigma_{\mathbf{X}_{i,j}}(t_0)\right)$; MSS
      sample $\{\tilde{\mathbf{x}}_k(t_0)\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$, sample weights $\{w_k\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$ ; matrix $\mathbf{M}$ for mapping from
      complete relative state to relative position (see subsection III.A)
$\delta_t \leftarrow \frac{t_f - t_0}{N_t - 1}$
$a \leftarrow \mathbf{0}_{1\times N_{\text{samples}}}$
$B_2 \leftarrow 0$
for $m \leftarrow 1$ to $N_t$ do
    $t_m \leftarrow t_0 + \frac{t_f - t_0}{N_t - 1}(m-1)$
    $\bar{\mathbf{X}}_{i,j}(t_m) \leftarrow \Phi_{i,j}(t_m, t_0)\,\bar{\mathbf{X}}_{i,j}(t_0)$
    $\Sigma_{\mathbf{X}_{i,j}}(t_m) \leftarrow \Phi_{i,j}(t_m, t_0)\,\Sigma_{\mathbf{X}_{i,j}}(t_0)\Phi^T_{i,j}(t_m, t_0)$
    $\mathbf{R}_{i,j}(t_m|t_0) \sim \mathcal{N}\left(\mathbf{M}\bar{\mathbf{X}}_{i,j}(t_m), \mathbf{M}\Sigma_{\mathbf{X}_{i,j}}(t_m)\mathbf{M}^T\right)$
    $\mathrm{KPC}_{i,j}(t_m|t_0) \leftarrow \int_{\mathbf{R}\in V_{i,j}} \mathrm{pdf}_{\mathbf{R}_{i,j}|t,t_0}(\mathbf{R}|t_m, t_0)\,d\mathbf{R}$    // KPC truth
    $B_1 \leftarrow 0$
    for $k \leftarrow 1$ to $N_{\text{samples}}$ do
        if $m > 1$ then
            $\tilde{\mathbf{x}}_k \leftarrow \Phi_{i,j}(t_m, t_m - \delta_t)\,\tilde{\mathbf{x}}_k$
        $\tilde{\mathbf{r}}_k \leftarrow \mathbf{M}\tilde{\mathbf{x}}_k$
        if $\|\tilde{\mathbf{r}}_k\| \leq l_{i,j}$ then    // $i$-$j$ collision event at time $t_m$ ($k^{th}$ particle)
            $B_1 \leftarrow B_1 + w_k$
            if $[a]_k = 0$ then    // $i$-$j$ collision ($k^{th}$ particle) had not yet occurred
                $[a]_k \leftarrow 1$    // mark $i$-$j$ collision as having occurred ($k^{th}$ particle)
                $B_2 \leftarrow B_2 + w_k$
    $\widetilde{\mathrm{KPC}}_{i,j}(t_m|t_0) \leftarrow B_1$
    $\widetilde{\mathrm{WPC}}_{i,j}(t_0, t_m) \leftarrow B_2$
$\mathcal{T} \leftarrow \{t_m\}_{m\in\{1,\cdots,N_t\}}$
KPC error RMS $\leftarrow \sqrt{\frac{1}{N_t}\sum_{m=1}^{N_t}\left(\widetilde{\mathrm{KPC}}_{i,j}(t_m|t_0) - \mathrm{KPC}_{i,j}(t_m|t_0)\right)^2}$
return $\mathcal{T}$; $\mathrm{KPC}_{i,j}(t|t_0)$, $\widetilde{\mathrm{KPC}}_{i,j}(t|t_0)$, $\widetilde{\mathrm{WPC}}_{i,j}(t_0,t)$, $t \in \mathcal{T}$; KPC error RMS
```

Algorithm 2: Computation of KPC truth, KPC and WPC MSS estimates, assuming normal prior state distribution and linear relative dynamics.

of the MSS sampling method to the problem of the computation of the probability of collision between any two agents in proximity, if the distribution of their initial relative state is normal and known, and if the relative dynamics are linear.

Figure 7 illustrates graphically an example of an application of the MSS method to compute KPC for a relative dynamic system that has normal distribution and whose dimension is $\mathbb{R}^2$, specifically by having one-dimensional relative position and one-dimensional relative velocity. Unlike direct KPC computation from the relative position pdf (illustrated in Figure 2) or computation of KPC through marginalization of the full state pdf into the relative position pdf (illustrated in Figure 3), when MSS is applied, the original distribution is truncated until a maximum Mahalanobis distance $d_{\max}$, divided into Mahalanobis Shells, and then each shell is further subdivided, and one point from within each shell is added to the sample and given a weight proportional to the probability mass in the shell and inversely proportional to the number of points in the shell.

The notation of the MSS sampling method and of the MSS KPC computation method assume that the normally distributed multi-dimensional random variable to be sampled is a relative dynamic state (specifically, the initial conditions of such state). The reason for this notation choice is that the primary motivation for the development of this algorithm is to help in the efficient computation of KPC and TPc/WPC for relative dynamic systems. However, it is helpful to note at this point that, in principle, the MSS sampling algorithm could be used to directly sample from an

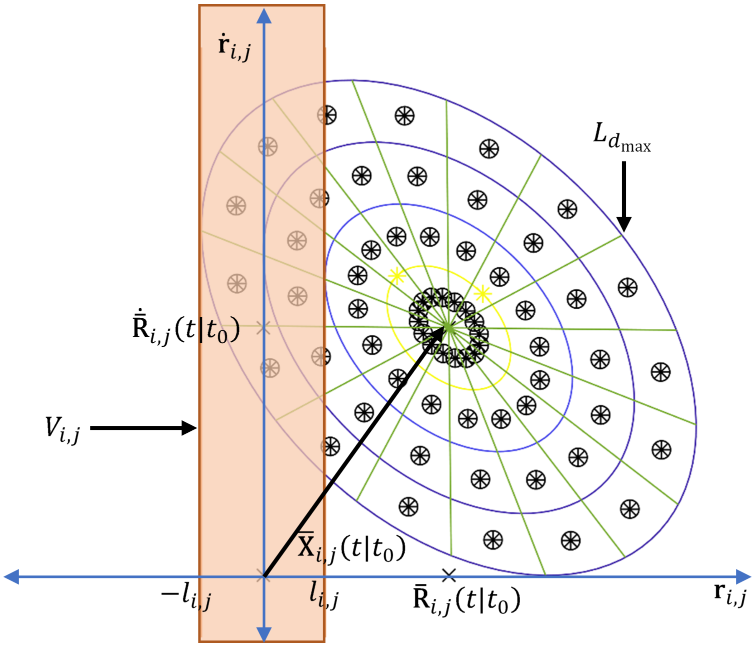


Fig. 7 Computing kinematic probability of collision (KPC) for a system with one-dimensional relative position and normally distributed (two-dimensional) relative state (computing through application of MSS).

arbitrary multidimensional, normally distributed random variable, whether or not it has any physical interpretation. Similarly, in principle, the MSS KPC computation method could be adapted to include nonlinear dynamics, and its end goal could be to obtain other time-integrals of the distribution, or even to just propagate the distribution itself. Thus, the MSS sampling method is a potential choice for other applications, such as filtering, as an alternative to the Unscented Transform or other particle-based sampling methods.

V. Results and discussion

In this section, the Mahalanobis Shell Sampling (MSS) algorithm (see Algorithm 1) is applied in order to compute kinematic probability of collision (KPC) in the context of a dynamic example (see Algorithm 2). The current example involves a set of two mass-spring-damper systems in $\mathbb{R}^2$ (specifically, by having one-dimensional position and one-dimensional velocity). This pedagogical example is useful because an analytical solution can be found for KPC, which is presented in subsection V.A. Then, a slight variant of the MSS applied to KPC computation is presented in subsection V.B for relative dynamic systems in $\mathbb{R}^2$. Finally, the efficacy of the MSS application to KPC computation is examined by comparing the discrete time error RMS between the analytical and estimated KPC waveforms for two sets of initial conditions and system parameters, as discussed in subsection V.C.

A. Relative mass-spring-damper system (rel. position in $\mathbb{R}^1$, rel. state in $\mathbb{R}^2$) - setting up dynamics

In this example, two "boxes" (labeled i and j, respectively) are modeled as individual mass-spring-damper systems. It is assumed that these boxes experience no external forces, and the contact dynamics between these boxes are ignored.

The dynamic state of the center of mass of box i, $\mathbf{X}_i$, (see subsection III.D) is defined in terms of its position R_i (see subsection III.A) and its corresponding velocity $\dot{R}_i$ as $\mathbf{X}_i^T \doteq \left[R_i \,,\, \dot{R}_i\right]$. Denoting the i^{th} mass, damping coefficient and spring constant by m_i, b_i and k_i, respectively, its dynamics can be expressed as follows:

$$\frac{d\mathbf{X}_i}{dt} = \frac{d}{dt}\begin{bmatrix} R_i \\ \dot{R}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_i/m_i & -b_i/m_i \end{bmatrix}\begin{bmatrix} R_i \\ \dot{R}_i \end{bmatrix} \tag{V.1}$$

The dynamics of the position of the center of mass of box j can be expressed similarly. Furthermore, it is assumed that the center of mass of box j is stationary at the origin initially, while the initial state of box i has some nontrivial value.

Therefore, the state of box j is trivial for all time (i.e. $\mathbf{X}_j(t) = \mathbf{0}_{2\times 1}$), and the dynamics of of i relative to j (i.e. the dynamics of $\mathbf{X}_{i,j} \doteq \mathbf{X}_i - \mathbf{X}_j$) are the same as the dynamics of i, namely,

$$\frac{d\mathbf{X}_{i,j}}{dt} = \frac{d}{dt}\begin{bmatrix} R_{i,j} \\ \dot{R}_{i,j} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_i/m_i & -b_i/m_i \end{bmatrix} \begin{bmatrix} R_{i,j} \\ \dot{R}_{i,j} \end{bmatrix} \tag{V.2}$$

The system parameters m_i, b_i and k_i (all positive-valued) are chosen in order to observe an underdamped (i.e. decaying oscillatory) response; thus, $k_i > b_i^2/4m_i$. Letting $\bar{\mathbf{X}}_{i,j}(t_0)$ denote the initial mean relative state, where $\bar{\mathbf{X}}_{i,j}(t_0) = \left[\bar{R}_{i,j}(t_0)\ ,\ \dot{\bar{R}}_{i,j}(t_0)\right]^T$, the mean relative state $\bar{\mathbf{X}}_{i,j}(t)$ can be computed as

$$\bar{\mathbf{X}}_{i,j}(t) = \Phi_{i,j}(t, t_0)\,\bar{\mathbf{X}}_{i,j}(t_0) \tag{V.3}$$

where the relative state transition matrix $\Phi_{i,j}(t, t_0)$ can be computed as

$$\Phi_{i,j}(t, t_0) = \Phi_{i,j}(t - t_0, 0) \tag{V.4}$$

where

$$\Phi_{i,j}(t, 0) = \begin{bmatrix} [\Phi_{i,j}(t,0)]_{1,1} & [\Phi_{i,j}(t,0)]_{1,2} \\ [\Phi_{i,j}(t,0)]_{2,1} & [\Phi_{i,j}(t,0)]_{2,2} \end{bmatrix} \tag{V.5}$$

$$[\Phi_{i,j}(t,0)]_{1,1} = \exp(-\zeta\omega_n t)\left[\cos(\omega_d t) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d t)\right] \tag{V.6}$$

$$[\Phi_{i,j}(t,0)]_{1,2} = \exp(-\zeta\omega_n t)\frac{1}{\omega_n\sqrt{1-\zeta^2}}\sin(\omega_d t) \tag{V.7}$$

$$[\Phi_{i,j}(t,0)]_{2,1} = -\exp(-\zeta\omega_n t)\frac{\omega_n}{\sqrt{1-\zeta^2}}\sin(\omega_d t) \tag{V.8}$$

$$[\Phi_{i,j}(t,0)]_{2,2} = \exp(-\zeta\omega_n t)\left[\cos(\omega_d t) - \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d t)\right] \tag{V.9}$$

and

$$\omega_n = \sqrt{k_i/m_i} \tag{V.10}$$

$$\zeta = b_i/2\sqrt{k_i m_i} \tag{V.11}$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2} = \sqrt{k_i/m_i - (b_i/2m_i)^2} \tag{V.12}$$

Let the relative state covariance $\Sigma_{\mathbf{X}_{i,j}} > 0$ be related to the variables $\sigma_{R_{i,j}}$, $\sigma_{\dot{R}_{i,j}}$ and $\rho_{(R_{i,j},\dot{R}_{i,j})}$ as follows

$$\Sigma_{\mathbf{X}_{i,j}} = \begin{bmatrix} [\Sigma_{\mathbf{X}_{i,j}}]_{1,1} & [\Sigma_{\mathbf{X}_{i,j}}]_{1,2} \\ [\Sigma_{\mathbf{X}_{i,j}}]_{1,2} & [\Sigma_{\mathbf{X}_{i,j}}]_{2,2} \end{bmatrix} = \begin{bmatrix} \sigma_{R_{i,j}}^2 & \rho_{(R_{i,j},\dot{R}_{i,j})}\sigma_{R_{i,j}}\sigma_{\dot{R}_{i,j}} \\ \rho_{(R_{i,j},\dot{R}_{i,j})}\sigma_{R_{i,j}}\sigma_{\dot{R}_{i,j}} & \sigma_{\dot{R}_{i,j}}^2 \end{bmatrix} \tag{V.13}$$

Suppose $\Sigma_{\mathbf{X}_{i,j}}(t_0) > 0$ is known. Since the i-j dynamics are linear, the relative state covariance can be computed as

$$\Sigma_{\mathbf{X}_{i,j}}(t|t_0) = \Phi_{i,j}(t, t_0)\,\Sigma_{\mathbf{X}_{i,j}}(t_0)\,\Phi_{i,j}^T(t, t_0) \tag{V.14}$$

Suppose the initial probability distribution of the relative center of mass is normal (see Eq. III.21). Because the dynamics of the relative state $\mathbf{X}_{i,j}$ are linear and uncontrolled, its distribution remains normal for all time, i.e.

$$\mathbf{X}_{i,j}(t) \sim \mathcal{N}\left(\Phi_{i,j}(t, t_0)\,\bar{\mathbf{X}}_{i,j}(t_0), \Phi_{i,j}(t, t_0)\,\Sigma_{\mathbf{X}_{i,j}}(t_0)\,\Phi_{i,j}^T(t, t_0)\right) \tag{V.15}$$

Based on Remark III.12, the KPC between agents i and j, $\mathrm{KPC}_{i,j}(t|t_0)$, can be computed as

$$\mathrm{KPC}_{i,j}(t|t_0) = \int_{-l_{i,j}}^{l_{i,j}} \mathrm{pdf}_{\mathbf{R}_{i,j}|t,t_0}(\mathbf{R}|t, t_0)\, d\mathbf{R} \tag{V.16}$$

For this example, the kinematic probability of collision $\mathrm{KPC}_{i,j}\left(t|t_0\right)$ waveform can be found analytically as

$$\mathrm{KPC}_{i,j}\left(t|t_0\right) = \frac{1}{2}\left[\mathrm{erf}\left(\frac{l_{i,j} - \bar{R}_{ij}\left(t\right)}{\sigma_{R_{i,j}}\left(t\right)\sqrt{2}}\right) - \mathrm{erf}\left(\frac{-l_{i,j} - \bar{R}_{ij}\left(t\right)}{\sigma_{R_{i,j}}\left(t\right)\sqrt{2}}\right)\right] \tag{V.17}$$

where

$$\bar{R}_{ij}\left(t\right) = \left[\Phi_{i,j}\left(t,t_0\right)\right]_{1,1}\bar{R}_{i,j}\left(t_0\right) + \left[\Phi_{i,j}\left(t,t_0\right)\right]_{1,2}\bar{\dot{R}}_{i,j}\left(t_0\right) \tag{V.18}$$

and

$$\begin{aligned}\sigma^2_{R_{i,j}}\left(t\right) = {} & \left[\Phi_{i,j}\left(t,t_0\right)\right]^2_{1,1}\sigma^2_{R_{i,j}}\left(t_0\right) + \left[\Phi_{i,j}\left(t,t_0\right)\right]^2_{1,2}\sigma^2_{\dot{R}_{i,j}}\left(t_0\right) \\ & + 2\left(\left[\Phi_{i,j}\left(t,t_0\right)\right]_{1,1}\left[\Phi_{i,j}\left(t,t_0\right)\right]_{1,2}\right)\rho_{\left(R_{i,j},\dot{R}_{i,j}\right)}\left(t_0\right)\sigma_{R_{i,j}}\left(t_0\right)\sigma_{\dot{R}_{i,j}}\left(t_0\right)\end{aligned} \tag{V.19}$$

B. Mahalanobis Shell Sampling (MSS) algorithm - variant for system with rel. position in $\mathbb{R}^1$, rel. state in $\mathbb{R}^2$.

A variant of the general $n_{\mathbf{X}}$-dimensional MSS sampling algorithm (see Algorithm 1), specifically for the case of a two-dimensional relative dynamic state with one-dimensional relative position, is presented as Algorithm 3 and implemented for the examples of this section.

```
Input: $N_{\text{shells}}$, $N_{\text{samples/shell}}$, $d_{\max}$ (max. sample Mahalanobis distance)
Output: MSS sample $\{\tilde{\mathbf{x}}_k(t_0)\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$, sample weights $\{w_k\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$
Data: Initial conditions $\bar{\mathbf{X}}_{i,j}(t_0), \Sigma_{\mathbf{X}_{i,j}}(t_0)$; prior state distribution $\mathbf{X}_{i,j}(t_0) \sim \mathcal{N}\left(\bar{\mathbf{X}}_{i,j}(t_0), \Sigma_{\mathbf{X}_{i,j}}(t_0)\right)$
$\delta_d \leftarrow d_{\max}/N_{\text{shells}}$
$\mathbf{U}, \mathbf{S}, \mathbf{V} \leftarrow$ such that $\Sigma_{\mathbf{X}_{i,j}}(t_0) = \mathbf{U}\mathbf{S}\mathbf{V}^T$          // output of SVD
$\Sigma^{1/2}_{\mathbf{X}_{i,j}}(t_0) \leftarrow \mathbf{U}\mathbf{S}^{1/2}\mathbf{V}^T$
$k \leftarrow 0$
$N_{\text{samples}} \leftarrow N_{\text{shells}}N_{\text{samples/shell}}$
for $l \leftarrow 1$ to $N_{\text{shells}}$ do
    $d_l \leftarrow (l - \frac{1}{2})\delta_d$
    $W_l \leftarrow \mathrm{cdf}_{\chi^2_2}\left((d_l + \frac{1}{2}\delta_d)^2\right) - \mathrm{cdf}_{\chi^2_2}\left((d_l - \frac{1}{2}\delta_d)^2\right)$
                                  // probability mass in in $l^{th}$ (2-dimensional) Mahalanobis shell
    $r \sim U[0,1]$
    $\theta_0 \leftarrow 2\pi r/N_{\text{samples/shell}}$
    for $p \leftarrow 1$ to $N_{\text{samples/shell}}$ do
        $k \leftarrow k + 1$
        $\theta_{l,p} = \theta_0 + 2\pi(p-1)/N_{\text{samples/shell}}$
        $\mathbf{z}_k \leftarrow \begin{bmatrix}\cos(\theta_{l,p}) & \sin(\theta_{l,p})\end{bmatrix}^T$          // point in 1-sphere (i.e.  circle)
        $\tilde{\mathbf{x}}_k(t_0) = \bar{\mathbf{X}}_{i,j}(t_0) + d_l\left[\Sigma^{1/2}_{\mathbf{X}_{i,j}}(t_0)\right]\mathbf{z}_k$          // point in $l^{th}$ Mahalanobis shell
        $w_k \leftarrow W_l/N_{\text{samples/shell}}$          // same weight for pts in the same Mahalanobis shell
return $\{\tilde{\mathbf{x}}_k(t_0)\}_{k\in\{1,\cdots,N_{\text{samples}}\}}, \{w_k\}_{k\in\{1,\cdots,N_{\text{samples}}\}}$
```

Algorithm 3: Generation of MSS sample, 2-dimensional rel. state (one-dimensional rel. position), normal prior state distribution

C. Relative mass-spring-damper system (rel. position in $\mathbb{R}^1$, rel. state in $\mathbb{R}^2$) - KPC and WPC - MSS results and discussion

Two sets of initial conditions and system parameters are considered for the example of the current section, and they are listed in Table 1. In both cases, system parameters are chosen so that the relative system response is underdamped (see subsection V.A), and since this implies that both cases have asymptotically stable dynamics, the mean relative state converges to the origin asymptotically, and the relative covariance vanishes. Therefore, these system parameters imply that the kinematic probability of collision (KPC) asymptotically approaches the value 1, which implies a sure collision. Additionally, the joint hard body radius $l_{i,j}$ is chosen to be large in order to observe higher collision probabilities.

Table 1 MSS applications 1D mass-spring-damper system examples

Parameter	Example #1D.001	Example #1D.002
$\bar{R}_{i,j}\,(t_0)$ (m)	1	1
$\bar{\dot{R}}_{i,j}\,(t_0)$ (m/s)	0	4
$\sigma_{R_{i,j}}\,(t_0)$ (m)	1	1
$\sigma_{\dot{R}_{i,j}}\,(t_0)$ (m/s)	1	1
$\rho_{\left(R_{i,j},\dot{R}_{i,j}\right)}\,(t_0)$	0	0
m_i (kg)	4	4
b_i (kg/s)	1	0.25
k_i (kg/s^2)	1	2
$l_{i,j}$ (m)	0.5	0.5
$\left(t_f - t_0\right)$ (s)	20	45

The first case, Example #1D.001, is discussed in subsubsection V.C.1, and it is chosen so that a “faster” convergence to a sure collision scenario (within 1.25 oscillations) can be observed compared to that of Example #1D.002 (within 8.3 oscillations), which is discussed in subsubsection V.C.2. Discussion of results comprises the following: first, qualitative comparison of KPC among analytically-, Monte Carlo- and MSS-generated waveforms; second, motivating TPc/WPC as a collision risk indicator; and third, comparing the performance of changing MSS sample parameters by directly comparing the analytically- and MSS-generated waveforms. The metric used to compare these waveforms is the discrete-time Euclidean distance, i.e. the difference root-mean-square (RMS) between the waveforms (referred to as “error” RMS for brevity), or simply referred to as KPC error RMS.

1. Kinematic probability of collision (KPC) results - Example #1D.001

Qualitatively, for Example #1D.001, the KPC waveforms generated analytically, through Monte Carlo sampling of the initial state distribution, and through MSS (shown in Figures 8, 9 and 10, respectively) are in agreement with one another, i.e. they start at a local minimum (consistent with agent *i* starting at a location furthest from the origin (on its right) before moving towards the origin), a local maximum is crossed at 1/4-oscillation (consistent with agent *i* crossing the origin) before another local minimum at 1/2-oscillation (consistent with agent *i* being locally furthest from the origin on the left), before another local maximum at 3/4-oscillation (consistent with agent *i* crossing the origin) before another local minimum at 1-oscillation (consistent with agent *i* being locally furthest from the origin on the right again). This behavior is intuitive and consistent with two facts: 1) the expectation that the KPC waveform should asymptotically approach the value of 1, implying that a collision is happening “almost surely” after a threshold, which for this case is after 1.25 oscillations; and 2) that the system response is decaying and sinusoidal, as previously described.

It is interesting to note that the “Total Probability of collision” (TPc, also labeled as WPC, see Frigm et. al. [36]) waveforms generated through Monte Carlo sampling of the initial state distribution and through MSS (shown in Figures 9 and 10, respectively) are also consistent, and they show that (almost) every region of the initial relative state pdf has crossed the collision volume $V_{i,j}$ (see Fig. 3 for reference) by the 1/3-oscillation, which implies that a collision has “almost surely” occurred by this time. This observed TPc/WPC behavior is not in contradiction to that of the KPC waveform; in fact, the information obtained from observing the KPC and TPc/WPC waveforms is distinct but complementary: while a collision *is “almost surely” occurring* at the 1.25-oscillation (and any time after that), a collision *has “almost surely” occurred* sometime between the simulation start time and the 1/3-oscillation.

This example shows that, from a collision risk analysis perspective, the question that the TPc/WPC is trying to answer (the probability of collision anytime within a compact time window) has the potential of being an important collision risk indicator. Furthermore, in some cases, the TPc/WPC datum might be more appropriate as a collision risk indicator than the maximum kinematic probability of collision (KPC) at any given time within the same time window. For reference, by the 0.44-oscillation, the TPc/WPC value was exactly 1 for the Monte Carlo case (i.e. 100%

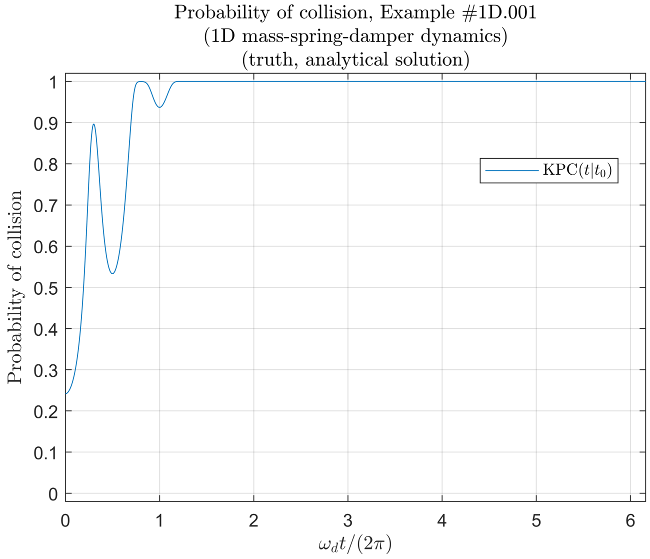


Fig. 8 Probability of collision waveform results, analytically-generated, Example #1D.001

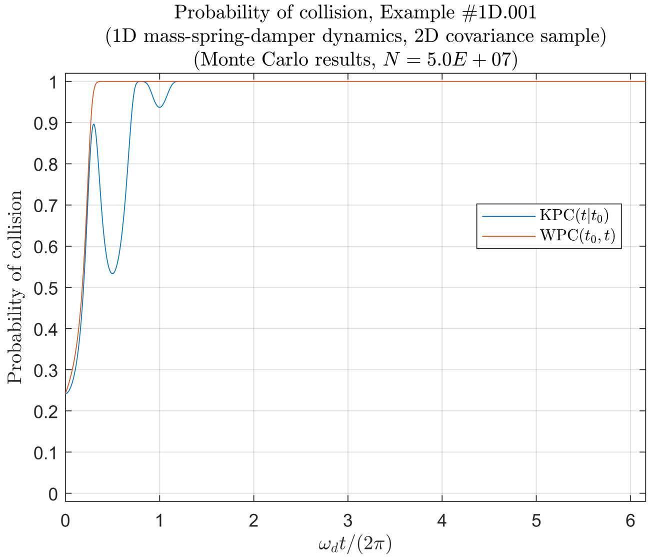


Fig. 9 Probability of collision waveform results, Monte Carlo-generated, Example #1D.001

of all 50 million Monte Carlo particles had collided by then), while the TPc/WPC value for the MSS case was exactly $1 - 1.6075E - 11$, which is very close to the total sample weight $\text{cdf}_{\chi_2^2}\left(7.05^2\right) = 1 - 1.6115E - 11$.

As was discussed in the Introduction (see section II), in order to compute the TPc/WPC, a sampling-based method is required; therefore, it is helpful to compare such methods. It is interesting to note that, through MSS, samples were made of points within $d_{\max} = 7.05$ (colloquially, to within 7.05-"σ") utilizing only 16920 points. The probability mass outside $d_{\max} = 7.05$ is equal to $1 - \text{cdf}_{\chi_2^2}\left(7.05^2\right) \approx 1.6115E - 11$, i.e. the chance of that a point is Monte Carlo-sampled at or beyond 7.05-"σ" is about 1 in 6.205E10, which is more than three orders of magnitude larger than the Monte Carlo sample size (5E7). In fact, the chance of a Monte Carlo sample point to be at or beyond 5.9544-"σ" is is equal to $1 - \text{cdf}_{\chi_2^2}\left(5.9544^2\right) \approx 2E - 8$, or about 1 in 5E7; therefore, only one point in the entire Monte Carlo sample is expected to cross this threshold with the number of points in the sample (5E7). In turn, there are 22 MSS shells (i.e. 2640 points total) with points in that same range (i.e. at or beyond 5.9544-"σ").

Thus, it seems that the MSS sample method can account for probabilistic outliers without unduly penalizing the

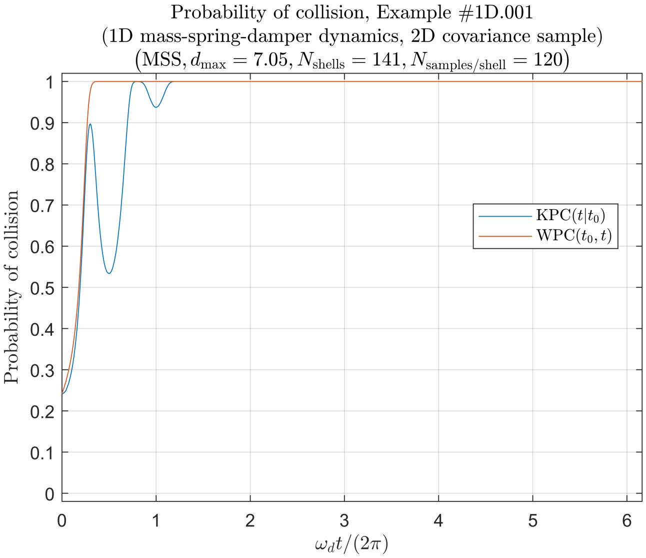


Fig. 10 Probability of collision waveform results, MSS-generated, Example #1D.001

sample size; thus, the MSS sample method might enable achieving comparable accuracy (for low-valued probabilities of collision) when compared to Monte Carlo (at the very least in a qualitative sense) at a fraction of the effort. For Example #1D.001, waveform computation using MSS method took 0.5942 sec, while Monte Carlo took 453.7546 sec (in both methods, execution time in MATLAB was averaged over 10 runs, with 0.02 sec timestep and accounting for sample generation).

As can be seen in Algorithm 1, there are three primary inputs to perform an MSS sample: the number of shells N_{shells}, the number of samples per shell $N_{\text{samples/shell}}$, and the cutoff Mahalanobis distance $d_{\max}$. Figures 11, 12 and 13 are concerned with comparing how accurately the MSS KPC waveform approximates the analytical KPC waveform, specifically, when changing the number of shells N_{shells} and the cutoff Mahalanobis distance $d_{\max}$ (for a fixed, arbitrary number of samples per shell $N_{\text{samples/shell}}$).

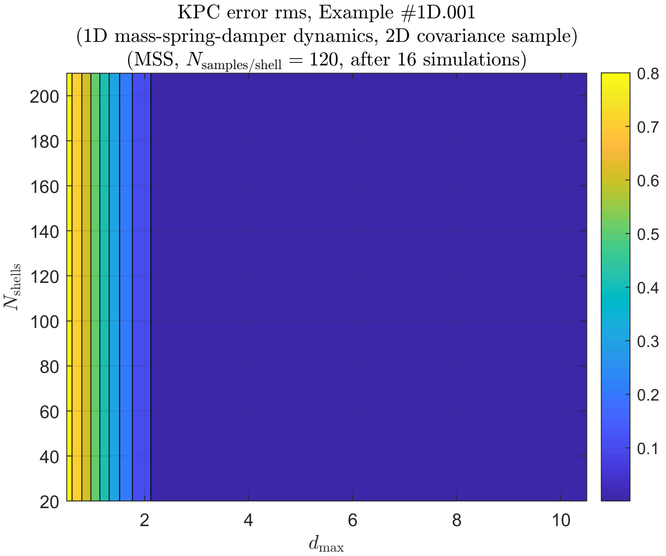


Fig. 11 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing number of shells N_{shells} and cutoff Mahalanobis distance $d_{\max}$ (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.001.

Upon examining the trade of changing the number of shells N_{shells} versus changing cutoff Mahalanobis distance $d_{\max}$

(as seen in Figure 11), it is seen that while the MSS KPC method is insensitive to the number of shells N_{shells}, increasing the cutoff Mahalanobis distance d_{max} results in a monotonic increase in error RMS accuracy. However, this observation only holds for “low” cutoff Mahalanobis distances d_{max}. This observation can be explained by noting that the MSS sample algorithm does not implement sample weight normalization, i.e. the sum of the weights of the elements of the sample is not 1, but $1 - \text{cdf}_{\chi^2_{(n_{\mathbf{x}})}}\left(d^2_{\text{max}}\right)$, so a high enough d_{max} is needed for a valid sample.

Next, Figure 12 is examined, where representative cutoff Mahalanobis distances $d_{\text{max}} \in \{4, \dots, 10\}$ are held constant, while changing the number of shells N_{shells}. It is clear that, for constant cutoff Mahalanobis distances d_{max}, increasing the number of shells N_{shells} improves RMS residuals. Additionally, it can be seen that, after $d_{\text{max}} = 4$, better RMS residuals are obtained with lower d_{max}. It can be seen that, after $d_{\text{max}} = 4$, equal accuracy can be achieved with higher d_{max} by increasing N_{shells}; conversely, after $d_{\text{max}} = 4$, increasing d_{max} while holding N_{shells} constant decreases RMS accuracy. This trend is opposite to that observed in Figure 11, where for low d_{max} and wht, when holding N_{shells} constant, increasing d_{max} ile holding N_{shells} constant, increasing d_{max} increases RMS accuracy. Combined, these trends suggest thaimproves RMS accuracy up to a point, after which continuing to increase d_{max} worsens RMS accuracy.

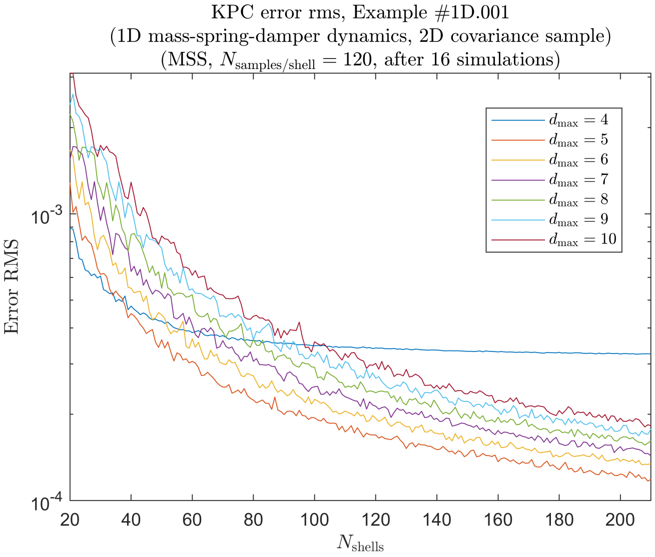


Fig. 12 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing number of shells N_{shells} while holding cutoff Mahalanobis distance d_{max} constant (with fixed $N_{\text{samples/shell}}$); Example #1D.001.

In Figure 13, representative $N_{\text{shells}} \in [20, 210]$ are held constant, while changing cutoff Mahalanobis distances d_{max}. It is seen that, in every case, RMS residuals improve up to a certain “transition” cutoff Mahalanobis distance d'_{max} (which depends on N_{shells}), and the RMS residuals deteriorate with $d_{\text{max}} > d'_{\text{max}}$. Furthermore the, “transition” cutoff Mahalanobis distance d'_{max} increases with increased number of shells N_{shells}. For $N_{\text{shells}} \in [20, 210]$, the “transition” cutoff Mahalanobis distance d'_{max} is between 4 and 5, which is consistent with observations from Figure 12.

These observations imply that there exists an underlying requirement for Mahalanobis shell resolution; in other words, in order to maintain or improve error RMS performance while increasing d_{max}, it is necessary to increase the number of shells N_{shells}. Thus, MSS KPC error RMS accuracy is not insensitive to the number of shells N_{shells}. In fact, for constant KPC error RMS, increasing d_{max} requires an increase in N_{shells}; conversely, for constant KPC error RMS, increasing N_{shells} allows sampling from increasingly greater cutoff Mahalanobis distances d_{max}.

It should be noted that increasing d_{max} by itself does not affect sample size. Therefore, increasing d_{max} while keeping N_{shells} constant effectively creates a grid that, while including more probabilistic outliers, becomes increasingly more coarse. However, while correctly increasing N_{shells} in tandem, increasing d_{max} should always improve RMS accuracy (at the cost of additional computation and longer time to converge).

2. *Kinematic probability of collision (KPC) results - Example #1D.002*

For Example #1D.002, most insights obtained are similar to those obtained through examination of Example #1D.001 (see Subsubsection V.C.1). Qualitatively, for Example #1D.002, the KPC waveforms generated analytically,

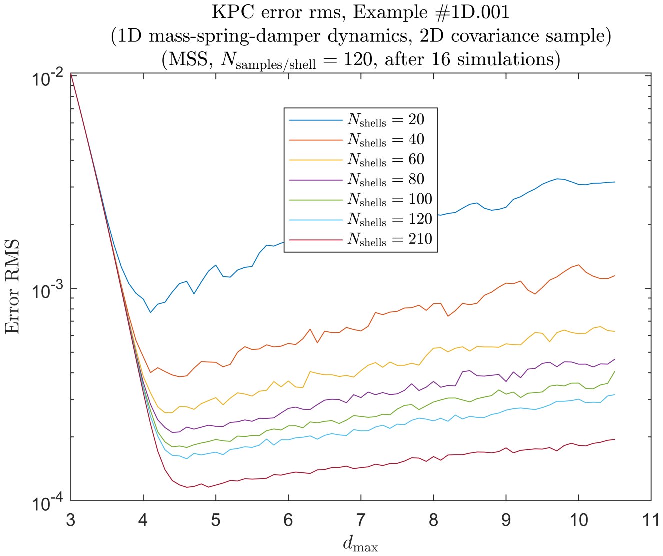


Fig. 13 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing cutoff Mahalanobis distance $d_{\max}$ while holding number of shells N_{shells} constant (with fixed $N_{\text{samples/shell}}$); Example #1D.001.

through Monte Carlo sampling of the initial state distribution, and through MSS (shown in Figures 14, 15, and 16, respectively, are in agreement with one another. Namely, during every oscillation, there is one KPC local minimum followed (1/4-oscillation after) by one KPC local maximum, before repeating the sequence after 1/4-oscillation. Unlike Example #1D.001, the KPC waveforms do not start at a local minimum because agent *i* is initially moving further away from the origin.

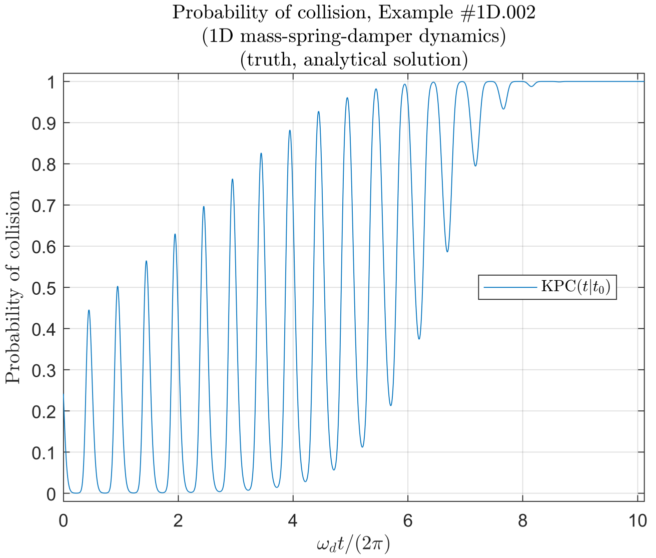


Fig. 14 Probability of collision waveform results, analytically-generated, Example #1D.002

Example #1D.002 is an interesting example that further illustrates how TPc/WPC might be an important collision risk indicator. The TPc/WPC waveforms generated through Monte Carlo sampling of the initial state distribution and through MSS (shown in Figures 15 and 16, respectively) are also consistent, and they show that (almost) every region of the initial relative state pdf has crossed the collision volume $V_{i,j}$ (see Fig. 3 for reference) by the 1/2-oscillation, which implies that a collision has "almost surely" occurred by this time. In contrast, the KPC waveform indicates that the first time a collision is “almost surely” occurring is at the 7.933-oscillation, and a collision is “almost surely” occurring at the 8.789-oscillation and every time after that.

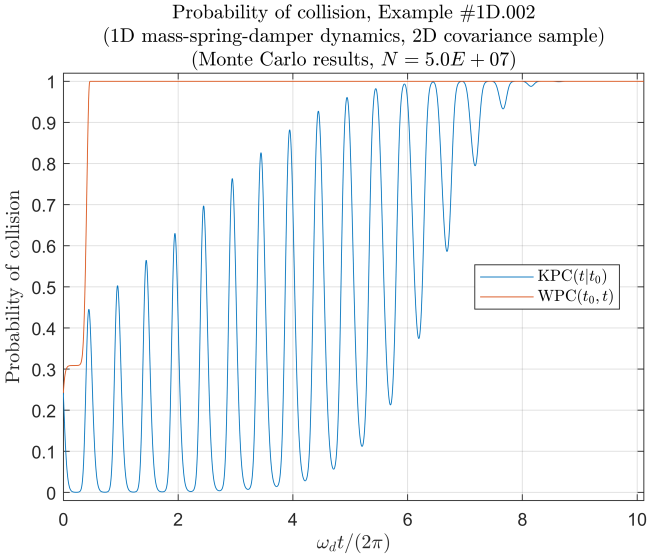


Fig. 15 Probability of collision waveform results, Monte Carlo-generated, Example #1D.002

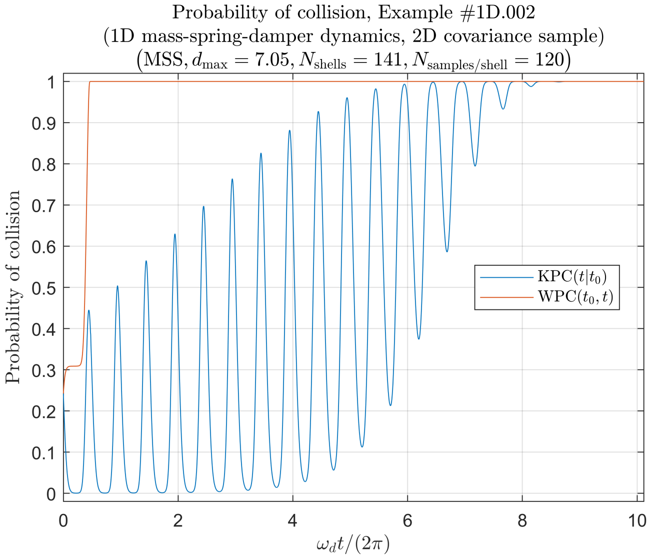


Fig. 16 Probability of collision waveform results, MSS-generated, Example #1D.002

The contrast between the information that can be gleaned from the KPC and TPc/WPC waveforms is especially noteworthy between the simulation start time and the end of the 2$^{\mathrm{nd}}$ oscillation. On one hand, in that interval, four times the KPC waveform indicates that a collision either is not occurring or has low probability of occurring; on the other hand, at those same four times, the TPc/WPC shows that a collision has occurred by said times with probability of over 30% (the first time), and almost 100% (the other three times). Regardless of the dynamics of convergence of the KPC waveform to an “almost sure” collision, in an operational context, such concerns would be irrelevant when considering that a collision might be imminently expected much sooner.

For reference, by the 0.48-oscillation, the TPc/WPC value was exactly 1 for the Monte Carlo case (i.e. 100% of all 50 million Monte Carlo particles had collided by then), while the TPc/WPC value for the MSS case was exactly

$1 - 1.6105E - 11$, which is very close to the total sample weight $\mathrm{cdf}_{\chi_2^2}\left(7.05^2\right) = 1 - 1.6115E - 11$. Additionally, it is worth noting that, for Example #1D.002, waveform computation using MSS method took 1.2808 sec, while Monte Carlo took 1007.1877 sec (in both methods, execution time in MATLAB was averaged over 10 runs, with 0.02 sec timestep and accounting for sample generation). MSS execution time for Example #1D.001 was 2.16 times faster than for Example #1D.002, while the former had a propagation horizon 2.25 times shorter than the latter.

In a similar fashion to Example #1D.001, in Example #1D.002, upon examining the trade of changing the number of shells N_{shells} versus changing cutoff Mahalanobis distance $d_{\max}$, it is observed that, for "low" cutoff Mahalanobis distances $d_{\max}$, the MSS KPC method is insensitive to the number of shells N_{shells}, and that increasing the cutoff Mahalanobis distance $d_{\max}$ results in a monotonic increase in error RMS accuracy. This observed behavior is consistent after 1 simulation (see Figure 17) and after averaging over 23 simulations (see Figure 20).

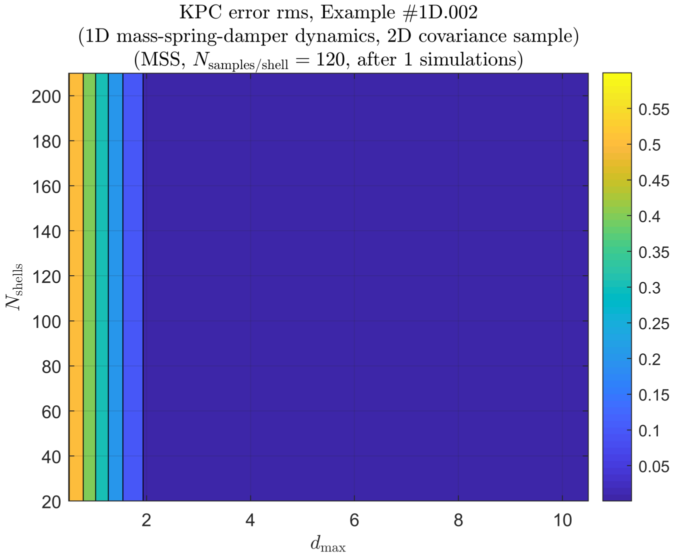


Fig. 17 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing number of shells N_{shells} and cutoff Mahalanobis distance $d_{\max}$ (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.002 (after 1 simulation).

For Example #1D.002, when examining the trade of changing the number of shells N_{shells} in the MSS sample, while holding cutoff Mahalanobis distances $d_{\max} \in \{4, \ldots, 10\}$ are held constant. First, for constant cutoff Mahalanobis distance $d_{\max}$, increasing the number of shells N_{shells} improves RMS residuals. Additionally, it is seen that, after $d_{\max} = 4$, better RMS residuals are obtained with lower $d_{\max}$. The observed trends are consistent after 1 simulation (see Figure 18) and after averaging over 23 simulations (see Figure 21).

Second, for constant number of shells N_{shells}, increasing cutoff Mahalanobis distance $d_{\max}$ starts improving RMS residuals before stagnating at a "transition" cutoff Mahalanobis distance $d'_{\max}$, after which continuing to increase cutoff Mahalanobis distance $d_{\max}$ degrades RMS residuals. The observed trends are consistent after 1 simulation (see Figure 19) and after averaging over 23 simulations (see Figure 22).

To examine the reproducibility of these trends, the same simulation is averaged over 23 simulations. It is noteworthy that, after averaging, the trends appear to smooth out, which show that for the one-dimensional position, 2D state MSS sampling method, rotating each unit shell by a random angle, and thus adding an element of randomness to the specific location of each shell point (within its respective cell), gives the algorithm probabilistic consistency (see Algo 3, line 9), at least within each dynamic system example. When extending the algorithm to higher dimensions, it is worth examining whether it might be beneficial to rotate unit shells through random rotations as well, which would entail modifying the main MSS sampling algorithm (see Algo 1) slightly in order to reflect this change.

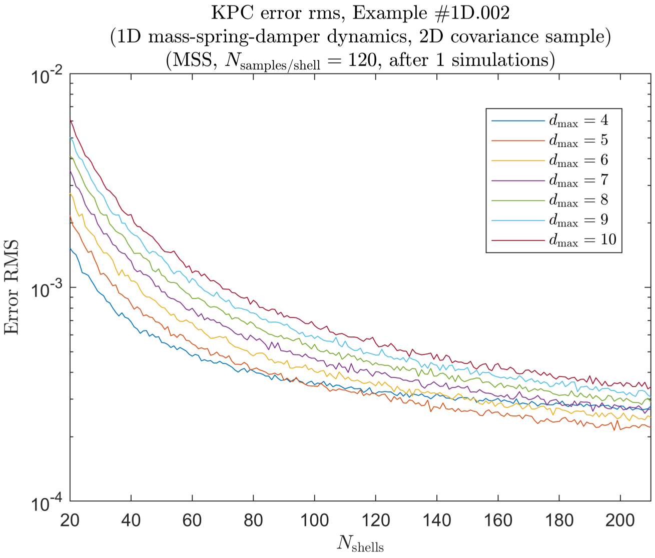


Fig. 18 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing number of shells N_{shells} while holding cutoff Mahalanobis distance d_{max} constant (with fixed $N_{\text{samples/shell}}$); Example #1D.002 (after 1 simulation).

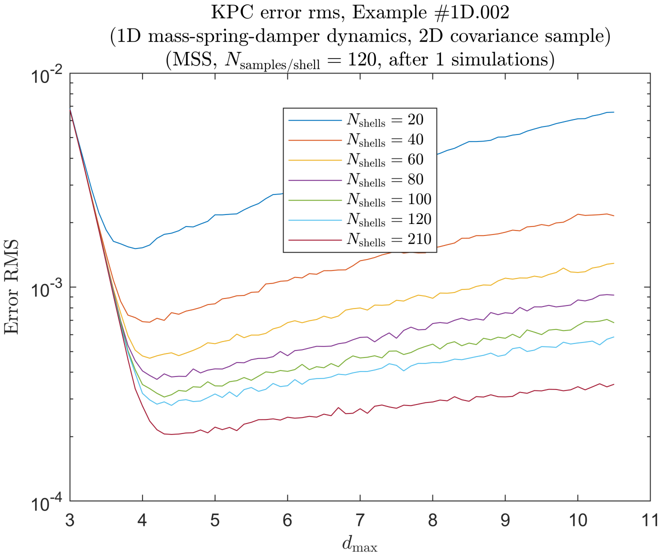


Fig. 19 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing cutoff Mahalanobis distance d_{max} while holding number of shells N_{shells} constant (with fixed $N_{\text{samples/shell}}$); Example #1D.002 (after 1 simulation).

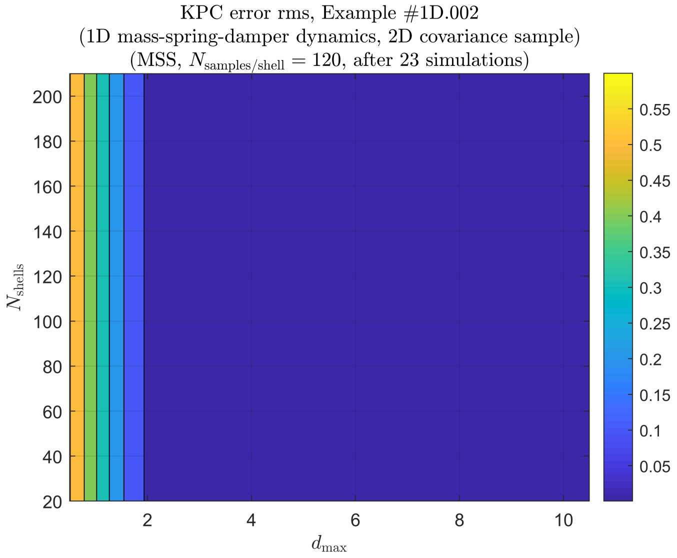


Fig. 20 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing number of shells N_{shells} and cutoff Mahalanobis distance $d_{\max}$ (with fixed $N_{\mathrm{samples/shell}}$) in MSS sample; Example #1D.002 (averaging over 23 simulations).

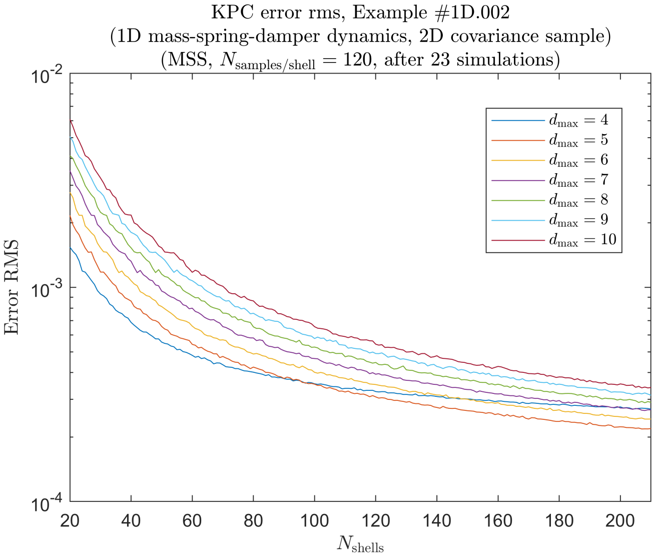


Fig. 21 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing number of shells N_{shells} while holding cutoff Mahalanobis distance $d_{\max}$ constant (with fixed $N_{\mathrm{samples/shell}}$); Example #1D.002 (averaging over 23 simulations).

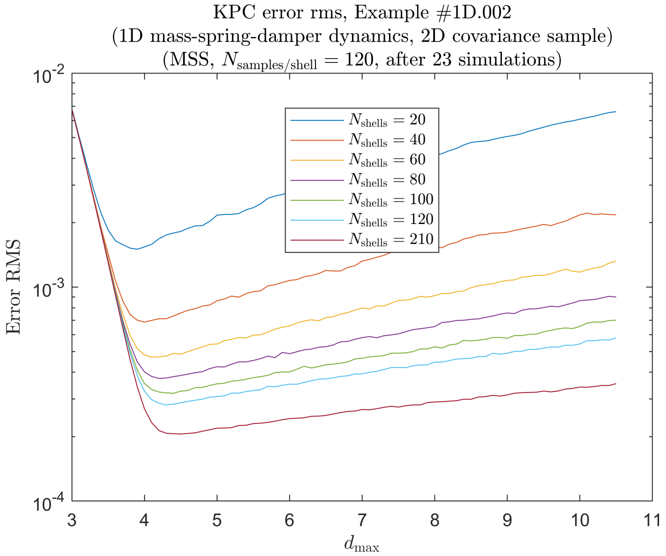


Fig. 22 Euclidean distance (i.e. error RMS) between analytical and MSS KPC waveforms; changing cutoff Mahalanobis distance $d_{\max}$ while holding number of shells N_{shells} constant (with fixed $N_{\mathrm{samples/shell}}$); Example #1D.002 (averaging over 23 simulations).

VI. Conclusion

In this work, the notions of collision events and the kinematic probability of collision (KPC) are reintroduced and formalized. Then, for the specific case of normal probability distribution functions (pdfs), certain pdf integrals (namely, those bounded by contours of constant Mahalanobis distance, or integrals over Mahalanobis shells) are analytically linked to the chi-square distribution. Upon the basis of this relationship, a sampling method, referred to as the Mahalanobis Shell Sampling algorithm (MSS), is developed. Using the parameters of normal, nondegenerate distributions, the MSS algorithm transforms a sample from the unit hypersphere into a sample of the original normal, nondegenerate distribution. The MSS algorithm accomplishes this by giving weights to each sample point that are proportional to the probability mass in the target shell and inversely proportional to the number of points in such shell. The MSS sampling method could in principle be applied to sample from arbitrary (non-degenerate) normal pdfs, regardless of their physical interpretation or application (or lack thereof). However, in accordance with a motivation of this work, a method is presented in which MSS samples are used to compute kinematic probability of collision (KPC) waveforms, as well as total/window probability of collision (TPc/WPC) waveforms.

Given a choice of unit hypersphere sampling algorithm, a MSS sample is parameterized by three primary inputs: the number of shells N_{shells}, the number of samples per shell $N_{\mathrm{samples/shell}}$, and the cutoff Mahalanobis distance $d_{\max}$. The performance of MSS samples for KPC computation is examined through a simple, intuitive example, where the relative position between agents is one-dimensional and the relative state is two-dimensional, and an analytical KPC solution is found for this example. Further, the system parameters and initial conditions are chosen so as to guarantee a sure collision. The performance of the MSS KPC method is measured through the Euclidean distance (or difference RMS) between the analytically and MSS generated waveforms.

The KPC waveforms generated through analytical, Monte Carlo and MSS methods are qualitatively consistent, and they follow the intuition of the dynamics of the example presented. It is found that, for low cutoff Mahalanobis distances $d_{\max}$, the KPC error RMS is insensitive to the number of shells N_{shells} in the sample, and KPC error RMS monotonically decreases with increasing cutoff Mahalanobis distances $d_{\max}$. However, more generally, increasing the number of shells N_{shells} in the sample (while holding cutoff Mahalanobis distances $d_{\max}$ constant) monotonically improves RMS residuals. In contrast, when increasing cutoff Mahalanobis distances $d_{\max}$ in the sample (while holding the number of shells N_{shells} constant), an improvement in RMS residuals is observed up to "transition" cutoff Mahalanobis distance $d'_{\max}$, after which continuing to increased $d_{\max}$ results in RMS residual deterioration. Further, it is found that this "transition" cutoff Mahalanobis distance $d'_{\max}$ increases with increased number of shells N_{shells}.

These observations give insight for application of this method to KPC computation. This work is developed with the motivation of having an alternative method of KPC computation in an operational context where the KPC values might be low (for orders of magnitude between 1E-10 to 1E-3 in the context of spacecraft formation collision risk analysis), which motivates the need from sampling from "tails" of probability distributions. From inspection of KPC error RMS results, continuing to increase cutoff Mahalanobis distances $d_{\max}$ (which means increasing, in a normalized sense, the maximum distance of sample points from the sample mean, and implying having more extreme probabilistic outliers), without also increasing the number of shells N_{shells}, is ultimately detrimental to KPC accuracy. However, increasing the number of shells N_{shells}, without also increasing cutoff Mahalanobis distances $d_{\max}$, always results in KPC accuracy improvements. Further examination of applications of MSS to KPC computation should quantify the required increment in N_{shells} in order to obtain non-deteriorating KPC error RMS properties for incremental cutoff Mahalanobis distances $d_{\max}$. Additionally, in future work, the MSS sample parameters required have desired convergence rate and error properties should be characterized; conversely, for certain families of dynamic systems and sets of initial conditions, appropriate KPC error RMS values should be identified in order to obtain desired KPC and TPc/WPC accuracy. In accordance with the motivation of this work, spacecraft formations should be among the dynamic systems explored in this effort.

It should be noted that, even though this work demonstrates that the MSS sampling method can faithfully reproduce pdfs for a specific, low-dimensionality dynamic example, the suitability of this method to approximate pdfs representative of other dynamic processes should be examined, as well how the sample size required for comparable accuracy scales in higher dimensions. In future work, it would be beneficial to quantify the relative performance improvement of MSS KPC/TPc/WPC computation compared not only to Monte Carlo, but also to other sampling algorithms, and to make these comparisons for different families of dynamic systems and sets of initial conditions. Additionally, in future work, it would be beneficial to quantify the relative performance improvement of MSS KPC computation compared not only to Monte Carlo, but also to other sampling algorithms, and to make these comparisons for different families of dynamic systems and sets of initial conditions, including relative orbital dynamics.

Results were also examined for the "Total Probability of collision" (TPc, also labeled as WPC) waveforms. From the perspective of collision risk analysis, the results indicate that TPc/WPC information has the potential of being an important collision risk indicator; specifically, in the few sets of initial conditions examined in this work, the TPc/WPC indicates that a collision would have *"almost surely" occurred* much earlier than the KPC indicates that a collision would be *"almost surely" occurring*. It appears that, even though TPc/WPC information is distinct from KPC information, the former is complementary to the latter, thus suggesting that TPc/WPC information is operationally useful, especially if an imminent collision (on an intolerable high risk of collision) is expected much earlier (or at all) as evidenced by the TPc/WPC indicator than a later (or possibly not existing) expectation of collision as evidenced by the KPC indicator. Specifically, the TPc/WPC indicator can be used to indicate whether a collision is likely to occur within a future time window. Since a sampling method is required to compute TPc/WPC, as an algorithm that can sample from "tails" of normal probability distributions efficiently (provided enough shells are added to the sample), the MSS KPC method is a candidate for TPc/WPC computation. It is expected that MSS KPC method would perform better than Monte Carlo in most cases, but this also should be quantified, and other algorithms should be considered as well.

In future work, it would be appropriate to formally define the TPc/WPC collision risk indicator, similarly to how it was done in this work. Additionally, it would be appropriate to prove some properties of the TPc/WPC waveform, including its relationship to the KPC waveform. Furthermore, application of the TPc/WPC indicator as an actionable collision risk indicator (i.e. as information that is usable to trigger a process of collision avoidance) should be explored.

The MSS method presupposes a sample on the unit hypersphere (i.e. on the surface of the unit hyperball), and the Appendix is dedicated to exploring this subject in detail. With the exception of the Marsaglia method, most unit hypersphere sampling algorithms use a sample from the unit hypercube and a mapping from the unit hypercube to the unit hypersphere. In this work, the only unit hypercube sampling algorithm considered is presented by Roberts [53]; in future work, it would be beneficial to examine other methods of quasi-random number sequence generation (qrns) for uniform sampling on the hypercube (such as Sobol and Niederreiter).[54, 55] In this work, two mappings from the unit hypercube to the unit hypersphere sampling algorithms are considered: a variant of high-dimensional spherical coordinates, as well as an extension of the Lambert equal area projection to higher dimensions. Additionally, in future work, it would be beneficial to explore other mappings from the unit hypercube to the unit hypersphere as well. Finally, in this work, the minimum arclength between pairs of elements in a sample of the unit hypersphere is used as an indicator for how evenly distributed such sample is on the unit hypersphere, and this is done by observing how slowly that minimum arclength decreases when adding elements to the sequence. It is possible that metrics other than arclength might be advantageous, computationally or otherwise, in order to measure the dispersion of a sequence on the unit sphere.

In summary, this work creates a framework for evaluating collision risk in a precise manner. First, by allowing to sample directly from "tails" of normal probability distributions, the Mahalanobis Shell Sampling (MSS) method allows for efficient sample-based KPC computation, restricted to cases where relative dynamic states have normal prior distribution. Second, being validated through the reproducibility of KPC waveforms, the MSS method allows the computation of a recently proposed collision risk indicator, the "Total Probability of Collision" (TPc). The TPc figure is motivated in this work as an indicator whose information is both consistent with and complementary to KPC information, and examples shown demonstrate that TPc insights can be more relevant than KPC insights. Efficient, accurate and quick computation of collision risk indicators is enabled through this work, which is a necessary step before triggering any collision avoidance process.

VII. Appendix

A. Algorithms for sampling from the unit $(D-1)$-sphere $(D \in \{3,4,6\})$.

This subsection is concerned with algorithms for uniform sampling on the unit sphere in high-dimensions. The Marsaglia (subsubsection VII.A.1) and the Roberts and Brauchart (subsubsection VII.A.3) sampling algorithms are presented.

Additionally, the performance of these algorithms is quantified in terms of the d_0 metric, which represents the minimum great circle distance between any two distinct points in a sample, as given by Def. VII.1.

Definition VII.1 (d_0 performance metric)**.** The function $d_0 : \tilde{S}(N) \longrightarrow [0,\infty)$ is defined by the rule

$$d_0\left(\tilde{S}(N)\right) = \min\left\{\arccos\left(\mathbf{z}_i^T\mathbf{z}_j\right) : i,j \in \{1,\cdots,N\}, i \neq j, \mathbf{z}_i \in \tilde{S}(N)\,\forall i \in \{1,\cdots,N\}\right\} \tag{VII.1}$$

where $\left(\tilde{S}(N)\right)$ is a sample of points in the unit $(D-1)$-sphere with N distinct elements in the sample. ◊

The goal of these sampling algorithms is to maximize $d_0\left(\tilde{S}\left(N_{\text{sequence}}\right)\right)$ for any given sample $\tilde{S}\left(N_{\text{sequence}}\right)$, and to have the slowest possible decrease in d_0 with increased N_{sequence}. This is a traditional method for ensuring that sequences produced by these algorithms are "evenly" distributed on the unit $(D-1)$-sphere.[56]

The performance of these unit sphere sampling algorithms (from a d_0 perspective), for the 2-sphere, 3-sphere and 5-sphere cases, is discussed in Subsubsections VII.A.4, VII.A.5 and VII.A.6, respectively, and the figures presented are listed in Table 2.

Table 2 Performance charts for unit sphere sampling.

Metric	2-sphere	3-sphere	5-sphere
$d_0\left(\tilde{S}\left(N_{\text{sequence}}\right)\right)$	Fig. 23	Fig. 27	Fig. 32
$d_0\left(\tilde{S}\left(N_{\text{sequence}}\right)\right)\sqrt[5]{N_{\text{sequence}}}$			Fig. 33
$d_0\left(\tilde{S}\left(N_{\text{sequence}}\right)\right)\sqrt[4]{N_{\text{sequence}}}$		Fig. 28	Fig. 34
$d_0\left(\tilde{S}\left(N_{\text{sequence}}\right)\right)\sqrt[3]{N_{\text{sequence}}}$	Fig. 24	Fig. 29	Fig. 35
$d_0\left(\tilde{S}\left(N_{\text{sequence}}\right)\right)\sqrt{N_{\text{sequence}}}$	Fig. 25	Fig. 30	Fig. 36
$d_0\left(\tilde{S}\left(N_{\text{sequence}}\right)\right)\cdot N_{\text{sequence}}$	Fig. 26	Fig. 31	Fig. 37

The performance of these algorithms are summarized in Table 3. Overall, the Brauchart sphere sampling algorithm has the best d_0 performance of all algorithms considered. It is helpful to note that, while both the Marsaglia and Brauchart algorithms improve with increased dimensionality, the Roberts algorithm does not, so it is inadvisable to use the Roberts algorithm in higher dimensions, at least not with the present implementation (i.e. with the quasi-random number sequence generation algorithm on the unit hypercube shown in Algo. 5).

1. Marsaglia algorithm for uniform sampling on the unit $(D-1)$-sphere.

The Marsaglia algorithm for generation of sequences of points that are uniformly distributed on the unit $(D-1)$-sphere, presented as Algo. 4, is a simple algorithm in which, for each point in the sequence, each component is drawn from the standard normal distribution, and then the point is normalized so that it lies on the unit sphere. Unlike the other methods

Table 3 Performance results for unit sphere sampling algorithms.

	2-sphere		3-sphere		5-sphere	
Algorithm	L.B.	U.B.	L.B.	U.B.	L.B.	U.B.
Marsaglia	$\mathcal{O}\left(\frac{1}{N}\right)$	$\mathcal{O}\left(\frac{1}{N}\right)$	$\mathcal{O}\left(\frac{1}{N}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt[3]{N}}\right)$
Roberts	$\mathcal{O}\left(\frac{1}{N}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$	$\mathcal{O}\left(\frac{1}{N}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$	$\mathcal{O}\left(\frac{1}{N}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$
Brauchart	$\mathcal{O}\left(\frac{1}{N}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt[3]{N}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt[3]{N}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt[4]{N}}\right)$

presented in subsubsection VII.A.3, which attempt to have an uniform distribution in the unit sphere in a geometric sense, the Marsaglia algorithm distributes points on the surface uniformly in a probabilistic sense. This algorithm was first presented by M. Muller, and it was popularized by G. Marsaglia.[57, 58]

```
Input: Number of elements in sequence $N_{\text{sequence}}$; dimension of elements of the sequence $D$ ($D \in \mathbb{N}, D \geq 2$)
Output: Sequence of points on the unit $(D-1)$-sphere $\{\mathbf{z}_s\}_{s \in \{1,\cdots,N_{\text{sequence}}\}}$
for $s \leftarrow 1$ to $N_{\text{sequence}}$ do
    for $n \leftarrow 1$ to $D$ do
        $[\mathbf{y}_s]_n \sim \mathcal{N}(0,1)$
    end
    $\mathbf{z}_s \leftarrow \mathbf{y}_s / \|\mathbf{y}_s\|$        // point on the unit $(D-1)$-sphere
end
return $\{\mathbf{z}_s\}_{s \in \{1,\cdots,N_{\text{sequence}}\}}$
```

Algorithm 4: Sample of points uniformly spread over the surface of the unit $(D-1)$-sphere, Marsaglia method

2. Roberts algorithm for quasi-uniform sampling from the unit n-hypercube.

The Roberts quasi-random number sequence (qrns) generation algorithm is a low-discrepancy qrns generation algorithm for uniform sampling within the unit n-hypercube, and is listed in Algo. 5. This algorithm is based on a generalization of Fibonacci sequences' "golden ratio" number to higher dimensions, and it was presented by Roberts as an alternative to known low-discrepancy quasi-random number sequence (qrns) generation methods such as Sobol and Niederreiter.[53, 59]

```
Input: Number of elements in sequence $N_{\text{sequence}}$; dimension of elements of the sequence $n$ ($n \in \mathbb{N}, n \geq 2$)
Output: Sequence of quasi-random numbers$\{\mathbf{y}_s^n\}_{s \in \{1,\cdots,N_{\text{sequence}}\}}$
$\phi_n : (\phi_n)^{n+1} = \phi_n + 1$        // generalized golden ratio numbers
$\alpha \leftarrow \mathbf{0}_{n\times 1}$
for $j \leftarrow 1$ to $n$ do
    $[\alpha]_j \leftarrow \left(\frac{1}{\phi_n}\right)^j$
end
for $s \leftarrow 1$ to $N_{\text{sequence}}$ do
    for $j \leftarrow 1$ to $n$ do
        $[\mathbf{y}_s^n]_j \leftarrow \left(s\,[\alpha]_j\right) \bmod 1$
    end
end
return $\{\mathbf{y}_s^n\}_{s \in \{1,\cdots,N_{\text{sequence}}\}}$
```

Algorithm 5: Quasi-random number (low-discrepancy) sequence generation, quasi-uniformly distributed in unit n-hypercube $[0,1]^n$, Roberts method

3. Algorithms for quasi-random, quasi-uniform sampling on the unit $(D-1)$*-sphere*

The following algorithms show how to make samples from the unit $(D-1)$-sphere (see Definition III.4). These algorithms rely on the Roberts quasi-random number sequence on the $(D-1)$ unit hypercube (see Algo. 5), although in principle any other such algorithm could be used for this purpose as well.[53]

The Roberts algorithm for generation of quasi-uniformly distributed sequences of points on the unit $(D-1)$-sphere is presented as Algo 6. This algorithm was presented in 3D for the 2-sphere by Brannon, and it was implemented by Roberts.[53, 60] The algorithm was slightly modified in order to use a latitude angle instead of an inclination angle, and a naive extension of this method to high-dimensional spherical coordinates is presented in this work using a procedure similar to the one shown by Blumenson.[61, 62]

```
Input: $N_{\text{samples/shell}}$; dimension of elements of the sequence $D$ $(D \in \mathbb{N}, D \geq 3)$
Output: Sequence of points on the unit $(D-1)$-sphere $\left\{\mathbf{z}_p\right\}_{p\in\{1,\cdots,N_{\text{samples/shell}}\}}$
$\left\{\mathbf{y}_p^{(D-1)}\right\}_{p\in\{1,\cdots,N_{\text{samples/shell}}\}} \leftarrow$ output of Algo. 5, $\left(N_{\text{sequence}} = N_{\text{samples/shell}}, n = D-1\right)$    // collection of quasi-uniformly distributed points in the $[0,1]^{(D-1)}$ unit hypercube
for $p \leftarrow 1$ to $N_{\text{samples/shell}}$ do
    $\mathbf{x} \leftarrow \mathbf{y}_p^{(D-1)}$
    $\theta \leftarrow 2\pi\,[\mathbf{x}]_1$
    for $j \leftarrow 2$ to $D-1$ do
        $\lambda_{(j-1)} : \sin\left(\lambda_{(j-1)}\right) = 2\,[\mathbf{x}]_j - 1$
    end
    $\mathbf{z} \leftarrow \mathbf{0}_{D\times 1}$
    $[\mathbf{z}]_1 \leftarrow \cos(\theta)\prod_{k=1}^{D-2}\cos(\lambda_k)$
    $[\mathbf{z}]_2 \leftarrow \sin(\theta)\prod_{k=1}^{D-2}\cos(\lambda_k)$
    if $D > 3$ then
        for $j \leftarrow 3$ to $D-1$ do
            $[\mathbf{z}]_j \leftarrow \sin\left(\lambda_{(j-2)}\right)\prod_{k=j-1}^{D-2}\cos(\lambda_k)$
        end
    end
    $[\mathbf{z}]_D \leftarrow \sin\left(\lambda_{(D-2)}\right)$
    $\mathbf{z}_p \leftarrow \mathbf{z}$
end
return $\left\{\mathbf{z}_p\right\}_{p\in\{1,\cdots,N_{\text{samples/shell}}\}}$
```

Algorithm 6: Sample of points quasi-uniformly spread over the surface of the unit $(D-1)$-sphere, extension of Roberts method (based on high-dimensional spherical coordinates)

The Brauchart algorithm for generation of quasi-uniformly distributed sequences of points on the unit $(D-1)$-sphere is presented as Algo 7. This algorithm was developed by Brauchart et. al. as an extension of the area preserving Lambert transform (from points in unit hypercubes to surfaces of hyperspheres) for higher dimensions, and their work includes a proof of the area preserving property of the proposed transform.[54] This method utilizes the regularized incomplete beta (h_m) function, shown in Def. VII.2.[54]

Definition VII.2 (Regularized incomplete beta (h_m) function)**.** The function $h_m : [0,1] \to [0,1]$ is defined by the rule

$$h_m(x) = \frac{B(x; d/2, d/2)}{B(1; m/2, m/2)} \tag{VII.2}$$

where $m \in \mathbb{N}$, and $B(z; a, b)$ is the incomplete beta function, given by

$$B(z; a, b) = \int_0^z u^{a-1}(1-u)^{b-1}\,du \qquad \diamond \tag{VII.3}$$

```
Input: $N_{\text{samples/shell}}$; dimension of elements of the sequence $D$ ($D \in \mathbb{N}, D \geq 3$)
Output: Sequence of points on the unit $(D-1)$-sphere $\left\{\mathbf{z}_p\right\}_{p \in \{1,\cdots,N_{\text{samples/shell}}\}}$
$\left\{\mathbf{y}_p^{(D-1)}\right\}_{p \in \{1,\cdots,N_{\text{samples/shell}}\}} \leftarrow$ output of Algo. 5, $\left(N_{\text{sequence}} = N_{\text{samples/shell}}, n = D-1\right)$    // collection of quasi-uniformly distributed points in the $[0,1]^{(D-1)}$ unit hypercube
for $p \leftarrow 1$ to $N_{\text{samples/shell}}$ do
    $\mathbf{x} \leftarrow \mathbf{y}_p^{(D-1)}$
    $\mathbf{z} \leftarrow \begin{bmatrix} \cos\left(2\pi\,[\mathbf{x}]_1\right) & \sin\left(2\pi\,[\mathbf{x}]_1\right) \end{bmatrix}^T$
    $\mathbf{z} \leftarrow \begin{bmatrix} \left(\sqrt{1-(1-2\,[\mathbf{x}]_2)^2}\right)\mathbf{z}^T & 1-2\,[\mathbf{x}]_2 \end{bmatrix}^T$
    if $D > 3$ then
        for $j \leftarrow 3$ to $D-1$ do
            $\mathbf{z} \leftarrow \begin{bmatrix} \left(\sqrt{1-\left(1-2h_j^{-1}\left([\mathbf{x}]_j\right)\right)^2}\right)\mathbf{z}^T & 1-2h_j^{-1}\left([\mathbf{x}]_j\right) \end{bmatrix}^T$
    $\mathbf{z}_p \leftarrow \mathbf{z}$
return $\left\{\mathbf{z}_p\right\}_{p \in \{1,\cdots,N_{\text{samples/shell}}\}}$
```

Algorithm 7: Sample of points quasi-uniformly spread over the surface of the unit $(D-1)$-sphere, Brauchart method

4. Comparison of unit 2-sphere sampling algorithm performance.

Over the 2-sphere, both the Roberts and Brauchart algorithm have comparable d_0 performance. For both, d_0 decreases faster than $O\left(1/\sqrt{N}\right)$ (see Fig. 25), but they both decrease slower than $O(1/N)$ (see Fig. 26). It looks like Brauchart may perhaps decrease as $O\left(1/\sqrt{N}\right)$ (see Fig. 25), but further examinations would need to be made in order to ascertain the specific rate of decay for these algorithms.

However, it is clear that, over the 2-sphere, both Roberts and Brauchart have better d_0 performance than Marsgalia does, since d_0 decreases as $O(1/N)$ (see Fig. 26), which is faster decay than either Roberts or Brauchart.

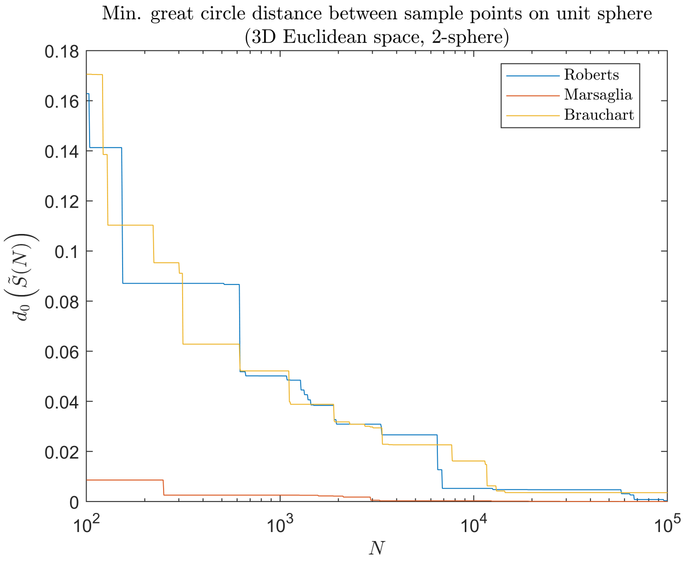


Fig. 23 Minimum great circle distance between points in samples of unit 2-sphere, d_0, no adjustment.

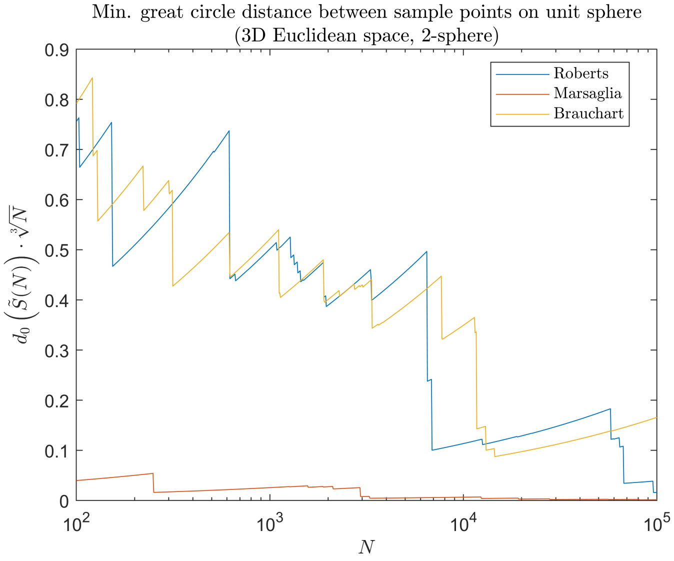


Fig. 24 Minimum great circle distance between points in samples of unit 2-sphere, d_0, adjusted by $\sqrt[3]{N_{\text{sequence}}}$.

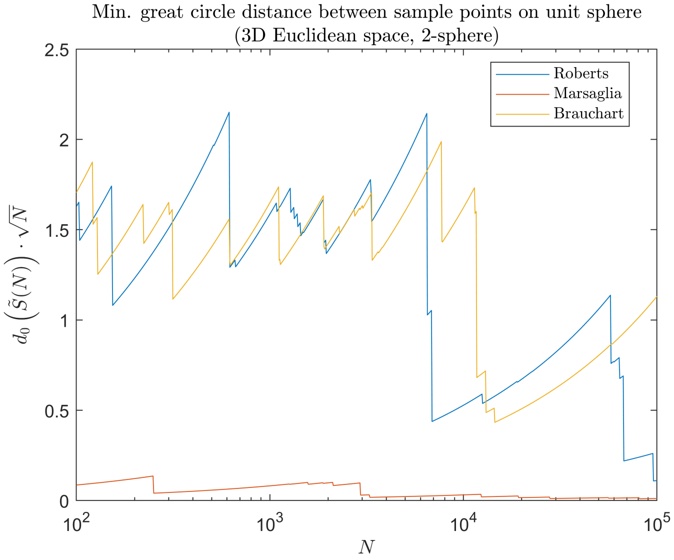


Fig. 25 Minimum great circle distance between points in samples of unit 2-sphere, d_0, adjusted by $\sqrt{N_{\text{sequence}}}$.

5. Comparison of unit 3-sphere sampling algorithm performance.

Over the 3-sphere, the Brauchart sampling algorithm has superior d_0 performance to both Roberts and Marsaglia. Its d_0 decrease is faster than $O\left(1/\sqrt[3]{N}\right)$ (see Fig. 29) but it is at least as slow as $O\left(1/\sqrt{N}\right)$ (see Fig. 30).

On the other hand, the Roberts algorithm still has better d_0 performance than Marsaglia. Both of these algorithms decay faster than $O\left(1/\sqrt{N}\right)$ (see Fig. 30) and slower than $O\left(1/N\right)$ (see Fig. 31). However, it can be seen from Fig. 31 that the decay of Marsaglia is faster than that of Roberts. Nonetheless, further examinations would need to be made in order to ascertain the specific rate of decay for these algorithms.

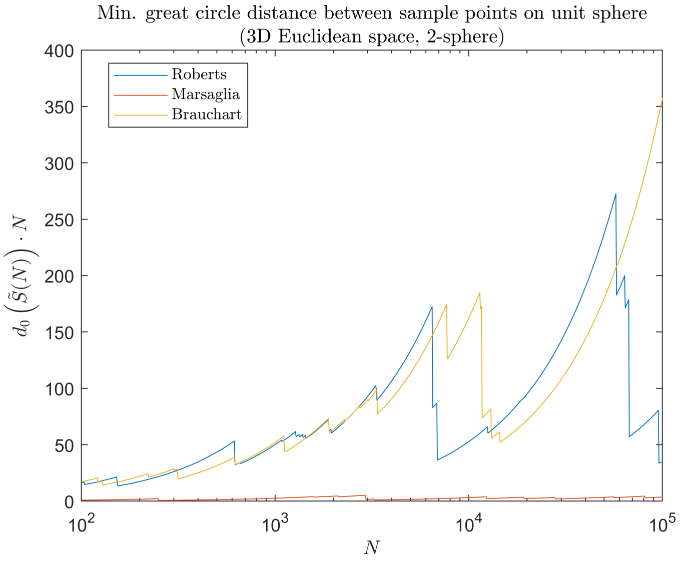


Fig. 26 Minimum great circle distance between points in samples of unit 2-sphere, d_0, adjusted by N_{sequence}.

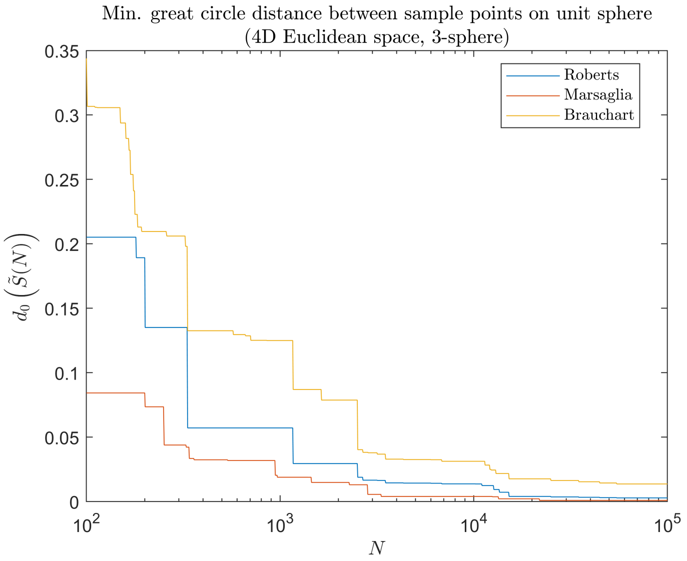


Fig. 27 Minimum great circle distance between points in samples of unit 3-sphere, d_0, no adjustment.

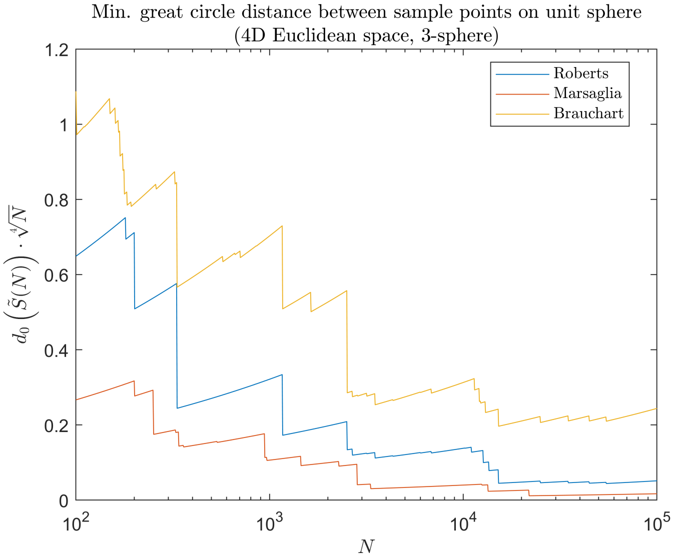


Fig. 28 Minimum great circle distance between points in samples of unit 3-sphere, d_0, adjusted by $\sqrt[4]{N_{\text{sequence}}}$.

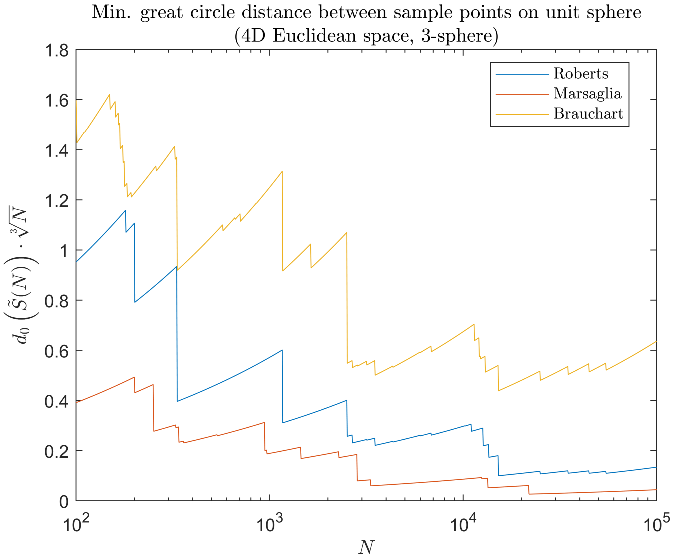


Fig. 29 Minimum great circle distance between points in samples of unit 3-sphere, d_0, adjusted by $\sqrt[3]{N_{\text{sequence}}}$.

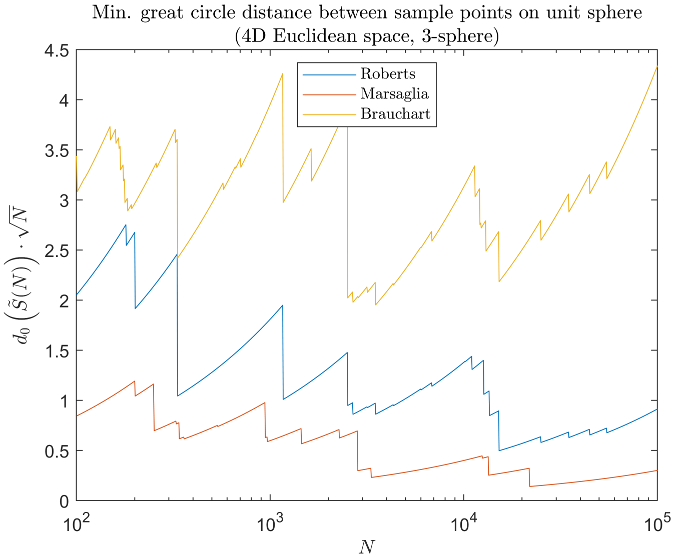


Fig. 30 Minimum great circle distance between points in samples of unit 3-sphere, d_0, adjusted by $\sqrt{N_{\text{sequence}}}$.

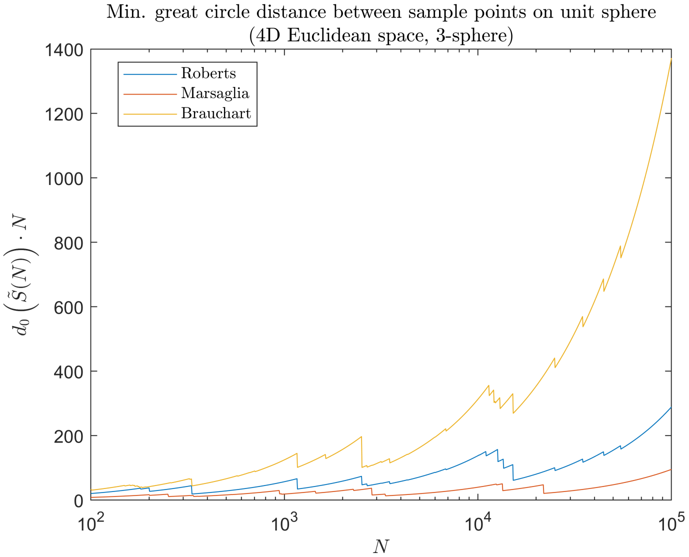


Fig. 31 Minimum great circle distance between points in samples of unit 3-sphere, d_0, adjusted by N_{sequence}.

6. Comparison of unit 5-sphere sampling algorithm performance.

Over the 3-sphere, the Brauchart sampling algorithm has superior d_0 performance to both Roberts and Marsaglia. Its d_0 decrease is faster than $O\left(1/\sqrt[4]{N}\right)$ (see Fig. 34) but it is at least as slow as $O\left(1/\sqrt[3]{N}\right)$ (see Fig. 35).

On the other hand, and unlike for the 2-sphere and 3-sphere cases, the Marsaglia algorithm has better d_0 performance than Roberts. For the Marsaglia algorithm, its d_0 decrease is faster than $O\left(1/\sqrt[3]{N}\right)$ (see Fig. 35) but it is at least as slow as $O\left(1/\sqrt{N}\right)$ (see Fig. 36).

The Roberts algorithm retains the same performance for the 5-sphere as for the 3-sphere. The d_0 decrease for the Roberts algorithm is faster than $O\left(1/\sqrt{N}\right)$ (see Fig. 36) and slower than $O\left(1/N\right)$ (see Fig. 37). It is expected that further examinations would confirm these trends.

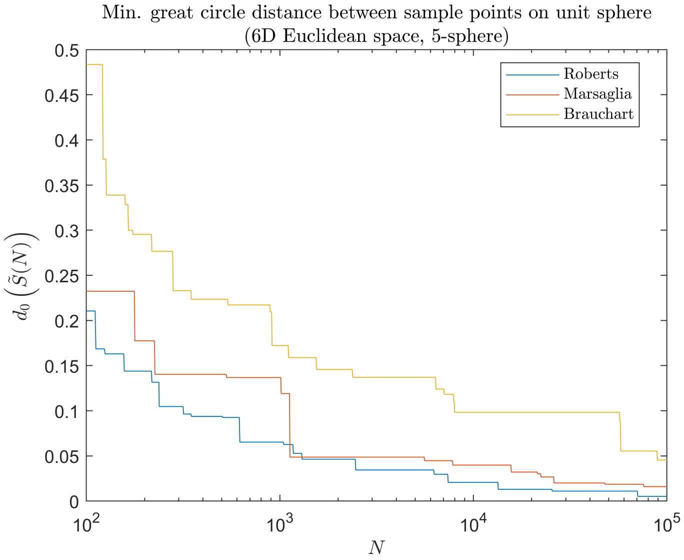


Fig. 32 Minimum great circle distance between points in samples of unit 5-sphere, d_0, no adjustment.

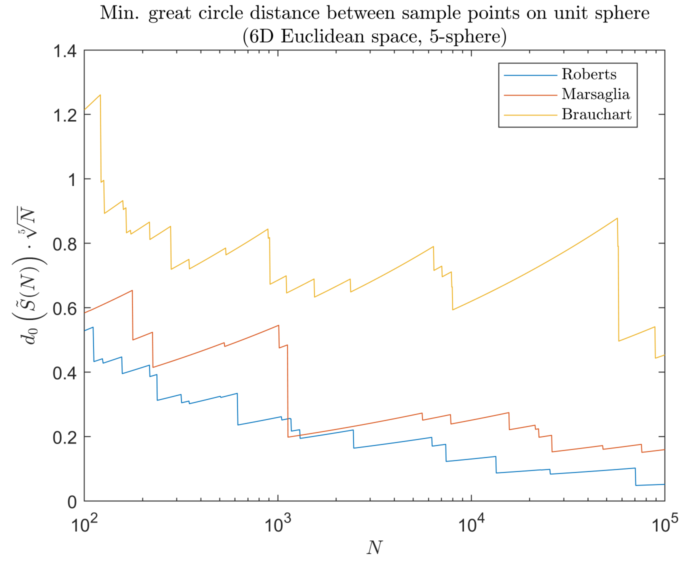


Fig. 33 Minimum great circle distance between points in samples of unit 5-sphere, d_0, adjusted by $\sqrt[5]{N_{\text{sequence}}}$.

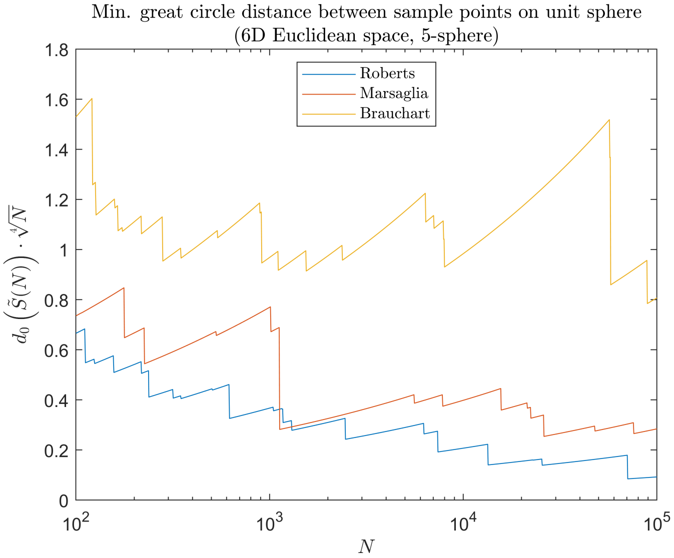


Fig. 34 Minimum great circle distance between points in samples of unit 5-sphere, d_0, adjusted by $\sqrt[4]{N_{\text{sequence}}}$.

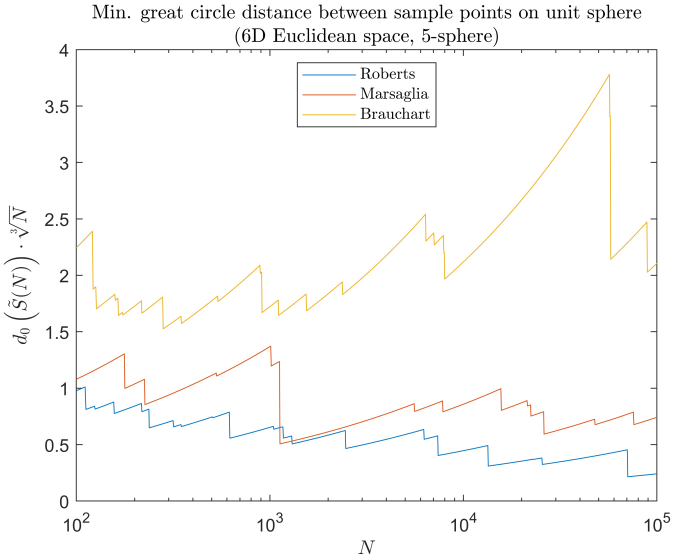


Fig. 35 Minimum great circle distance between points in samples of unit 5-sphere, d_0, adjusted by $\sqrt[3]{N_{\text{sequence}}}$.

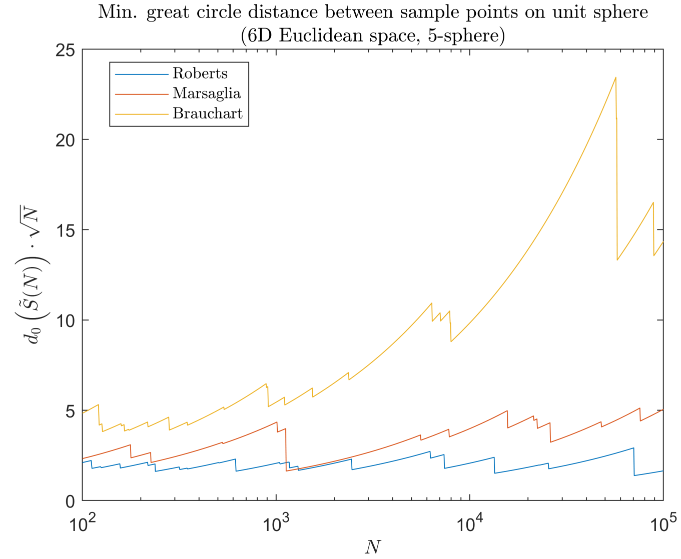


Fig. 36 Minimum great circle distance between points in samples of unit 5-sphere, d_0, adjusted by $\sqrt{N_{\text{sequence}}}$.

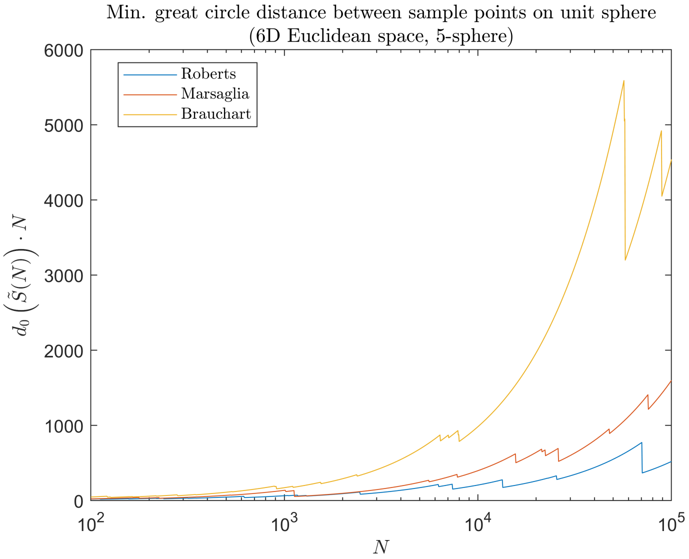


Fig. 37 Minimum great circle distance between points in samples of unit 5-sphere, d_0, adjusted by N_{sequence}.

Acknowledgments

The authors would like to thank Andrew J. Fear, PhD student at the Georgia Tech Space Systems Design Lab, who introduced the authors to potential relationships between normal and chi-square distributions. Additionally, the authors would like to thank Dr. Konstantin Tikhomirov, Assistant Professor at the School of Mathematics at Georgia Tech, for his perspective on the Total Probability of Collision (TPc), which has motivated present and future work on the subject.

References

[1] Fernandez Delicado, R., Bellver Cabello, D., and Navarro Sánchez, M. D., "Synthetic Aperture Radiometer: Revision," *55th International Astronautical Congress of the International Astronautical Federation*, IAC-04, Vol. 2, the International Academy of Astronautics and the International Institute of Space Law, Vancouver, Canada, 2004, pp. 957–965. doi:10.2514/6.IAC-04-B.3.06.

[2] Edelsohn, C. R., "Applications of synthetic aperture radiometry," *Geoscience and Remote Sensing Symposium, 1994*, IGARSS

’94, Vol. 3, IEEE, Pasadena, CA, 1994, pp. 1326–1328. doi:10.1109/IGARSS.1994.399430.

[3] El Maghraby, A. K. S., Grubisic, A., Colombo, C., and Tatnall, A., “A novel approach to microwave interferometric radiometry in the geostationary obit using formation flight,” *67th International Astronautical Congress of the International Astronautical Federation*, IAC-2016, Vol. 3, the International Academy of Astronautics and the International Institute of Space Law, Guadalajara, Mexico, 2016, pp. 1–14.

[4] Schaub, H., and Junkins, J. L., *Analytical Mechanics of Space Systems*, 2^{nd} ed., AIAA Education Series, Reston, VA, 2009, Chap. 14, pp. 593–673.

[5] Wang, G., and Ni, W. T., “Numerical simulation of time delay interferometry for eLISA/NGO,” *Classical and Quantum Gravity*, Vol. 30, No. 6, 2013, pp. 1 – 19. doi:10.1088/0264-9381/30/6/065011.

[6] Wang, G., and Ni, W. T., “Orbit optimization and time delay interferometry for inclined ASTROD–GW formation with half-year precession–period,” *Chinese Physics B*, Vol. 24, No. 5, 2015, pp. 1 – 11.

[7] Alfriend, K., Vadali, S. R., Gurfil, P., How, J., and Breger, L., *Spacecraft Formation Flying: Dynamics, Control and Navigation*, Elsevier Astrodynamics Series, Butterworth–Heinemann (Elsevier Science), Oxford, UK, 2009, Chap. 14, pp. 329–330.

[8] Wertz, J. R., Everett, D. F., and Puschell, J. J., *Space Mission Engineering: The New SMAD*, Space Technology Library, Microcosm Press, Hawthorne, CA, 2011, Chap. 4, pp. 61–82.

[9] Chung, S. J., Bandyopadhyay, S., Foust, R., Subramanian, G. P., and Hadaegh, F., “Review of Formation Flying and Constellation Missions Using Nanosatellites,” *Journal of Spacecraft and Rockets*, Vol. 53, No. 3, 2016, pp. 567–578. doi:10.2514/1.A33291.

[10] Scharf, D. P., Hadaegh, F. Y., and Ploen, S. R., “A survey of spacecraft formation flying guidance and control (part 1): guidance,” *Proceedings of the 2003 American Control Conference, 2003*, Vol. 2, IEEE, Denver, CO, 2003, pp. 1733–1739. doi:10.1109/ACC.2003.1239845.

[11] Scharf, D. P., Hadaegh, F. Y., and Ploen, S. R., “A survey of spacecraft formation flying guidance and control (part II): control,” *Proceedings of the 2004 American Control Conference*, Vol. 4, IEEE, Boston, MA, 2004, pp. 2976–2985.

[12] Alfriend, K., Vadali, S. R., Gurfil, P., How, J., and Breger, L., *Spacecraft Formation Flying: Dynamics, Control and Navigation*, Elsevier Astrodynamics Series, Butterworth–Heinemann (Elsevier Science), Oxford, UK, 2009, Chap. 1, pp. 1–11.

[13] Bryson, A. E., and Ho, Y., *Applied optimal control: optimization, estimation, and control*, Taylor and Francis Group, New York, 1975, Chap. 4, pp. 128–147.

[14] Stengel, R. F., *Optimal Control and Estimation*, Dover Books on Advanced Mathematics, Dover Publications, 1994, Chap. 3, pp. 184–298.

[15] Vu, T., and Rahmani, A., “Analysis of a distributed estimation and control scheme for formation flying spacecraft,” *Aerospace Science and Technology*, Vol. 73, 2018, pp. 232–238. doi:10.1016/j.ast.2017.10.028.

[16] Holzinger, M. J., and McMahon, J. W., “Decentralized Mean Orbit-Element Formation Guidance, Navigation, and Control: Part 1,” *AIAA/AAS Astrodynamics Specialist Conference*, AIAA/AAS, Minneapolis, MN, 2012, pp. 1–13. doi:10.2514/6.2012-4515.

[17] McMahon, J. W., and Holzinger, M. J., “Decentralized Mean Orbit-Element Formation Guidance, Navigation, and Control: Part 2,” *AIAA/AAS Astrodynamics Specialist Conference*, AIAA/AAS, Minneapolis, MN, 2012, pp. 1–20. doi:10.2514/6.2012-4516.

[18] Douglass, E. J., Holzinger, M. J., McMahon, J. W., and Jaunzemis, A., “Formation control problems for decentralized spacecraft systems,” *AAS/AIAA Astrodynamics Specialist Conference*, AAS/AIAA, Hilton Head, SC, 2013, pp. 1337–1356.

[19] Nazari, M., Butcher, E. A., Yucelen, T., and Sanyal, A. K., “Decentralized Consensus Control of a Rigid-Body Spacecraft Formation with Communication Delay,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 4, 2016, pp. 838–851. doi:10.2514/1.G001396.

[20] Mesbahi, M., and Egerstedt, M., *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.

[21] Tschan, C., Yucel, A., and Nguyen, N. (eds.), “Roadmap for Intelligent Systems in Aerospace,” Tech. rep., AIAA, Intelligent Systems Technical Committee (ISTC), Hampton, VA, June 2016.

[22] NASA Office of the Chief Technologist, “TA 4: Robotics and Autonomous Systems,” *2015 NASA Technology Roadmaps*, 2015, pp. 2 – 188.

[23] Alami, R., Chatila, R., Fleury, S., Ghallab, M., and Ingrand, F., "An architecture for autonomy," *The International Journal of Robotics Research*, Vol. 17, No. 4, 1998, pp. 315–337.

[24] Russell, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, CreateSpace Independent Publishing Platform, 2016, Chaps. 1–2, pp. 1–63.

[25] Newman, L. K., C, F. R., G, D. M., and D, H. M., "Evolution and Implementation of the NASA Robotic Conjunction Assessment Risk Analysis Concept of Operations," *2014 Advanced Maui Optical and Space Surveillance Technologies Conference Proceedings*, Maui, HI, 2014, pp. 1–14.

[26] Patera, R. P., "General Method for Calculating Satellite Collision Probability," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, 2001, pp. 716–722. doi:10.2514/2.4771.

[27] Patera, R. P., "Satellite Collision Probability for Nonlinear Relative Motion," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 5, 2003, pp. 728–733. doi:10.2514/2.5127.

[28] Slater, G. L., Byram, S. M., and Williams, T. W., "Collision Avoidance for Satellites in Formation Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 5, 2006, pp. 1140–1146. doi:10.2514/1.16812.

[29] Chan, F. K., *Spacecraft collision probability*, Aerospace Press ; American Institute of Aeronautics and Astronautics, El Segundo, CA; Reston, VA, 2008.

[30] Serra, R., Arzelier, D., Joldes, M., Lasserre, J.-B., Rondepierre, A., and Salvy, B., "Fast and Accurate Computation of Orbital Collision Probability for Short-Term Encounters," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 5, 2016, pp. 1009–1021. doi:10.2514/1.G001353.

[31] Lee, S., Lyu, H., and Hwang, I., "Analytical Uncertainty Propagation for Satellite Relative Motion Along Elliptic Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 7, 2016, pp. 1593–1601. doi:10.2514/1.G001848.

[32] Hejduk, M. D., and Johnson, L. C., "Approaches to Evaluating Probability of Collision Uncertainty," *26th AAS/AIAA Space Flight Mechanics Meeting*, AAS/AIAA, Napa, CA, 2016, pp. 1–15.

[33] D, H. M., "CARA Risk Assessment Thresholds," *2016 International Conjunction Assessment Technical Advisory Council Meeting*, Centre National d'Etudes Spatiales, Paris, France, 2016, pp. 1–26.

[34] Hur-Diaz, S., Ruschmann, M., Heyne, M., and Phillips, M., "Computing Collision Probability Using Linear Covariance and Unscented Transforms," *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA, Boston, MA, 2013, pp. 1–10. doi:10.2514/6.2013-5189.

[35] Julier, S. J., and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422. doi:10.1109/JPROC.2003.823141.

[36] Frigm, R. C., Hejduk, M. D., Johnson, L. C., and Plakalovic, D., "Total Probability of Collision as a Metric for Finite Conjunction Assessment and Collision Risk Management," *2015 Advanced Maui Optical and Space Surveillance Technologies Conference Proceedings*, Maui, HI, 2015, pp. 1–14.

[37] Heil, C., *Introduction to Real Analysis*, Graduate Texts in Mathematics, Springer International Publishing, 2019, Chap. 1, pp. 15–32.

[38] Papoulis, A., and Pillai, S. U., *Probability, Random Variables, and Stochastic Processes*, 4th ed., McGraw-Hill, Boston, 2002, Chap. 9, pp. 373–434.

[39] Bartle, R. G., *The Elements of Real Analysis*, 1st ed., John Wiley & Sons, Inc., New York, 1964, Chap. 1, pp. 58–97.

[40] Weisstein, E. W., "Sphere (From MathWorld–A Wolfram Web Resource)," , Dec 2019. URL `http://mathworld.wolfram.com/Sphere.html`, online; accessed 20 December 2019.

[41] Hocking, J. G., and Young, G. S., *Topology*, 1st ed., Addison-Wesley, Inc., Reading, MA, 1961, Chap. 1, pp. 1–36.

[42] Hill, G. W., "Researches in the Lunar Theory," *American Journal of Mathematics*, Vol. 1, No. 1, 1878, pp. 5–26. doi:10.2307/2369430.

[43] Clohessy, W. H., and Wiltshire, R. S., "Terminal guidance system for satellite rendezvous," *Journal of Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658. doi:10.2514/8.8704.

[44] Mahalanobis, P. C., “On the generalised distance in statistics,” *Proceedings of the National Institute of Sciences of lndia*, Vol. 2, No. 1, 1936, pp. 49–55.

[45] McLachlan, G. J., “Mahalanobis distance,” *Resonance*, Vol. 4, No. 6, 1999, pp. 20–26. doi:10.1007/BF02834632.

[46] Bhattacharya, P. K., and Burman, P., *Theory and Methods of Statistics*, Academic Press, 2016, Chap. 4, pp. 385–388. doi:10.1016/B978-0-12-802440-9.00012-6.

[47] De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L., “The Mahalanobis distance,” *Chemometrics and Intelligent Laboratory Systems*, Vol. 50, No. 1, 2000, pp. 1 – 18. doi:10.1016/S0169-7439(99)00047-7.

[48] Hubert, M., Rousseeuw, P. J., and Van Aelst, S., “Multivariate Outlier Detection and Robustness,” *Data Mining and Data Visualization*, Handbook of Statistics, Vol. 24, edited by C. Rao, E. Wegman, and J. Solka, Elsevier, 2005, Chap. 10, pp. 263 – 302. doi:10.1016/S0169-7161(04)24010-X.

[49] Mathai, A. M., and Provost, S. B., *Quadratic Forms in Random Variables: Theory and Applications*, Statistics: Textbooks and Monographs, Marcel Dekker, Inc., New York, 1992, Chap. 5, pp. 195–196.

[50] Kroese, D. P., Taimre, T., and Botev, Z. I., *Handbook of Monte Carlo Methods*, John Wiley & Sons, Ltd, 2011, Chap. 4, pp. 143–146. doi:10.1002/9781118014967.ch4.

[51] Bernstein, D. S., *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed., Princeton University Press, Princeton, NJ, 2009, Chap. 8, pp. 417–542.

[52] Durrett, R., *Probability: theory and examples*, 5th ed., Cambridge University Press, Durham, NC, 2019, Chap. 1, pp. 15–32.

[53] Roberts, M., “The Unreasonable Effectiveness of Quasirandom Sequences,” , Aug 2018. URL `http://extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/`, online; accessed 19 November 2019.

[54] Brauchart, J. S., Dick, J., and Fang, L., “Spatial low-discrepancy sequences, spherical cone discrepancy, and applications in financial modeling,” *Journal of Computational and Applied Mathematics*, Vol. 286, 2015, pp. 28–53. doi:10.1016/j.cam.2015.02.023.

[55] Niederreiter, H., “Low-discrepancy and low-dispersion sequences,” *Journal of Number Theory*, Vol. 30, No. 1, 1988, pp. 51 – 70. doi:10.1016/0022-314X(88)90025-X.

[56] Brauchart, J. S., and Grabner, P., “Distributing many points on spheres: Minimal energy and designs,” *Journal of Complexity*, Vol. 31, No. 3, 2015, pp. 293–326. doi:10.1016/j.jco.2015.02.003.

[57] Muller, M. E., “A Note on a Method for Generating Points Uniformly on N-Dimensional Spheres,” *Communications of the ACM*, Vol. 2, No. 4, 1959, p. 19–20. doi:10.1145/377939.377946, URL `https://doi.org/10.1145/377939.377946`.

[58] Marsaglia, G. W., “Choosing a Point from the Surface of a Sphere,” *Annals of Mathematical Statistics*, Vol. 43, No. 2, 1972, pp. 645–646. doi:10.1214/aoms/1177692644, URL `https://doi.org/10.1214/aoms/1177692644`.

[59] Weisstein, E. W., “Golden Ratio (From MathWorld–A Wolfram Web Resource),” , Jan 2020. URL `http://mathworld.wolfram.com/Latitude.html`, online; accessed 6 January 2020.

[60] Brannon, R. M., *Rotation, Reflection, and Frame Changes: Orthogonal tensors in computational engineering mechanics*, 1st ed., IOP Publishing, London, 2018, Chap. 17, pp. 17–14 – 17–15. doi:10.1088/978-0-7503-1454-1.

[61] Weisstein, E. W., “Latitude (From MathWorld–A Wolfram Web Resource),” , Jan 2020. URL `http://mathworld.wolfram.com/Latitude.html`, online; accessed 6 January 2020.

[62] Blumenson, L. E., “A Derivation of n-Dimensional Spherical Coordinates,” *The American Mathematical Monthly*, Vol. 67, No. 1, 1960, pp. 63–66. URL `http://www.jstor.org/stable/2308932`.