AIAA SciTech Forum
11–15 & 19–21 January 2021, VIRTUAL EVENT
AIAA Scitech 2021 Forum

10.2514/6.2021-1855



# 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 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 also 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 new method is suggested as an alternative to random sampling algorithms such as Monte Carlo methods or the Unscented Transform.**



## I. Nomenclature

| | | |
|---|---|---|
| $\mathbb{B}_r^n(\underline{x})$ | = | ball of radius $r \in (0,\infty)$ centered at $\underline{x} \in \mathbb{R}^n$ |
| CARA | = | "Conjunction Analysis and Risk Assessment" group at NASA Goddard Space Flight Center |
| $\text{cdf}_X$ | = | cumulative distribution function (cdf) of random variable $X$ |
| $\text{cdf}_{X\|Y}$ | = | conditional cdf of random variable $X$ given outcome of random variable $Y$ |
| $D$ | = | dimension of points in the unit $(D-1)$-sphere |
| $d_0\left(\tilde{S}(N)\right)$ | = | minimum arc length between any pair of points in the unit hypersphere sample $\tilde{S}(N)$ |
| $d_{\text{max}}$ | = | cutoff Mahalanobis distance (MSS sample parameter) |
| $d'_{\text{max}}$ | = | "transition" cutoff Mahalanobis distance |
| $d_{\underline{\mathbf{R}}}$ | = | dimensionality of position states (1, 2 or 3) |
| $D_{\underline{X}}(\underline{x})$ | = | Mahalanobis distance of instance $\underline{x} \in \mathbb{R}^n$ of random variable $\underline{X} \in \mathbb{R}^n$ |
| $F_{\underline{X}}$ | = | "alternative", Mahalanobis distance-based cdf for normal random variable $\underline{X} \in \mathbb{R}^n$ |
| $\text{KPC}_{i,j}(t\|t_0)$ | = | kinematic probability of collision between agents $i$ and $j$ at time $t$, given initial conditions at time $t_0$ |
| $L_d(\underline{X})$ | = | $d$-Mahalanobis contour of normally distributed random variable $\underline{X} \in \mathbb{R}^n$ |
| $l_{i,j}$ | = | $i$-$j$ joint hard-body radius |
| $\mathcal{N}\left(\underline{\mu}, \Sigma\right)$ | = | normal distribution with mean $\underline{\mu} \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{R}^{n\times n}$, $\Sigma > 0$ |
| MSS | = | Mahalanobis Shell Sampling algorithm |
| $N_{\text{shells}}$ | = | number of shells (MSS sample parameter) |
| $N_{\text{samples/shell}}$ | = | number of samples per shell (MSS sample parameter) |
| $n_{\underline{\mathbf{X}}}$ | = | dimensionality of dynamic state |
| $\text{pdf}_X$ | = | probability density function (pdf) of random variable $X$ |
| $\text{pdf}_{X\|Y}$ | = | conditional pdf of random variable $X$ given outcome of random variable $Y$ |
| $\phi_n$ | = | generalized golden ratio numbers of order $n$ $\mathbb{N}$ |
| $\Phi_{i,j}(t,t_0)$ | = | $i$-$j$ relative state transition matrix (from time $t_0$ to time $t$) |
| $\underline{\mathbf{R}}_i$ | = | position of the center of mass of agent $i$ |
| $\underline{\mathbf{R}}_{i,j}$ | = | $i$-$j$ relative position |

*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.

| | | |
|---|---|---|
| $\tilde{S}(N)$ | = | sample of points on the unit hypersphere with size $N$ |
| $\Sigma_{\underline{\mathbf{R}}_{i,j}}$ | = | $i$-$j$ relative position covariance |
| $\Sigma_{\underline{\mathbf{X}}_{i,j}}$ | = | $i$-$j$ relative state covariance |
| $\mathbb{S}_r^{(n-1)}(\underline{x})$ | = | $(n-1)$-sphere of radius $r \in (0,\infty)$ centered at $\underline{x} \in \mathbb{R}^n$ |
| SVD | = | singular value decomposition |
| TPc | = | "total probability of collision", as defined by NASA CARA |
| $V_d(\underline{X})$ | = | $d$-Mahalanobis volume of normally distributed random variable $\underline{X} \in \mathbb{R}^n$ |
| $V_{d_1}^{d_2}(\underline{X})$ | = | $d_1, d_2$-Mahalanobis shell of normally distributed random variable $\underline{X} \in \mathbb{R}^n$ |
| $V_{i,j}$ | = | $i$-$j$ intersection volume |
| $x$ | = | instance (written in lower case) of random variable $X$ (written in uppercase) |
| WPC | = | window probability of collision (equivalent in meaning to TPc) |
| $\underline{\mathbf{X}}_i$ | = | dynamic state of agent $i$ |
| $\underline{\mathbf{X}}_{i,j}$ | = | $i$-$j$ relative dynamic state |
| $\chi_n^2$ | = | chi-square distribution with $n \in \mathbb{N}$ degrees of freedom |



## II. Introduction

Spacecraft formations are attractive from a space mission design perspective. In particular, spacecraft formation flying (SFF) can enable missions with increased system robustness, as deterioration or failure of an agent in a spacecraft formation may only cause performance degradation in the mission, rather than causing the end of the mission.[1] SFF missions may also have performance improvements over their mission lifetimes due to the ability to replace failed agents or add new ones. This adds a new layer to space mission architecture options beyond traditional, monolithic spacecraft missions.[2] Additionally, missions that implement SFF have an opportunity for enhanced system flexibility through improved "adaptability, scalability, evolvability, and maintainability".[3] Spacecraft formations can also enable high precision scientific missions by distributing a formation over regions larger than those spanned by large, monolithic spacecraft and by using sensor fusion.[4] Doing so has a plethora of applications, such as gravimetry,[5] weather forecasting and climate monitoring,[6] exoplanet detection,[7] gravitational wave detection,[8] and more.

A fundamental challenge to any spacecraft formation mission is avoiding collisions of two kinds: first, collisions among agents within the formation; and second, collisions between external debris and members of the formation. Even though agents in a spacecraft formation operate in close enough proximity to each other that their individual dynamics can be modeled as small variations near a reference orbit, they must operate without ever colliding or coming dangerously close together.[1, 9] Furthermore, several Earth-orbit regimes, such as Low Earth Orbit (LEO) and Geostationary Orbit (GEO), have become hosts to surging populations of debris, whose presence in these orbits is hazardous to present and future missions.[10] Space debris is a pervasive, "self-perpetuating" concern, since new objects are introduced to the environment every time a new mission is made operational.[11] Collision avoidance maneuvers remain the "single most important technique in managing the risk associated with space object collision".[12] However, the paramount concern of collision avoidance must be balanced with the reality of limited onboard resources. First, spacecraft in a formation must not drift apart from one another without bound (i.e. formations must be stable). Second, it is desirable that spacecraft in a formation have relative geometry that repeats in a way that enables certain observation properties.[13] However, errors in state knowledge, in dynamics modeling, and in maneuver timing and pointing exist and cannot be fully eliminated for any spacecraft mission. Consequently, for SFF missions, the combination of these facts imposes the need for constant station-keeping, which implies that SFF mission lifetime is constrained by onboard propellant.[1] Thus, accurate collision risk computations are needed in order to sufficiently safeguard spacecraft formations without unduly compromising onboard propellant (and, consequently, mission lifetime) by overestimating collision risk.

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 referenced in other work as Pc,[14] restricted to cases for which the relative dynamic states are normally distributed random variables. 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, either through planar approximations or using three-dimensional geometry, and/or by attempting to include certain nuances of the dynamics.[9, 15–19] Sampling methods are attractive in this context because they allow relaxing assumptions in KPC computation, namely, by allowing the propagation of individual particles through arbitrary processes, which gives sampling methods general applicability.

The second purpose of this work is to enable the efficient computation of a new collision likelihood measure that could be more operationally useful than KPC: the probability of the event that two agents may collide "at any time within an arbitrary, compact time window". This "window probability of collision" (WPC) has been motivated and referred to as "Total Probability of Collision (TPc),[20] and it 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 their initial relative state statistics. 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 WPC 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 TPc/WPC computation.

However, the practice of the Conjunction Assessment and Risk Analysis (CARA) group at NASA Goddard Space Flight Center (GSFC) elucidates the difficulties with implementing sampling methods for collision probability computation. At NASA CARA, when analyzing conjunction events between debris and individual spacecraft of interest, 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.[14, 21] Contemplated mitigation for such high risk events includes intervening actions up to collision avoidance maneuver planning and execution.[14] 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.[14] 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.[22] This approach demonstrates the need for high resolution when computing KPC. Performing KPC computations through Monte Carlo methods is challenging if the debris CARA community requirement that KPC values above 1E-7 are considered significant were to be adopted for SFF collision risk assessment.[14, 23] Since it is cost prohibitive to have ground-based SFF control,[24] and with limited computational onboard resources, implementing Monte Carlo methods for KPC computation in SFF missions 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 also been proposed.[25] 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),[26] 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 to reproduce low-valued integrals over arbitrary regions of the original distribution.

This work is organized as follows. First, the Background section (Section III) expounds the notation used in this work, defines collision events topologically, and defines the kinematic probability of collision (KPC). 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 examine 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 VI) 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. Finally, the kinematic probability of collision (KPC) is characterized and discussed.

### A. Notation

Vectors are underlined, while matrices and functions are not. Although boldface is reserved for multidimensional variables (i.e. vectors and matrices), sometimes boldfacing such variables may be avoided for clarity. Let $\underline{\mathbf{X}} \in \mathbb{R}^{n_{\underline{\mathbf{X}}}}$

and $\underline{\mathbf{R}} \in \mathbb{R}^{d_{\underline{\mathbf{R}}}}$ denote a dynamic state and position state, respectively. The dimensions of $\underline{\mathbf{X}}$ and $\underline{\mathbf{R}}$ are denoted by $n_{\underline{\mathbf{X}}}$ and $d_{\underline{\mathbf{R}}}$, respectively. (Note: $d_{\underline{\mathbf{R}}} \in \{1, 2, 3\}$.) When used with the subscript $i$, $\underline{\mathbf{X}}_i$ and $\underline{\mathbf{R}}_i$ denote the dynamic state and position of agent $i$, respectively. It is understood that $\underline{\mathbf{R}}_i$ specifically refers to the position of the center of mass of agent $i$. When used with a composite subscript such as "$i$-$j$", $\underline{\mathbf{X}}_{i,j}$ and $\underline{\mathbf{R}}_{i,j}$ denote the dynamic state and position of agent $i$ and relative to agent $j$, respectively, i.e. $\underline{\mathbf{X}}_{i,j} \doteq \underline{\mathbf{X}}_i - \underline{\mathbf{X}}_j$ and $\underline{\mathbf{R}}_{i,j} \doteq \underline{\mathbf{R}}_i - \underline{\mathbf{R}}_j$. When written in uppercase, $\underline{\mathbf{X}}$ and $\underline{\mathbf{R}}$ denote an uncertain dynamic state and uncertain position, respectively, i.e. a dynamic state $\underline{\mathbf{X}}(t)$ and position $\underline{\mathbf{R}}(t)$ are random variables for any time $t$. Conversely, when written in lowercase, $\underline{\mathbf{x}}$ and $\underline{\mathbf{r}}$ denote specific, deterministic "instances" or values that $\underline{\mathbf{X}}$ and $\underline{\mathbf{R}}$ may take on, respectively. When overlaid by a bar, $\bar{\underline{\mathbf{X}}}$ and $\bar{\underline{\mathbf{R}}}$ denote the expected (or "mean") values of $\underline{\mathbf{X}}$ and $\underline{\mathbf{R}}$, respectively, i.e. $\bar{\underline{\mathbf{X}}} \doteq \mathbb{E}\left[\underline{\mathbf{X}}\right]$ and $\bar{\underline{\mathbf{R}}} \doteq \mathbb{E}\left[\underline{\mathbf{R}}\right]$. In this work, the variable $\Sigma$ denotes a covariance matrix. Thus, when in a subscript, $\Sigma_{\underline{\mathbf{X}}}$ and $\Sigma_{\underline{\mathbf{R}}}$ denote the covariance of $\underline{\mathbf{X}}$ and $\underline{\mathbf{R}}$, respectively, i.e.

$$\Sigma_{\underline{\mathbf{X}}} \doteq \mathbb{E}\left[\left(\underline{\mathbf{X}} - \bar{\underline{\mathbf{X}}}\right)\left(\underline{\mathbf{X}} - \bar{\underline{\mathbf{X}}}\right)^T\right] \tag{III.1}$$

and

$$\Sigma_{\underline{\mathbf{R}}} \doteq \mathbb{E}\left[\left(\underline{\mathbf{R}} - \bar{\underline{\mathbf{R}}}\right)\left(\underline{\mathbf{R}} - \bar{\underline{\mathbf{R}}}\right)^T\right] \tag{III.2}$$

It is assumed that $\underline{\mathbf{R}}$ is a linear combination of the components of $\underline{\mathbf{X}}$, i.e. that there exists a mapping $g : \mathbb{R}^{n_{\underline{\mathbf{X}}}} \to \mathbb{R}^{d_{\underline{\mathbf{R}}}}$ defined by the rule $\underline{\mathbf{R}} = g\left(\underline{\mathbf{X}}\right) = \mathbf{M}\underline{\mathbf{X}}$ for some matrix $\mathbf{M} \in \mathbb{R}^{d_{\underline{\mathbf{R}}} \times n_{\underline{\mathbf{X}}}}$. In particular, if $\underline{\mathbf{X}}$ is partitioned as $\underline{\mathbf{X}}^T = \left[\underline{\mathbf{R}}^T, \underline{\mathbf{U}}^T\right]$ (where $\underline{\mathbf{U}} \in \mathbb{R}^{n_{\underline{\mathbf{X}}} - d_{\underline{\mathbf{R}}}}$ is a vector whose components are the components of $\underline{\mathbf{X}}$ different from those of $\underline{\mathbf{R}}$), then $\mathbf{M}$ is given by $\mathbf{M} = \left[\mathbb{I}_{d_{\underline{\mathbf{R}}}}, \mathbf{0}_{d_{\underline{\mathbf{R}}} \times \left(n_{\underline{\mathbf{X}}} - d_{\underline{\mathbf{R}}}\right)}\right]$. Additionally, when referring to a matrix (e.g. the covariance matrix $\Sigma$), the notation "$> 0$" implies that such is a symmetric, positive definite matrix.[27]

It is assumed that $\underline{\mathbf{X}}(t)$ and $\underline{\mathbf{R}}(t)$ have continuous distributions for any time $t$. In an obvious way, variables or functions such as $\underline{\mathbf{X}}(t)$ and its probability density function, $\mathrm{pdf}_{\underline{\mathbf{X}}|t}(\cdot|t)$, are dependent on their initial conditions. This is highlighted by making such dependencies explicit, i.e. by denoting $\underline{\mathbf{X}}(t)$ by $\underline{\mathbf{X}}(t|t_0)$, and denoting $\mathrm{pdf}_{\underline{\mathbf{X}}|t}(\cdot|t)$ by $\mathrm{pdf}_{\underline{\mathbf{X}}|t,t_0}(\cdot|t, t_0)$. However, sometimes these dependencies are not expressed for the sake of simplicity. These comments also apply to $\underline{\mathbf{R}}(t)$ and its probability density function, $\mathrm{pdf}_{\underline{\mathbf{R}}|t}(\cdot|t)$, as well as to other relevant variables and functions.

If the dynamics of $\underline{\mathbf{X}}$ are linear, its state transition matrix (STM) from an initial time $t_0$ to a final time $t$ is denoted by $\Phi_{\underline{\mathbf{X}}}(t, t_0)$. For simplicity, the STM of the relative state $\underline{\mathbf{X}}_{i,j}$ is denoted by $\Phi_{i,j}(t, t_0)$.

### B. Definition of a collision event

A collision event (between two agents) occurs whenever their respective physical, nonempty "volumes" in $d_{\underline{\mathbf{R}}}$-dimensional space have a nonempty intersection. In other words, a collision event means 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). (Note: in this work, norm operations refer to the Euclidean norm.[28])

**Definition III.1** ($n$-ball and $(n-1)$-sphere [28, 29])**.** The $n$-ball of radius $r$, centered at $\underline{x} \in \mathbb{R}^n$, denoted by $\mathbb{B}_r^n\left(\underline{x}\right)$, is defined as the set

$$\mathbb{B}_r^n\left(\underline{x}\right) \doteq \left\{\underline{y} \in \mathbb{R}^n : \left\|\underline{x} - \underline{y}\right\| < r, r > 0\right\} \tag{III.3}$$

The $(n-1)$-sphere of radius $r$, centered at $\underline{x} \in \mathbb{R}^n$, denoted by $\mathbb{S}_r^{(n-1)}\left(\underline{x}\right)$, is defined as the set

$$\mathbb{S}_r^{(n-1)}\left(\underline{x}\right) \doteq \left\{\underline{y} \in \mathbb{R}^n : \left\|\underline{x} - \underline{y}\right\| = r, r > 0\right\} \tag{III.4}$$

Note: when the dimensionality of elements in $\mathbb{B}_r^n\left(\underline{x}\right)$ is implicit, it will be referred to as $\mathbb{B}_r\left(\underline{x}\right)$ for simplicity. Also, the symbol "$\mathbb{S}^{(n-1)}$" denotes $\mathbb{S}_1^{(n-1)}(0)$, i.e. the unit $(n-1)$-sphere centered at the origin. ◊

**Notation III.2** (Characteristic length)**.** Let the "body of agent $i$", $B_i$, be defined as the set

$$B_i \doteq \left\{\underline{x} \in \mathbb{R}^{d_{\underline{\mathbf{R}}}} : \underline{x} \text{ is in the body of agent } i\right\} \tag{III.5}$$

Then, the $i^{th}$ characteristic length, $l_i$, is defined as

$$l_i \doteq \sup_{\underline{x} \in B_i} \left\|\underline{x} - \underline{\mathbf{r}}_i\right\| \tag{III.6}$$

Note: the body of agent $i$ is circumscribed within $\mathbb{B}_{l_i}^{d_{\underline{\mathbf{R}}}}\left(\underline{\mathbf{r}}_i\right)$, i.e. $B_i \subseteq \mathbb{B}_{l_i}^{d_{\underline{\mathbf{R}}}}\left(\underline{\mathbf{r}}_i\right)$. ◊

**Definition III.3** (Hard-body radius simplification)**.** The body of agent $i$, $B_i$, satisfies $B_i = \mathbb{B}_{l_i}^{d_{\underline{\mathbf{R}}}}(\underline{\mathbf{r}}_i)$ by assumption. ◇

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

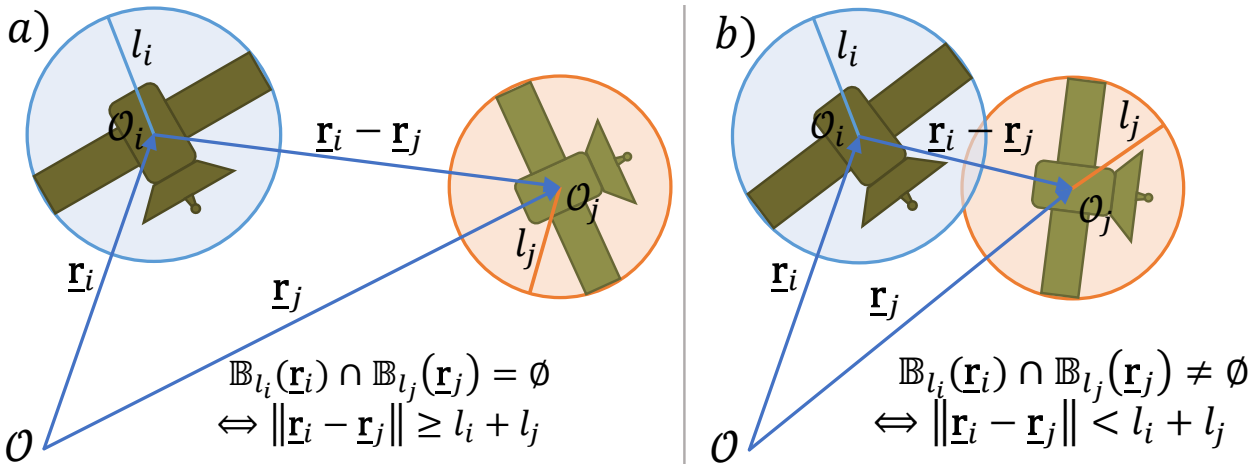


**Fig. 1 Agents under the HBR simplification: a) not colliding, and b) colliding. Note: these circles represent $d_{\underline{\mathbf{R}}}$-balls that circumscribe agent bodies, not position pdf's.**

Suppose there exist two agents $i$ and $j$ in proximity. Through the HBR simplification, $l_i$ represents a no-contact zone, i.e. agent $j$ does not collide with any other agent $j$ ($i \neq j$) if no point belonging to the body of agent $j$ becomes closer to the $i^{th}$ center of mass than a distance $l_i$. 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.4** (Collision event)**.** Assume the HBR simplification holds (see Definition III.3). Then, a collision event between agents $i$ and $j$ occurs when there is a nonempty intersection between the "volumes spanned" by agent $i$ ($\mathbb{B}_{l_i}(\underline{\mathbf{r}}_i)$) and agent $j$ ($\mathbb{B}_{l_j}(\underline{\mathbf{r}}_j)$), i.e. a collision occurs whenever

$$\mathbb{B}_{l_i}(\underline{\mathbf{r}}_i) \cap \mathbb{B}_{l_j}(\underline{\mathbf{r}}_j) \neq \emptyset \qquad \diamond \tag{III.7}$$

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.

**Notation III.5** (Joint hard-body radii)**.** The $i$-$j$ joint hard-body radius, denoted by $l_{i,j}$, is defined as

$$l_{i,j} \doteq l_i + l_j \qquad \diamond \tag{III.8}$$

**Notation III.6** (Intersection volumes)**.** The $i$-$j$ intersection volume, denoted by $V_{i,j}$, is defined as the set

$$V_{i,j} \doteq \mathbb{B}_{l_{i,j}}^{d_{\underline{\mathbf{R}}}}\left(\mathbf{0}_{d_{\underline{\mathbf{R}}}\times 1}\right) = \left\{\underline{\mathbf{r}} \in \mathbb{R}^{d_{\underline{\mathbf{R}}}} : \|\underline{\mathbf{r}}\| < l_{i,j}\right\} \qquad \diamond \tag{III.9}$$

**Proposition III.7** (Collision condition under HBR simplification)**.** Assume the HBR simplification holds (see Definition III.3). Suppose a collision event between two agents $i$ and $j$ is occurring (see Definition III.4). Then, the following statements are equivalent:

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

### C. Definition of Kinematic Probability of Collision (KPC)

In Subsection III.B, 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.7 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.4, 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.8** (Kinematic probability of collision)**.** The kinematic probability of collision of agents $i$ and $j$ at time $t$, denoted by $\mathrm{KPC}_{i,j}\,(t)$, is defined as the probability of the $i$-$j$ collision event, i.e. "the event that agents $i$ and $j$ are colliding at time $t$". Assuming the HBR simplification holds (see Definition III.3), then, $\mathrm{KPC}_{i,j}$ can be expressed as

$$\mathrm{KPC}_{i,j}\,(t|t_0) = p\left(\left\|\underline{\mathbf{R}}_{i,j}\,(t|t_0)\right\| < l_{i,j}\middle|t_0\right) = p\left(\underline{\mathbf{R}}_{i,j}\,(t|t_0) \in V_{i,j}\middle|t_0\right) \tag{III.10}$$

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

$$\mathrm{KPC}_{i,j}\,(t|t_0) = \int_{\underline{\mathbf{r}}\in V_{i,j}} \mathrm{pdf}_{\underline{\mathbf{R}}_{i,j}|t,t_0}\left(\underline{\mathbf{r}}\middle|t,t_0\right) d\underline{\mathbf{r}} \qquad \diamond \tag{III.11}$$

Definition III.8 is illustrated with an example where the (one-dimensional) relative position is normally distributed, as shown in Figure 2.

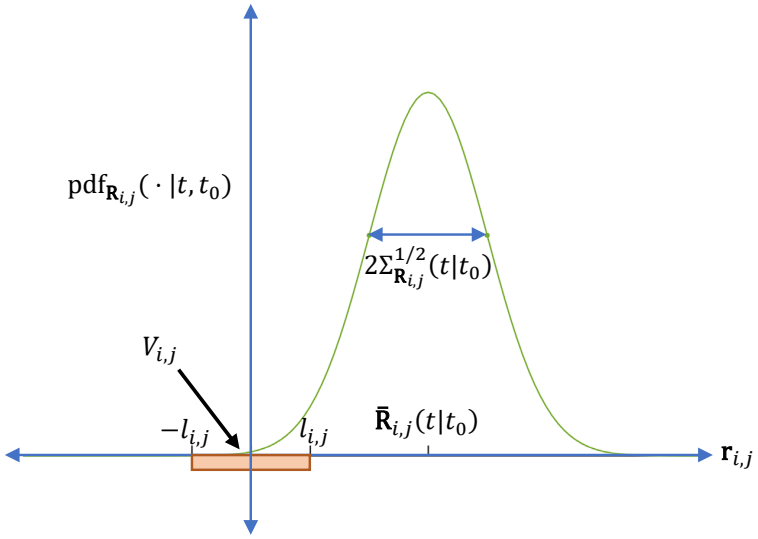


**Fig. 2 Conceptual KPC computation through integration of relative position pdf for a system with normally distributed (one-dimensional) relative position.**

Even though the $i$-$j$ collision event is defined as a condition in the relative position states $\underline{\mathbf{R}}_{i,j}$ at time $t$, these relative position states (and their uncertainty descriptions) are, in general, dynamically coupled with other states. Remark III.9 implies that, if the pdf of the complete relative state $\underline{\mathbf{X}}_{i,j}$ is known, $\mathrm{KPC}_{i,j}\,(t|t_0)$ can be computed through marginalization. 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 $\underline{\mathbf{U}}_{i,j}$ is the relative velocity $\dot{\underline{\mathbf{R}}}_{i,j}$.

**Remark III.9.** Assume, without loss of generality, that the relative state $\underline{\mathbf{X}}_{i,j}$ is partitioned as $\underline{\mathbf{X}}^T_{i,j} = \left[\underline{\mathbf{R}}^T_{i,j}\,,\,\underline{\mathbf{U}}^T_{i,j}\right]$ (see Subsection III.A), and that the pdf of the relative state $(\mathrm{pdf}_{\underline{\mathbf{X}}_{i,j}|t,t_0})$ is known. Then, the pdf of the relative position $(\mathrm{pdf}_{\underline{\mathbf{R}}_{i,j}|t,t_0})$ can be found as the marginal pdf of $\underline{\mathbf{X}}_{i,j}$ integrated over the "other relative states" $\underline{\mathbf{U}}_{i,j}$ as given by

$$\mathrm{pdf}_{\underline{\mathbf{R}}_{i,j}|t,t_0}\left(\underline{\mathbf{r}}\middle|t,t_0\right) = \int_{\underline{\mathbf{u}}\in\mathbb{R}^{n_{\underline{\mathbf{X}}}-d_{\underline{\mathbf{R}}}}} \mathrm{pdf}_{\underline{\mathbf{X}}_{i,j}|t,t_0}\left(\begin{bmatrix}\underline{\mathbf{r}}\\ \underline{\mathbf{u}}\end{bmatrix}\middle|t,t_0\right) d\underline{\mathbf{u}} \qquad \diamond \tag{III.12}$$

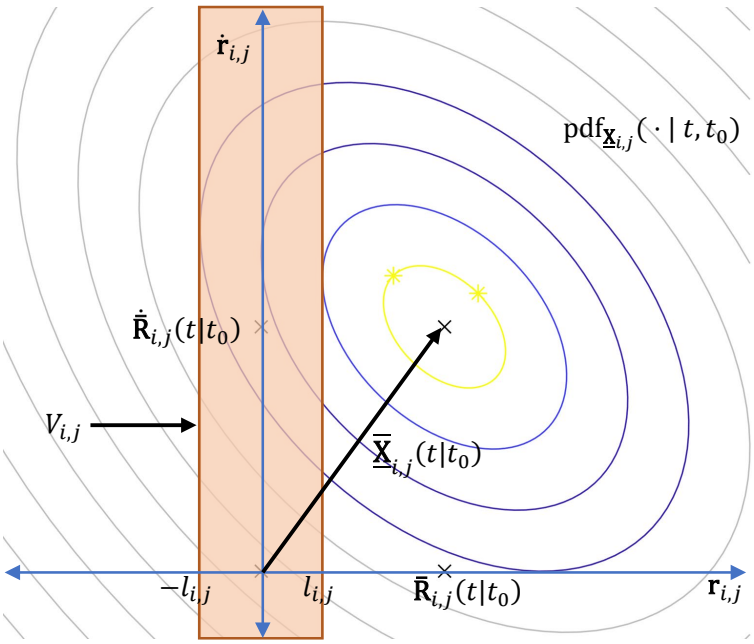


**Fig. 3 Conceptual KPC computation through integration of relative state pdf for a system with one-dimensional relative position and normally distributed (two-dimensional) relative state.**

It is helpful to note that these other relative states $\underline{\mathbf{U}}_{i,j}$ 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 Clohessy-Wiltshire (CW) relative orbital dynamics, $\underline{\mathbf{U}}_{i,j}$ would be the relative position rate state, which is different from a relative velocity state because the CW frame (or Hill frame) is not inertial.[30, 31]

## IV. Theory

In this section, integrals of multi-dimensional, normally distributed random variables bounded by certain hypersurfaces are linked to the chi-square distribution. The relationship between these 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 when they enter the collision region; finally, collision probabilities 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 $\underline{X} \sim \mathcal{N}\left(\bar{\underline{X}}, \Sigma\right)$, where $\bar{\underline{X}} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n\times n}$, $\Sigma > 0$. Let $\underline{x} \in \mathbb{R}^n$ be an instance of $\underline{X}$. Then, the mapping $D^2_{\underline{X}} \colon \mathbb{R}^n \to [0, \infty)$ transforms instances $\underline{x}$ of the random variable $\underline{X}$ by the rule

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

Thus, $D_{\underline{X}}\left(\underline{x}\right) \doteq \sqrt{D^2_{\underline{X}}\left(\underline{x}\right)}$ is the Mahalanobis distance of the point $\underline{x}$.[32]

Note: use of the Mahalanobis distance in this work is restricted to continuous, normal distributions. ◊

Now, the notions of the $d$-Mahalanobis contour and volume and of the $d_1, d_2$-Mahalanobis shell are introduced.

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

$$V_d\left(\underline{X}\right) \doteq \left\{\underline{x} \in \mathbb{R}^n \colon D^2_{\underline{X}}\left(\underline{x}\right) \leq d^2\right\} \tag{IV.2}$$

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

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

Finally, the $d_1, d_2$-Mahalanobis shell of $\underline{X}$, denoted by $V^{d_2}_{d_1}\left(\underline{X}\right)$, is defined as the set

$$V^{d_2}_{d_1}\left(\underline{X}\right) \doteq \left\{\underline{x} \in \mathbb{R}^n : 0 \le d_1 \le D_{\underline{X}}\left(\underline{x}\right) \le d_2\right\} \tag{IV.4}$$

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

Figure 4 illustrates the notions of Mahalanobis contour and volume for a non-degenerate, normally distributed (finite-dimensional) random variable $\underline{X}$. The $d$-Mahalanobis volumes $V_d\left(\underline{X}\right)$ are hypervolumes (specifically, hyperellipsoids) while the $d$-Mahalanobis contours $L_d\left(\underline{X}\right)$ are hypersurfaces (specifically, hyperellipses) in $n$-dimensions.

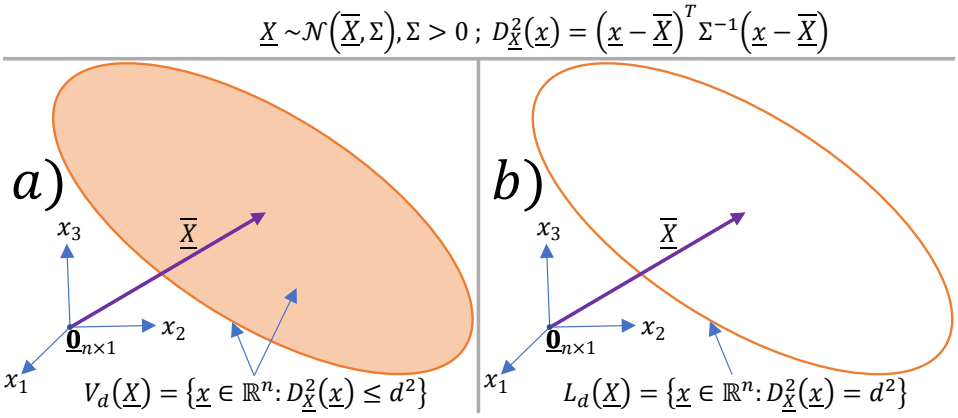


**Fig. 4 $d$-Mahalanobis volume and contour, a) $V_d$ and b) $L_d$, respectively.**

Lemma IV.3 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 Theorem 12.3.2).[32]

**Lemma IV.3** (Cumulative distributions bounded by contours of constant Mahalanobis distance)**.** Let $\underline{X} \sim \mathcal{N}\left(\bar{\underline{X}}, \Sigma\right)$, where $\bar{\underline{X}} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n\times n}$, $\Sigma > 0$. Then,

$$p\left(\left[\underline{X} - \bar{\underline{X}}\right]^T \Sigma^{-1}\left[\underline{X} - \bar{\underline{X}}\right] \le d^2\right) = \mathrm{cdf}_{\chi^2_n}\left(d^2\right) \tag{IV.5}$$

Let the function $F_{\underline{X}} : \mathbb{R} \to [0, 1]$ be defined by the rule

$$F_{\underline{X}}\left(d\right) \doteq \begin{cases} p\left(D^2_{\underline{X}}\left(\underline{X}\right) \le d^2\right) & \text{if } d \ge 0 \\ 0 & \text{if } d < 0 \end{cases} \tag{IV.6}$$

Then, $F_{\underline{X}}$ is a cumulative distribution function for $\underline{X}$. ◊

It is trivial that $F_{\underline{X}}(d) = 0$ for every $d < 0$, so only $d \ge 0$ are considered. Figure 5 illustrates the "alternative" cumulative distribution of a normally distributed random variable $\underline{X}$, $F_{\underline{X}}$, for representative dimensionalities of $\underline{X}$.

**Corollary IV.4** (Probability mass within Mahalanobis volume and shell)**.** Let $\underline{X} \sim \mathcal{N}\left(\bar{\underline{X}}, \Sigma\right)$, where $\bar{\underline{X}} \in \mathbb{R}^n$, and $\Sigma \in \mathbb{R}^{n\times n}, \Sigma > 0$. Let the $d$-Mahalanobis volume of $\underline{X}$, $V_d\left(\underline{X}\right)$, be as defined in Notation IV.2. Then,

$$p\left(\underline{X} \in V_d\left(\underline{X}\right)\right) = \mathrm{cdf}_{\chi^2_n}\left(d^2\right) \tag{IV.7}$$

Let the $d_1, d_2$-Mahalanobis shell of $\underline{X}$, $V^{d_2}_{d_1}\left(\underline{X}\right)$, be as defined in Notation IV.2. Then,

$$p\left(\underline{X} \in V^{d_2}_{d_1}\left(\underline{X}\right)\right) = \mathrm{cdf}_{\chi^2_n}\left(d_2^2\right) - \mathrm{cdf}_{\chi^2_n}\left(d_1^2\right) \quad ◊ \tag{IV.8}$$

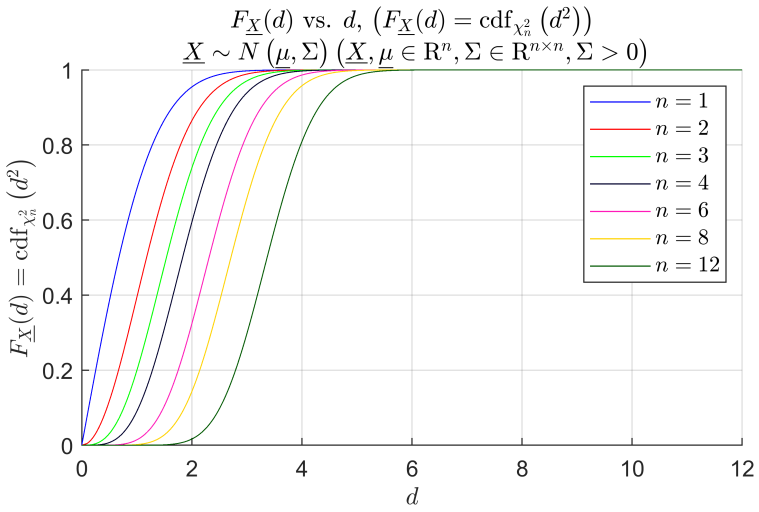


**Fig. 5 "Alternative", Mahalanobis distance-based cdf of normal random variables (and its connection to the chi-square cdf), for representative dimensionalities.**

The implications of Lemma IV.3 and Corollary IV.4 on nondegenerate, normal random variables are illustrated in Figure 6. 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 statistics and dimension of the random variable.

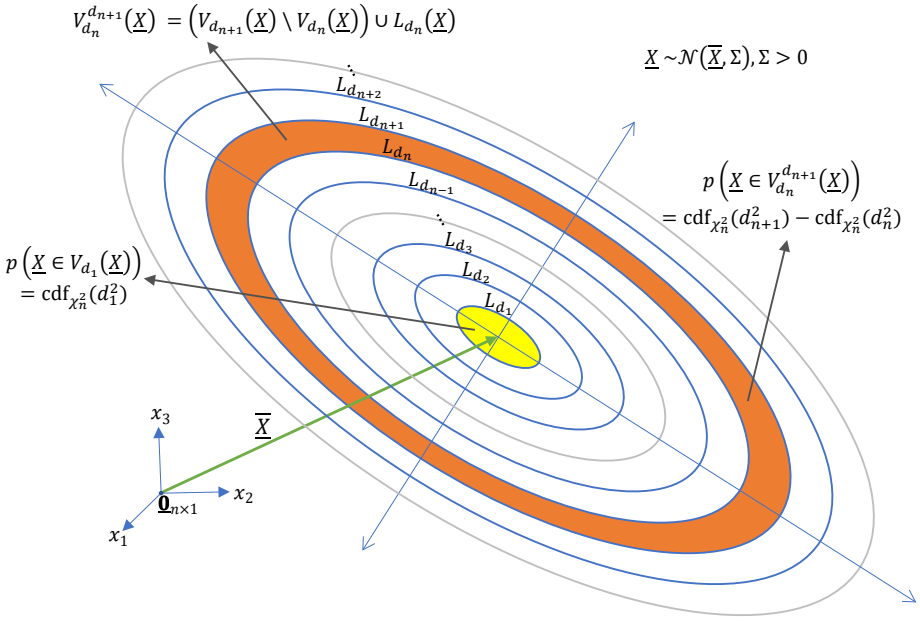


**Fig. 6 Probability measures of normally distributed random variables over $n$-hypervolumes bounded by $n$-hypersurfaces of constant $d$-Mahalanobis distance).**

### 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" (MSS) algorithm, and it enables sampling from multidimensional, non-degenerate normal distributions.

The MSS algorithm aims to do the following. First, a sample is made along high-dimensional unit sphere surfaces, and such unit hypersphere samples are uniformly distributed in a geometric sense. Second, the result from Corollary

**Input:** $N_{\text{shells}}$, $N_{\text{samples/shell}}$, $d_{\max}$ (max. sample Mahalanobis distance), dimension of elements of the sequence $n_{\underline{\mathbf{X}}}$ ($n_{\underline{\mathbf{X}}} \in \mathbb{N}, n_{\underline{\mathbf{X}}} \geq 2$)

**Output:** MSS sample $\left\{\underline{\tilde{\mathbf{x}}}_k(t_0)\right\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$, sample weights $\{w_k\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$

**Data:** Initial conditions $\underline{\bar{\mathbf{X}}}_{i,j}(t_0)$, $\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0)$; prior state distribution $\underline{\mathbf{X}}_{i,j}(t_0) \sim \mathcal{N}\left(\underline{\bar{\mathbf{X}}}_{i,j}(t_0), \Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0)\right)$

$N_{\text{samples}} \leftarrow N_{\text{shells}} N_{\text{samples/shell}}$

$\{\underline{\mathbf{z}}_k\}_{k\in\{1,\ldots,N_{\text{samples}}\}} \leftarrow$ output of some unit $(n_{\underline{\mathbf{X}}}-1)$-sphere sampling algorithm $\left(N_{\text{sequence}} = N_{\text{samples}}, D = n_{\underline{\mathbf{X}}}\right)$
// see Appendix for a list of algorithms

$\delta_d \leftarrow d_{\max}/N_{\text{shells}}$

$\mathbf{S}_1, \mathbf{Q}, \mathbf{S}_2 \in \mathbb{R}^{n_{\underline{\mathbf{X}}}\times n_{\underline{\mathbf{X}}}} \leftarrow$ such that $\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0) = \mathbf{S}_1\mathbf{Q}\mathbf{S}_2^T$ // output of SVD

$\Sigma^{1/2}_{\underline{\mathbf{X}}_{i,j}}(t_0) \leftarrow \mathbf{S}_1\mathbf{Q}^{1/2}\mathbf{S}_2^T$

$k \leftarrow 0$

**for** $l \leftarrow 1$ ***to*** $N_{\text{shells}}$ **do**

  $d_l \leftarrow (l-1)\delta_d$

  $W_l \leftarrow \text{cdf}_{\chi^2_D}\left((d_l+\delta_d)^2\right) - \text{cdf}_{\chi^2_D}\left(d_l^2\right)$ // probability mass in $l^{th}$ ($n_{\underline{\mathbf{X}}}$-dimensional) Mahalanobis shell

  **for** $p \leftarrow 1$ ***to*** $N_{\text{samples/shell}}$ **do**

    $k \leftarrow k+1$

    $u_k \sim U[0,1]$

    $\underline{\tilde{\mathbf{x}}}_k(t_0) \leftarrow \underline{\bar{\mathbf{X}}}_{i,j}(t_0) + (d_l + u_k\delta_d)\left[\Sigma^{1/2}_{\underline{\mathbf{X}}_{i,j}}(t_0)\right]\underline{\mathbf{z}}_k$ // 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** $\left\{\underline{\tilde{\mathbf{x}}}_k(t_0)\right\}_{k\in\{1,\ldots,N_{\text{samples}}\}}, \{w_k\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$

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

IV.4 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 ($N_{\text{shells}}$). Then, for every Mahalanobis shell in the sample, points on the unit hypersphere are transformed to points in their respective Mahalanobis shells. 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, through direct application of Corollary IV.4, the MSS algorithm generates samples of normal distributions by transforming unit hypersphere points into points in Mahalanobis shells. The MSS sampling method could in principle be applied to sample from arbitrary (non-degenerate) normal pdfs, regardless of their physical interpretation or application. However, in accordance with the motivation of this work, a method is presented which applies MSS sampling to the computation of collision probability between any two agents in proximity.

The MSS KPC and TPc/WPC computation method, listed as Algorithm 2, utilizes MSS samples for the computation of kinematic probability of collision (KPC) and total/window probability of collision (TPc/WPC) waveforms. First, an MSS sample is generated which 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, numerical integration is sidestepped by using weights that reflect integration over regions bounded by certain surfaces, and the difficulties of Monte Carlo methods are avoided by directly sampling on probability distribution tails.

Figure 7 illustrates an application of the MSS method to compute KPC for a normally distributed relative dynamic system in $\mathbb{R}^2$ (with 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.

**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 $\bar{\underline{\mathbf{X}}}_{i,j}(t_0)$, $\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0)$; prior state distribution $\underline{\mathbf{X}}_{i,j}(t_0) \sim \mathcal{N}\left(\bar{\underline{\mathbf{X}}}_{i,j}(t_0), \Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0)\right)$; MSS sample $\left\{\tilde{\underline{\mathbf{x}}}_k(t_0)\right\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$, sample weights $\{w_k\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$ ; matrix $\mathbf{M}$ for mapping from complete relative state to relative position (see subsection III.A)

$\delta_t \leftarrow \frac{t_f - t_0}{N_t - 1}$

$\underline{\mathbf{a}} \leftarrow \underline{\mathbf{0}}_{N_{\text{samples}}\times 1}$

$B_2 \leftarrow 0$

**for** $m \leftarrow 1$ ***to*** $N_t$ **do**

  $t_m \leftarrow t_0 + \frac{t_f - t_0}{N_t - 1}(m-1)$

  $\bar{\underline{\mathbf{X}}}_{i,j}(t_m) \leftarrow \Phi_{i,j}(t_m, t_0)\, \bar{\underline{\mathbf{X}}}_{i,j}(t_0)$

  $\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_m) \leftarrow \Phi_{i,j}(t_m, t_0)\, \Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0) \Phi_{i,j}^T(t_m, t_0)$

  $\underline{\mathbf{R}}_{i,j}(t_m|t_0) \sim \mathcal{N}\left(\mathbf{M}\bar{\underline{\mathbf{X}}}_{i,j}(t_m), \mathbf{M}\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_m)\mathbf{M}^T\right)$ // M as defined in Subsection III.A

  $\mathrm{KPC}_{i,j}(t_m|t_0) \leftarrow \int_{\underline{\mathbf{r}}\in V_{i,j}} \mathrm{pdf}_{\underline{\mathbf{R}}_{i,j}|t,t_0}\left(\underline{\mathbf{r}}\middle|t_m, t_0\right) d\underline{\mathbf{r}}$ // KPC truth

  $B_1 \leftarrow 0$

  **for** $k \leftarrow 1$ ***to*** $N_{\text{samples}}$ **do**

    **if** $m > 1$ **then**

      $\tilde{\underline{\mathbf{x}}}_k \leftarrow \Phi_{i,j}(t_m, t_m - \delta_t)\, \tilde{\underline{\mathbf{x}}}_k$

    $\tilde{\underline{\mathbf{r}}}_k \leftarrow \mathbf{M}\tilde{\underline{\mathbf{x}}}_k$

    **if** $\|\tilde{\underline{\mathbf{r}}}_k\| \le l_{i,j}$ **then** // $i$-$j$ collision event at time $t_m$ ($k^{th}$ particle)

      $B_1 \leftarrow B_1 + w_k$

      **if** $\left[\underline{\mathbf{a}}\right]_k = 0$ **then** // $i$-$j$ collision ($k^{th}$ particle) had not yet occurred

        $\left[\underline{\mathbf{a}}\right]_k \leftarrow 1$ // mark $i$-$j$ collision as having occurred ($k^{th}$ particle)

        $B_2 \leftarrow B_2 + w_k$

  $\widetilde{\mathrm{KPC}}_{i,j}(t_m|t_0) \leftarrow B_1$

  $\widetilde{\mathrm{WPC}}_{i,j}(t_0, t_m) \leftarrow B_2$

$\mathcal{T} \leftarrow \{t_m\}_{m\in\{1,\ldots,N_t\}}$

$\text{KPC error RMS} \leftarrow \sqrt{\frac{1}{N_t}\sum_{m=1}^{N_t}\left(\widetilde{\mathrm{KPC}}_{i,j}(t_m|t_0) - \mathrm{KPC}_{i,j}(t_m|t_0)\right)^2}$

**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.

## V. Results and discussion

In this section, the Mahalanobis Shell Sampling (MSS) algorithm (see Algorithm 1) is applied in the context of a dynamic example to kinematic probability of collision (KPC) computation (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 the KPC, which is presented in subsection V.A. Finally, the efficacy of the MSS application to KPC computation is examined through 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.B.

A variant of the general $n_{\underline{\mathbf{X}}}$-dimensional MSS sampling algorithm (see Algorithm 1) for the case of a relative dynamic state in $\mathbb{R}^2$ (relative position in $\mathbb{R}$) is presented as Algorithm 3 and implemented for the examples of this section.

### 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.

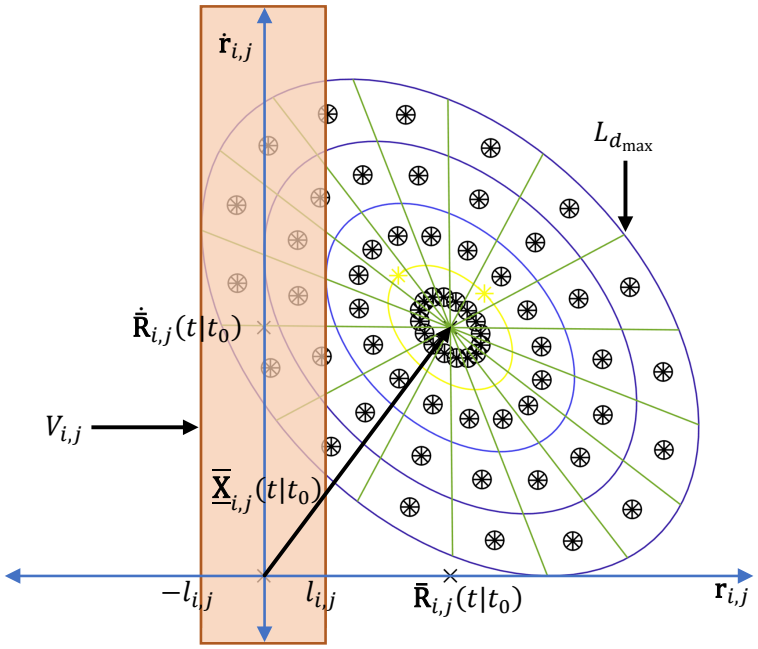


**Fig. 7 Conceptual application of MSS to KPC computation for a system with one-dimensional relative position and normally distributed (two-dimensional) relative state.**

The state of agent $i$, $\underline{\mathbf{X}}_i \in \mathbb{R}^2$, is defined in terms of the position of its center of mass $R_i \in \mathbb{R}$ (see subsection III.A) and its corresponding velocity $\dot{R}_i \in \mathbb{R}$ as $\underline{\mathbf{X}}_i^T \doteq \begin{bmatrix} R_i & \dot{R}_i \end{bmatrix}$. Furthermore, it is assumed that the center of mass of box $j$ is stationary at the origin initially, while $\underline{\mathbf{X}}_i(t_0)$ is nontrivial. Therefore, $\underline{\mathbf{X}}_j(t) = \mathbf{0}_{2\times 1}$ for any time $t$, and the dynamics of $\underline{\mathbf{X}}_{i,j}$ are the same as the dynamics of $\underline{\mathbf{X}}_i$. Denoting the $i^{th}$ mass, damping coefficient and spring constant by $m_i$, $b_i$ and $k_i$, respectively, the dynamics of $\underline{\mathbf{X}}_{i,j}$ can be expressed as follows:

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

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, $k_i > b_i^2/4m_i$. Since the present system exhibits linear, time-invariant (LTI) dynamics, the mean relative state $\bar{\underline{\mathbf{X}}}_{i,j}$ can be propagated as given by

$$\bar{\underline{\mathbf{X}}}_{i,j}(t) = \Phi_{i,j}(t, t_0)\,\bar{\underline{\mathbf{X}}}_{i,j}(t_0) = \Phi_{i,j}(t - t_0, 0)\,\bar{\underline{\mathbf{X}}}_{i,j}(t_0) \tag{V.2}$$

where the relative state STM, $\Phi_{i,j}$, is given by

$$\Phi_{i,j}(t,0) = \exp(-\zeta\omega_n t)\begin{bmatrix} \cos(\omega_d t) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d t) & \frac{1}{\omega_n\sqrt{1-\zeta^2}}\sin(\omega_d t) \\ \frac{\omega_n}{\sqrt{1-\zeta^2}}\sin(\omega_d t) & \cos(\omega_d t) - \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d t) \end{bmatrix} =: \begin{bmatrix} \phi_{11}(t) & \phi_{12}(t) \\ \phi_{21}(t) & \phi_{22}(t) \end{bmatrix} \tag{V.3}$$

where $\omega_n \doteq \sqrt{k_i/m_i}$, $\zeta \doteq b_i/\sqrt{4k_i m_i}$, and

$$\omega_d \doteq \omega_n\sqrt{1-\zeta^2} = \sqrt{k_i/m_i - 0.25\,(b_i/m_i)^2} \tag{V.4}$$

Let the relative state covariance $\Sigma_{\underline{\mathbf{X}}_{i,j}} > 0$ 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:

$$\Sigma_{\underline{\mathbf{X}}_{i,j}} = \begin{bmatrix} [\Sigma_{\underline{\mathbf{X}}_{i,j}}]_{1,1} & [\Sigma_{\underline{\mathbf{X}}_{i,j}}]_{1,2} \\ [\Sigma_{\underline{\mathbf{X}}_{i,j}}]_{1,2} & [\Sigma_{\underline{\mathbf{X}}_{i,j}}]_{2,2} \end{bmatrix} =: \begin{bmatrix} \sigma_{R_{i,j}}^2 & \rho_{(R_{i,j},\dot{R}_{i,j})}\sigma_{R_{i,j}}\sigma_{\dot{R}_{i,j}} \\ \rho_{(R_{i,j},\dot{R}_{i,j})}\sigma_{R_{i,j}}\sigma_{\dot{R}_{i,j}} & \sigma_{\dot{R}_{i,j}}^2 \end{bmatrix} \tag{V.5}$$

**Input:** $N_{\text{shells}}$, $N_{\text{samples/shell}}$, $d_{\max}$ (max. sample Mahalanobis distance)
**Output:** MSS sample $\left\{\tilde{\underline{\mathbf{x}}}_k(t_0)\right\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$, sample weights $\{w_k\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$
**Data:** Initial conditions $\bar{\underline{\mathbf{X}}}_{i,j}(t_0), \Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0)$; prior state distribution $\underline{\mathbf{X}}_{i,j}(t_0) \sim \mathcal{N}\left(\bar{\underline{\mathbf{X}}}_{i,j}(t_0), \Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0)\right)$

$N_{\text{samples}} \leftarrow N_{\text{shells}} N_{\text{samples/shell}}$
$\delta_d \leftarrow d_{\max}/N_{\text{shells}}$
$\mathbf{S}_1, \mathbf{Q}, \mathbf{S}_2 \in \mathbb{R}^{2\times 2} \leftarrow$ such that $\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0) = \mathbf{S}_1\mathbf{Q}\mathbf{S}_2^T$ // output of SVD
$\Sigma^{1/2}_{\underline{\mathbf{X}}_{i,j}}(t_0) \leftarrow \mathbf{S}_1\mathbf{Q}^{1/2}\mathbf{S}_2^T$
$k \leftarrow 0$
**for** $l \leftarrow 1$ ***to*** $N_{\text{shells}}$ **do**
  $d_l \leftarrow (l - \frac{1}{2})\delta_d$
  $W_l \leftarrow \text{cdf}_{\chi^2_2}\left((d_l + \frac{1}{2}\delta_d)^2\right) - \text{cdf}_{\chi^2_2}\left((d_l - \frac{1}{2}\delta_d)^2\right)$ // probability mass in $l^{th}$ Mahalanobis shell (in $\mathbb{R}^2$)
  $r_l \sim U[0,1]$
  $\theta_l \leftarrow 2\pi r_l / N_{\text{samples/shell}}$
  **for** $p \leftarrow 1$ ***to*** $N_{\text{samples/shell}}$ **do**
    $k \leftarrow k + 1$
    $\theta_k = \theta_l + 2\pi(p-1)/N_{\text{samples/shell}}$
    $\underline{\mathbf{z}}_k \leftarrow \left[\cos(\theta_k) \quad \sin(\theta_k)\right]^T$ // point in 1-sphere (i.e. circle)
    $\tilde{\underline{\mathbf{x}}}_k(t_0) = \bar{\underline{\mathbf{X}}}_{i,j}(t_0) + d_l\left[\Sigma^{1/2}_{\underline{\mathbf{X}}_{i,j}}(t_0)\right]\underline{\mathbf{z}}_k$ // 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** $\left\{\tilde{\underline{\mathbf{x}}}_k(t_0)\right\}_{k\in\{1,\ldots,N_{\text{samples}}\}}, \{w_k\}_{k\in\{1,\ldots,N_{\text{samples}}\}}$

**Algorithm 3:** MSS sample generation, relative state in $\mathbb{R}^2$ (relative position in $\mathbb{R}$), normal prior distribution.

Suppose $\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0) > 0$ is known. Additionally, suppose the initial relative state $\underline{\mathbf{X}}_{i,j}(t_0)$ is normally distributed. Because the dynamics of the relative state $\underline{\mathbf{X}}_{i,j}$ are linear and uncontrolled, its distribution remains normal, i.e.

$$\underline{\mathbf{X}}_{i,j}(t) \sim \mathcal{N}\left(\Phi_{i,j}(t,t_0)\,\bar{\underline{\mathbf{X}}}_{i,j}(t_0),\, \Phi_{i,j}(t,t_0)\,\Sigma_{\underline{\mathbf{X}}_{i,j}}(t_0)\,\Phi^T_{i,j}(t,t_0)\right) \;\forall t \geq t_0 \tag{V.6}$$

Based on Definition III.8, the KPC between agents $i$ and $j$, $\text{KPC}_{i,j}(t|t_0)$, can be computed as

$$\text{KPC}_{i,j}(t|t_0) = \int_{-l_{i,j}}^{l_{i,j}} \text{pdf}_{\underline{\mathbf{R}}_{i,j}|t,t_0}\left(\underline{\mathbf{r}}\middle|t,t_0\right) d\underline{\mathbf{r}} \tag{V.7}$$

For this example, the $\text{KPC}_{i,j}(\cdot|t_0)$ waveform can be found analytically as

$$\text{KPC}_{i,j}(t|t_0) = \frac{1}{2}\left[\text{erf}\left(\frac{l_{i,j} - \bar{R}_{ij}(t)}{\sigma_{R_{i,j}}(t)\sqrt{2}}\right) - \text{erf}\left(\frac{-l_{i,j} - \bar{R}_{ij}(t)}{\sigma_{R_{i,j}}(t)\sqrt{2}}\right)\right] \tag{V.8}$$

where $\text{erf}(\cdot)$ is the error function,[33] where $\Delta t \doteq t - t_0$, and where

$$\bar{R}_{ij}(t) = \phi_{11}(\Delta t)\,\bar{R}_{i,j}(t_0) + \phi_{12}(\Delta t)\,\dot{\bar{R}}_{i,j}(t_0) \tag{V.9}$$

$$\sigma^2_{R_{i,j}}(t) = \phi^2_{11}(\Delta t)\,\sigma^2_{R_{i,j}}(t_0) + \phi^2_{12}(\Delta t)\,\sigma^2_{\dot{R}_{i,j}}(t_0) + 2\phi_{11}(\Delta t)\,\phi_{12}(\Delta t)\,\rho_{(R_{i,j},\dot{R}_{i,j})}(t_0)\,\sigma_{R_{i,j}}(t_0)\,\sigma_{\dot{R}_{i,j}}(t_0) \tag{V.10}$$

### B. Relative mass-spring-damper system - KPC and WPC through MSS application - 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

**Table 1 Simulation parameters for 1D mass-spring-damper system examples.**

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

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.

The first case, Example #1D.001, is discussed in subsubsection V.B.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.B.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. Finally, because the MSS, Monte Carlo and analytical KPC waveforms are in agreement, only MSS KPC waveforms are shown for brevity. In all examples, it is assumed that $N_{\text{samples/shell}} = 120$. (MSS KPC waveforms are generated with $N_{\text{shells}} = 141$, and $d_{\max} = 7.05$).

*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 Figure 8) 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.

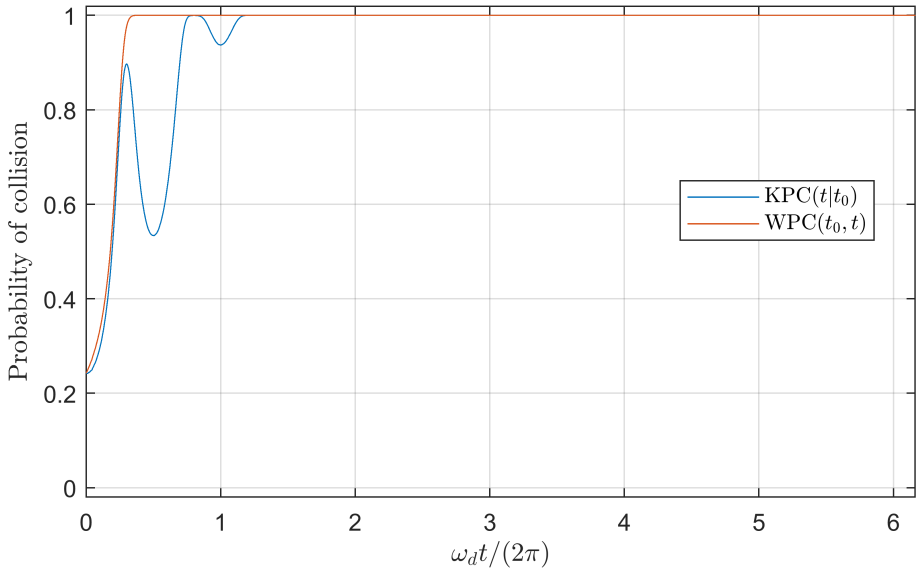


**Fig. 8 Probability of collision waveform results, MSS-generated, Example #1D.001** ($N_{\text{shells}} = 141$, $d_{\max} = 7.05$)**.**

Interestingly, the "Total Probability of collision" (TPc, see Frigm et al.,[20] also labeled as WPC) waveforms generated through Monte Carlo sampling of the initial state distribution and through MSS are also consistent, and they show that (almost) every region of the initial relative state pdf has crossed the collision region $V_{i,j}$ (see Figure 3 for reference) by the 1/3-oscillation, which implies that a collision has "almost surely" occurred by this time. This TPc/WPC behavior does not contradict KPC behavior; in fact, their insights are 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% 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 noted previously, a sampling-based method is required for TPc/WPC computation; therefore, it is helpful to compare such methods. Through MSS, samples were made of points within $d_{\text{max}} = 7.05$ (colloquially, to within 7.05-"$\sigma$") utilizing only 16920 points. The probability mass outside $d_{\text{max}} = 7.05$ 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-"$\sigma$" 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-"$\sigma$" 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-"$\sigma$").

Thus, the MSS sample method accounts for probabilistic outliers without unduly penalizing sample size. Thus, the MSS algorithm 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 $N_{\text{shells}}$, the number of samples per shell $N_{\text{samples/shell}}$, and the cutoff Mahalanobis distance $d_{\text{max}}$. Figures 9, 10 and 11 show how accurately the MSS KPC waveform approximates the analytical KPC waveform, specifically, when changing $N_{\text{shells}}$ and $d_{\text{max}}$ (with fixed $N_{\text{samples/shell}}$). (The results of Figures 9, 10 and 11 are averaged over 16 simulations.)

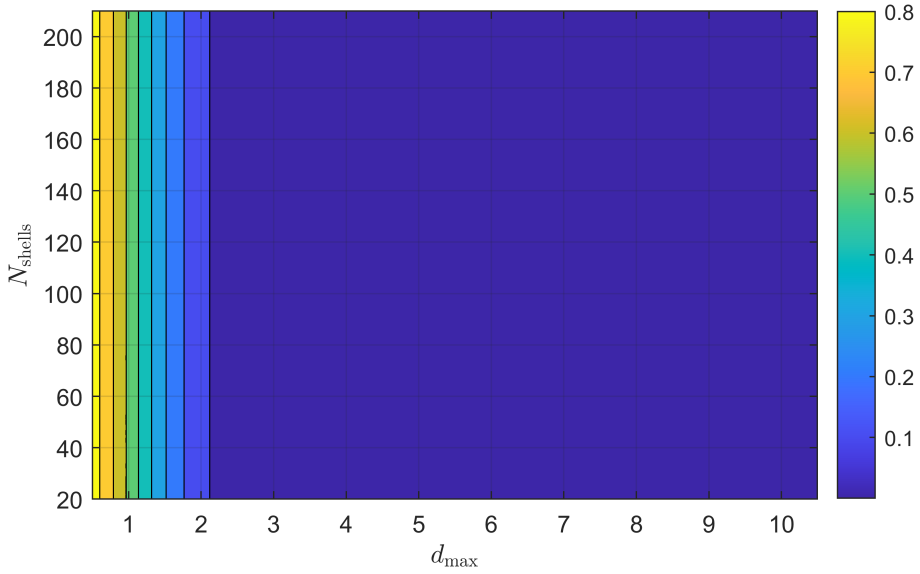


**Fig. 9 Error RMS between analytical and MSS KPC waveforms; changing $N_{\text{shells}}$ and $d_{\text{max}}$ (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.001 (after 16 simulations).**

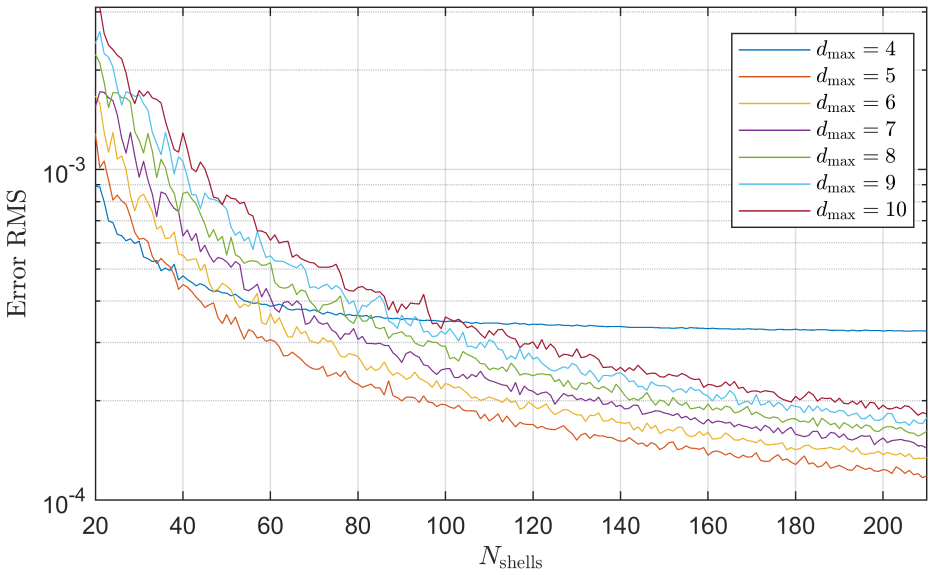


**Fig. 10 Error RMS between analytical and MSS KPC waveforms; changing $N_{\text{shells}}$ while holding $d_{\max}$ constant (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.001 (after 16 simulations).**

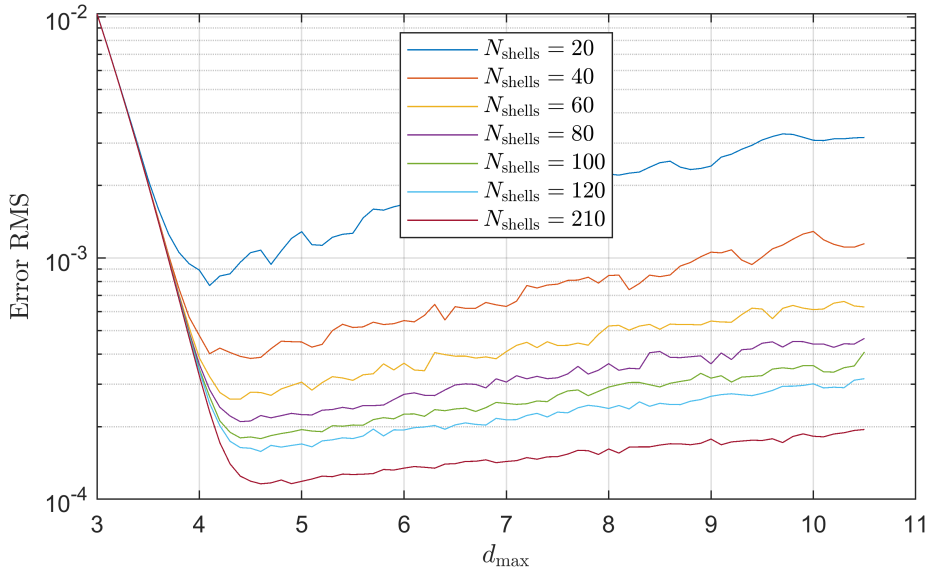


**Fig. 11 Error RMS between analytical and MSS KPC waveforms; changing $d_{\max}$ while holding $N_{\text{shells}}$ constant (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.001 (after 16 simulations).**

Upon examining the trade of changing $N_{\text{shells}}$ versus changing $d_{\max}$ (as seen in Figure 9), it is seen that the MSS KPC method is insensitive to $N_{\text{shells}}$, while increasing $d_{\max}$ results in a monotonic increase in error RMS accuracy. However, this observation only holds for "low" cutoff Mahalanobis distances $d_{\max}$. This behavior 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 equal to 1, but is equal to $\text{cdf}_{\chi^2_{(n_{\mathbf{x}})}}\left(d^2_{\max}\right)$, so a high enough $d_{\max}$ is needed for a valid sample.

Next, Figure 10 is examined, where representative $d_{\max} \in \{4, \ldots, 10\}$ are held constant, while changing $N_{\text{shells}}$. It is clear that, for constant $d_{\max}$, increasing $N_{\text{shells}}$ improves RMS residuals. Additionally, after $d_{\max} = 4$, better RMS residuals are obtained with lower $d_{\max}$. It can be seen that, after $d_{\max} = 4$, equal accuracy can be achieved with higher $d_{\max}$ by increasing $N_{\text{shells}}$; conversely, after $d_{\max} = 4$, increasing $d_{\max}$ while holding $N_{\text{shells}}$ constant decreases RMS accuracy. This trend is opposite to that observed in Figure 9, where for low $d_{\max}$ and while holding $N_{\text{shells}}$ constant,

increasing $d_{\max}$ increases RMS accuracy. Combined, these trends suggest that, when holding $N_{\text{shells}}$ constant, increasing $d_{\max}$ improves RMS accuracy up to a point, after which continuing to increase $d_{\max}$ worsens RMS accuracy.

In Figure 11, representative $N_{\text{shells}} \in [20, 210]$ are held constant, while changing $d_{\max}$. It is seen that, in every case, RMS residuals improve up to a certain "transition" cutoff Mahalanobis distance $d'_{\max}$ (which depends on $N_{\text{shells}}$), and the RMS residuals deteriorate with $d_{\max} > d'_{\max}$. Furthermore the, "transition" cutoff Mahalanobis distance $d'_{\max}$ increases with increased number of shells $N_{\text{shells}}$. 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 10.

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 $N_{\text{shells}}$. Thus, MSS KPC error RMS accuracy is not insensitive to the number of shells $N_{\text{shells}}$. In fact, for constant KPC error RMS, increasing $d_{\max}$ requires an increase in $N_{\text{shells}}$; conversely, for constant KPC error RMS, increasing $N_{\text{shells}}$ 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 $N_{\text{shells}}$ constant effectively creates a grid that, while including more probabilistic outliers, becomes increasingly more coarse. However, while correctly increasing $N_{\text{shells}}$ 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.B.1). Qualitatively, for Example #1D.002, the KPC waveforms generated analytically, through Monte Carlo sampling of the initial state distribution, and through MSS (shown in Figure 12) 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 another 1/4-oscillations. 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.

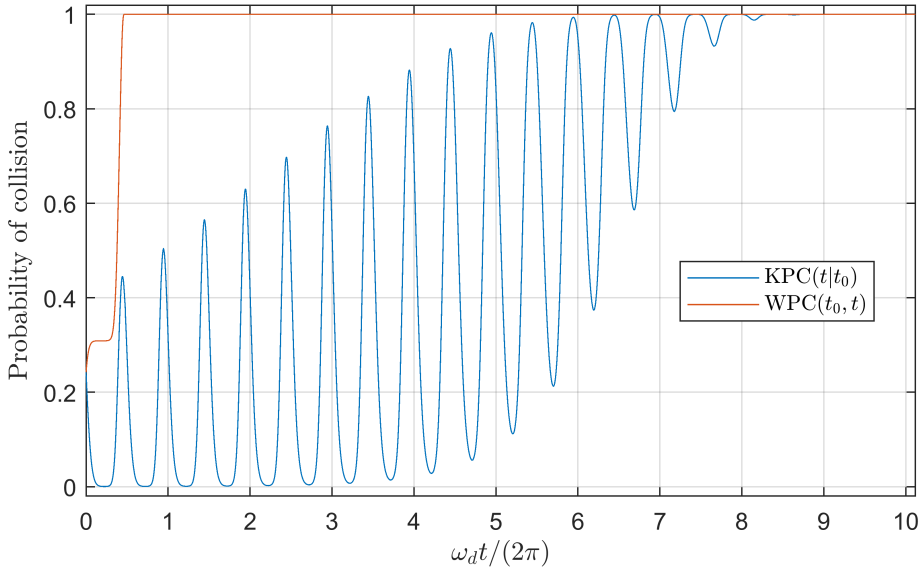


**Fig. 12 Probability of collision waveform results, MSS-generated, Example #1D.002** ($N_{\text{shells}} = 141$, $d_{\max} = 7.05$).

Example #1D.002 further illustrates how TPc/WPC is an important collision risk indicator. The TPc/WPC waveforms generated through Monte Carlo sampling of the initial state distribution and through MSS are also consistent, and they show that (almost) every region of the initial relative state pdf has crossed the collision volume $V_{i,j}$ 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. 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 2nd 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 $\text{cdf}_{\chi^2_2}\left(7.05^2\right) = 1 - 1.6115E - 11$. 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 $N_{\text{shells}}$ versus changing cutoff Mahalanobis distance $d_{\text{max}}$ (see Figure 13) , it is observed that, for "low" cutoff Mahalanobis distances $d_{\text{max}}$, the MSS KPC method is insensitive to the number of shells $N_{\text{shells}}$, and that increasing the cutoff Mahalanobis distance $d_{\text{max}}$ results in a monotonic increase in error RMS accuracy.

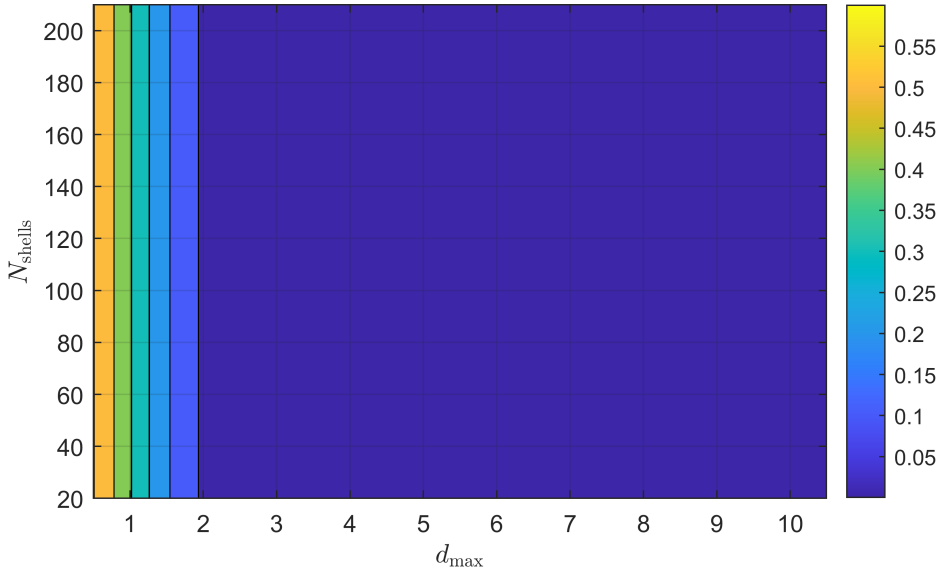


**Fig. 13 Error RMS between analytical and MSS KPC waveforms; changing $N_{\text{shells}}$ and $d_{\text{max}}$ (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.002 (after 1 simulation).**

For Example #1D.002, when examining the trade of changing $N_{\text{shells}}$ while holding constant $d_{\text{max}} \in \{4, \ldots, 10\}$ (see Figure 14), two observations can be made. First, for constant $d_{\text{max}}$, increasing $N_{\text{shells}}$ improves RMS residuals. Second, after $d_{\text{max}} = 4$, better RMS residuals are obtained with lower $d_{\text{max}}$. In the next trade (see Figure 15), for constant $N_{\text{shells}}$, increasing $d_{\text{max}}$ starts improving RMS residuals before stagnating at a "transition" cutoff Mahalanobis distance $d'_{\text{max}}$, after which continuing to increase $d_{\text{max}}$ degrades RMS residuals.

To examine the reproducibility of these trends, the same simulation is averaged over 23 simulations. Interestingly, 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 Algorithm 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 Algorithm 1) slightly in order to reflect this change.

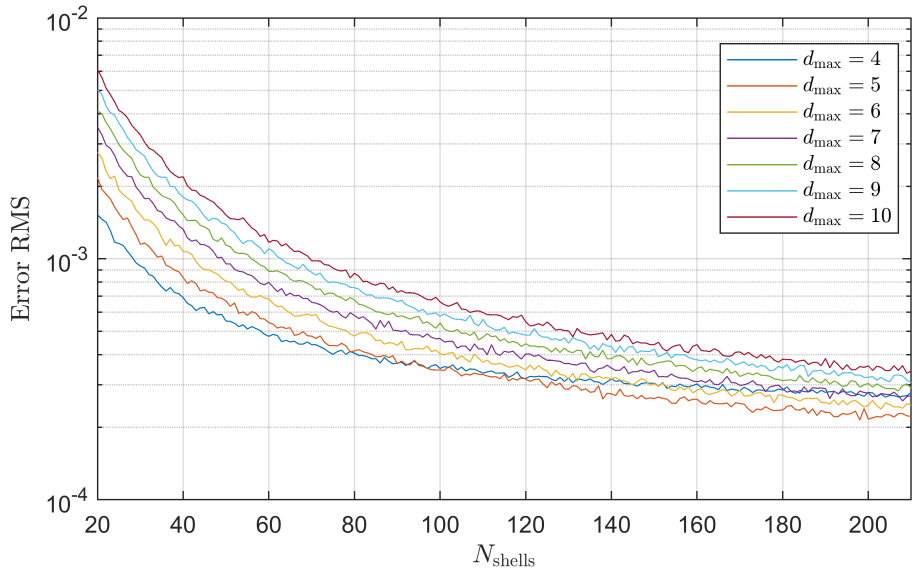


**Fig. 14 Error RMS between analytical and MSS KPC waveforms; changing $N_{\text{shells}}$ while holding $d_{\text{max}}$ constant (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.002 (after 1 simulation).**

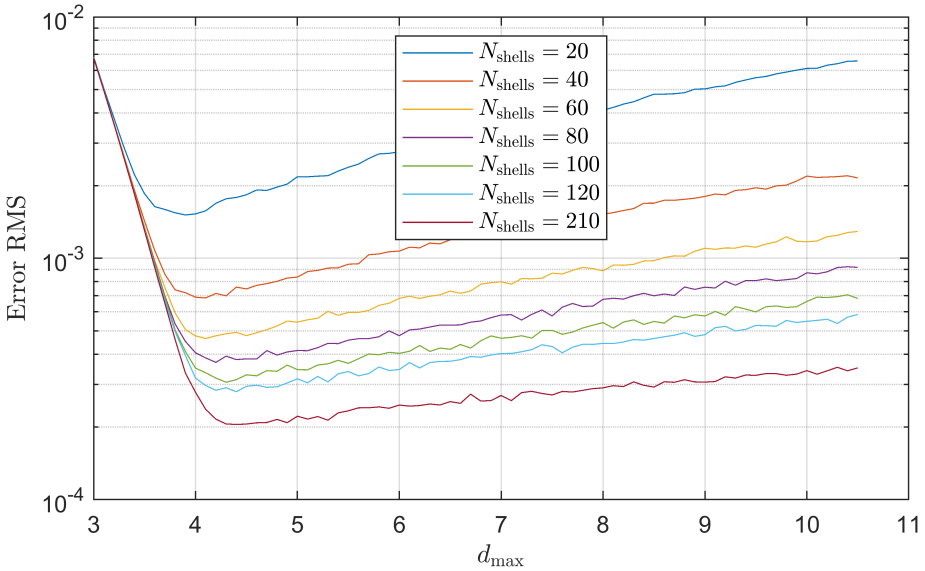


**Fig. 15 Error RMS between analytical and MSS KPC waveforms; changing $d_{\text{max}}$ while holding $N_{\text{shells}}$ constant (with fixed $N_{\text{samples/shell}}$) in MSS sample; Example #1D.002 (after 1 simulation).**

## 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 (MSS) algorithm, is established. Using the statistics of normal, nondegenerate distributions, the MSS algorithm transforms a sample from the unit hypersphere into a sample of the original 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 algorithm 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 collision risk analysis), which motivates the need for sampling from "tails" of probability distributions. 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 root-mean-square (RMS), or simply, error RMS) between the analytically and MSS generated waveforms.

The results obtained give insight into application of the MSS method to KPC computation. 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 sadmple mean, which implies having more extreme probabilistic outliers), without also increasing the number of shells $N_{\text{shells}}$, is ultimately detrimental to KPC accuracy. However, increasing $N_{\text{shells}}$ (without also increasing $d_{\max}$), always results in KPC accuracy improvements. Further examination of applications of MSS to KPC computation should quantify the required increment in $N_{\text{shells}}$ in order to obtain non-deteriorating KPC error RMS properties for incremental $d_{\max}$. Additionally, in future work, requirements on MSS sample parameters for 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 accuracy. In accordance with the motivation of this work, spacecraft formations should be among the dynamic systems explored in this effort.

Results were also examined for waveforms of the "Total Probability of collision" (TPc), also labeled as the window probability of collision (WPC). From the perspective of collision risk analysis, 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. This suggests that TPc/WPC information would be especially useful in two cases: 1) where the TPc/WPC indicates an intolerably high or imminent risk of collision while the KPC indicates little to no risk, and 2) where high risk is indicated much earlier by the TPc/WPC than by the KPC. 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 the 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 algorithm is a candidate for TPc/WPC computation. It is expected that the MSS algorithm would have better computational efficiency and predictive accuracy 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 KPC is defined in this work. Additionally, it would be appropriate to examine properties of the TPc/WPC waveform such as 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.

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.

The notation of the MSS sampling and KPC computation methods assume that the normal 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 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.

# Appendix

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

This subsection presents algorithms for uniform sampling on the unit sphere in high-dimensions. The Marsaglia (subsubsection VI.A.1) and the Roberts and Brauchart (subsubsection VI.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 Definition VI.1.

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

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

where $\tilde{S}(N)$ 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 $N_{\text{sequence}}$. This is a traditional method for ensuring that sequences produced by these algorithms are "evenly" distributed on the unit $(D-1)$-sphere.[34]

**Table 2 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 | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{\sqrt{N}}\right)$ | $O\left(\frac{1}{\sqrt{N}}\right)$ | $O\left(\frac{1}{\sqrt[3]{N}}\right)$ |
| Roberts | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{\sqrt{N}}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{\sqrt{N}}\right)$ | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{\sqrt{N}}\right)$ |
| Brauchart | $O\left(\frac{1}{N}\right)$ | $O\left(\frac{1}{\sqrt{N}}\right)$ | $O\left(\frac{1}{\sqrt{N}}\right)$ | $O\left(\frac{1}{\sqrt[3]{N}}\right)$ | $O\left(\frac{1}{\sqrt[3]{N}}\right)$ | $O\left(\frac{1}{\sqrt[4]{N}}\right)$ |

*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 Algorithm 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 presented in subsubsection VI.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.[35, 36]

**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 $\left\{\underline{\mathbf{z}}_s\right\}_{s\in\{1,\ldots,N_{\text{sequence}}\}}$

**for** $s \leftarrow 1$ ***to*** $N_{\text{sequence}}$ **do**
  **for** $n \leftarrow 1$ ***to*** $D$ **do**
    $\left[\underline{\mathbf{y}}_s\right]_n \sim \mathcal{N}(0,1)$
  $\underline{\mathbf{z}}_s \leftarrow \underline{\mathbf{y}}_s / \|\underline{\mathbf{y}}_s\|$ // point on the unit $(D-1)$-sphere
**return** $\{\underline{\mathbf{z}}_s\}_{s\in\{1,\ldots,N_{\text{sequence}}\}}$

**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 qrns generation algorithm is a low-discrepancy qrns generation algorithm for uniform sampling within the unit $n$-hypercube, and is listed in Algorithm 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.[37, 38]

**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 $\left\{\underline{\mathbf{y}}_s^n\right\}_{s\in\{1,\ldots,N_{\text{sequence}}\}}$

$\phi_n : (\phi_n)^{n+1} = \phi_n + 1$ // generalized golden ratio numbers
$\alpha \leftarrow \underline{\mathbf{0}}_{n\times 1}$
**for** $j \leftarrow 1$ ***to*** $n$ **do**
  $[\alpha]_j \leftarrow \left(\frac{1}{\phi_n}\right)^j$
**for** $s \leftarrow 1$ ***to*** $N_{\text{sequence}}$ **do**
  **for** $j \leftarrow 1$ ***to*** $n$ **do**
    $\left[\underline{\mathbf{y}}_s^n\right]_j \leftarrow \left(s\,[\alpha]_j\right) \bmod 1$
**return** $\left\{\underline{\mathbf{y}}_s^n\right\}_{s\in\{1,\ldots,N_{\text{sequence}}\}}$

**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.1). These algorithms rely on the Roberts quasi-random number sequence on the $(D-1)$ unit hypercube (see Algorithm 5), although in principle any other such algorithm could be used for this purpose as well.[37]

The Roberts algorithm for generation of quasi-uniformly distributed sequences of points on the unit $(D-1)$-sphere is presented as Algorithm 6. This algorithm was presented in three-dimensions (3D) for the 2-sphere by Brannon, and it was implemented by Roberts.[37, 39] 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.[40, 41]

**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 $\left\{\underline{\mathbf{z}}_p\right\}_{p\in\{1,\ldots,N_{\text{samples/shell}}\}}$

$\left\{\underline{\mathbf{y}}_p^{(D-1)}\right\}_{p\in\{1,\ldots,N_{\text{samples/shell}}\}} \leftarrow$ output of Algorithm 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**
  $\underline{\mathbf{x}} \leftarrow \underline{\mathbf{y}}_p^{(D-1)}$
  $\theta \leftarrow 2\pi\left[\underline{\mathbf{x}}\right]_1$
  **for** $j \leftarrow 2$ ***to*** $D-1$ **do**
    $\lambda_{(j-1)} : \sin\left(\lambda_{(j-1)}\right) = 2\left[\underline{\mathbf{x}}\right]_j - 1$
  $\underline{\mathbf{z}} \leftarrow \underline{\mathbf{0}}_{D\times 1}$
  $[\underline{\mathbf{z}}]_1 \leftarrow \cos(\theta)\prod_{k=1}^{D-2}\cos(\lambda_k)$
  $[\underline{\mathbf{z}}]_2 \leftarrow \sin(\theta)\prod_{k=1}^{D-2}\cos(\lambda_k)$
  **if** $D > 3$ **then**
    **for** $j \leftarrow 3$ ***to*** $D-1$ **do**
      $[\underline{\mathbf{z}}]_j \leftarrow \sin\left(\lambda_{(j-2)}\right)\prod_{k=j-1}^{D-2}\cos(\lambda_k)$
  $[\underline{\mathbf{z}}]_D \leftarrow \sin\left(\lambda_{(D-2)}\right)$
  $\underline{\mathbf{z}}_p \leftarrow \underline{\mathbf{z}}$
**return** $\left\{\underline{\mathbf{z}}_p\right\}_{p\in\{1,\ldots,N_{\text{samples/shell}}\}}$

**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 Algorithm 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.[42] This method utilizes the regularized incomplete beta ($h_m$) function, shown in Definition VI.2.[42]

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

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

where $m \in [3, \infty)$, and where $B(z; a, b)$ is the incomplete beta function, given by

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

**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 $\left\{\underline{\mathbf{z}}_p\right\}_{p \in \{1,\cdots,N_{\text{samples/shell}}\}}$

$\left\{\underline{\mathbf{y}}_p^{(D-1)}\right\}_{p \in \{1,\cdots,N_{\text{samples/shell}}\}} \leftarrow$ output of Algorithm 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**
  $\underline{\mathbf{x}} \leftarrow \underline{\mathbf{y}}_p^{(D-1)}$
  $\underline{\mathbf{z}} \leftarrow \left[\cos\left(2\pi \left[\underline{\mathbf{x}}\right]_1\right) \qquad \sin\left(2\pi \left[\underline{\mathbf{x}}\right]_1\right)\right]^T$
  $\underline{\mathbf{z}} \leftarrow \left[\left(\sqrt{1 - \left(1 - 2\left[\underline{\mathbf{x}}\right]_2\right)^2}\right)\underline{\mathbf{z}}^T \qquad 1 - 2\left[\underline{\mathbf{x}}\right]_2\right]^T$
  **if** $D > 3$ **then**
    **for** $j \leftarrow 3$ ***to*** $D-1$ **do**
      $\underline{\mathbf{z}} \leftarrow \left[\left(\sqrt{1 - \left(1 - 2h_j^{-1}\left(\left[\underline{\mathbf{x}}\right]_j\right)\right)^2}\right)\underline{\mathbf{z}}^T \qquad 1 - 2h_j^{-1}\left(\left[\underline{\mathbf{x}}\right]_j\right)\right]^T$
  $\underline{\mathbf{z}}_p \leftarrow \underline{\mathbf{z}}$
**return** $\left\{\underline{\mathbf{z}}_p\right\}_{p \in \{1,\ldots,N_{\text{samples/shell}}\}}$

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


## 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. Finally, the authors are grateful for Dr. Peter Schulte at the Charles Stark Draper Laboratory, Inc. and Dr. Julian Brew at the Johns Hopkins University Applied Physics Laboratory, whose insightful comments in review of earlier versions of this manuscript helped in its improvement.


## References


[1] 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, 2010, Chaps. 1, 14, pp. 1–11, 329–330.

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

[3] 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.

[4] Schaub, H., and Junkins, J. L., "Spacecraft Formation Flying," *Analytical Mechanics of Space Systems*, AIAA Education Series, Reston, VA, 2009, Chap. 14, 2nd ed., pp. 593–673. doi:10.2514/4.105210.

[5] Tapley, B., Ries, J., Bettadpur, S., Chambers, D., Cheng, M., Condi, F., Gunter, B., Kang, Z., Nagel, P., Pastor, R., Pekker, T., Poole, S., and Wang, F., "GGM02 – An improved Earth gravity field model from GRACE," *Journal of Geodesy*, Vol. 79, No. 8, 2005, pp. 467–478. doi:10.1007/s00190-005-0480-z.

[6] Sugihara El Maghraby, A. K., Grubisic, A., Colombo, C., and Tatnall, A., "A novel approach to microwave interferometric radiometry in the geostationary orbit using formation flight," *67th International Astronautical Congress of the International Astronautical Federation*, Guadalajara, Mexico, Sep. 2016, pp. 1–14.

[7] Koenig, A. W., Macintosh, B., and D'Amico, S., "Formation Design of Distributed Telescopes in Earth Orbit for Astrophysics Applications," *Journal of Spacecraft and Rockets*, Vol. 56, No. 5, 2019, pp. 1462–1477. doi:10.2514/1.A34420.

[8] 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. doi:10.1088/1674-1056/24/5/059501.

[9] 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.

[10] Klinkrad, H., *Space Debris: Models and Risk Analysis*, Springer-Praxis Books in Astronautical Engineering, Springer-Verlag Berlin Heidelberg, 2006, Chaps. 1–2, pp. 1–58. doi:10.1007/3-540-37674-7.

[11] Chobotov, V. A., Karrenberg, H. K., Chao, C., Miyamoto, J. Y., Lan, T. J., Kechichian, J. A., Johnson, C. G., Alekshun, J., and Campbell, E. T., "Space Debris," *Orbital Mechanics*, edited by V. A. Chobotov, AIAA Education Series, American Institute of Aeronautics and Astronautics, Reston, VA, 2011, Chap. 13, 3rd ed., pp. 301–334. doi:10.2514/4.862250.

[12] Hilton, S., Sabatini, R., Gardi, A., Ogawa, H., and Teofilatto, P., "Space traffic management: towards safe and unsegregated space transport operations," *Progress in Aerospace Sciences*, Vol. 105, Feb. 2019, pp. 98–125. doi:10.1016/j.paerosci.2018.10.006.

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

[14] Newman, L. K., Frigm, R. C., Duncan, M. G., and Hejduk, M. D., "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, Sep. 2014, pp. 1–14.

[15] 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.

[16] 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.

[17] Chan, F. K., *Spacecraft collision probability*, Aerospace Press and American Institute of Aeronautics and Astronautics, El Segundo, CA and Reston, VA, 2008. doi:10.2514/4.989186.

[18] 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.

[19] 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.

[20] 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, Jan. 2015, pp. 1–14.

[21] 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, Feb. 2016, pp. 1–15.

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

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

[24] 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. doi:10.23919/ACC.2004.1384365.

[25] 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.

[26] 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.

[27] Bernstein, D. S., *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed., Princeton University Press, Princeton, NJ, 2009, Chaps. 3, 8, pp. 179–252, 417–542.

[28] Heil, C., "Metric and Normed Spaces," *Introduction to Real Analysis*, Graduate Texts in Mathematics, Springer International Publishing, Cham, 2019, Chap. 1, pp. 15–32. doi:10.1007/978-3-030-26903-6.

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

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

[31] 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.

[32] Bhattacharya, P., and Burman, P., "12 - Multivariate Analysis," *Theory and Methods of Statistics*, Academic Press, 2016, pp. 383–429. doi:10.1016/B978-0-12-802440-9.00012-6.

[33] Nandagopal, M., "On the Evaluation of Correction Terms of Gaussian Integration," *Computing in Science & Engineering*, Vol. 13, No. 1, 2011, pp. 58–61. doi:10.1109/MCSE.2011.12.

[34] 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.

[35] 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.

[36] 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.

[37] Roberts, M., "The Unreasonable Effectiveness of Quasirandom Sequences," `extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/`, Aug. 2018. Online; accessed 19 November 2019.

[38] Weisstein, E. W., "Golden Ratio (From MathWorld–A Wolfram Web Resource)," `https://mathworld.wolfram.com/GoldenRatio.html`, Jan 2020. Online; accessed 6 January 2020.

[39] 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.

[40] Weisstein, E. W., "Latitude (From MathWorld–A Wolfram Web Resource)," `http://mathworld.wolfram.com/Latitude.html`, Jan 2020. Online; accessed 6 January 2020.

[41] Blumenson, L. E., "A Derivation of n-Dimensional Spherical Coordinates," *The American Mathematical Monthly*, Vol. 67, No. 1, 1960, pp. 63–66. doi:10.2307/2308932.

[42] 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, Oct. 2015, pp. 28–53. doi:10.1016/j.cam.2015.02.023.