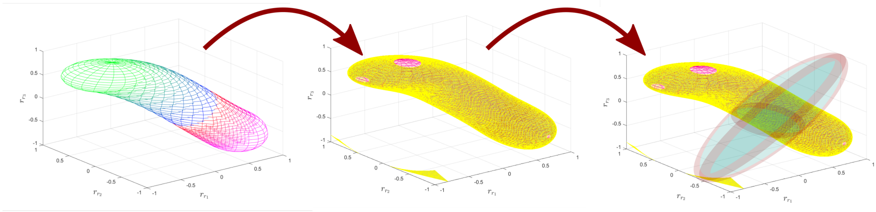
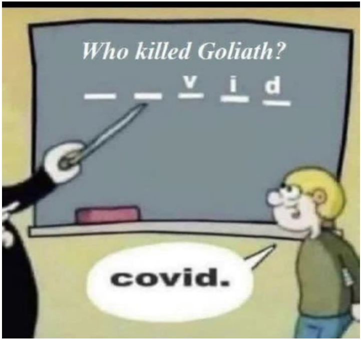
A polynomial optimization-based approach for computing the orbital collision probability for long-term encounters

D. Arzelier, F. Bréhard, M. Joldes, J.B. Lasserre, S. Laurens, A. Rondepierre
LAAS-CNRS

April 27, 2021

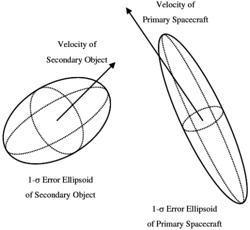




What's your favourite method
to compute volumes of semi-algebraic
sets?

_ _ _ _ _ _ _ S

General encounter model and probability of collision



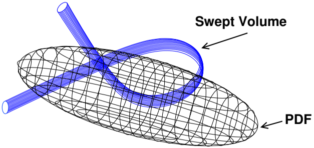
Primary P (**operational**) and secondary S (**debris**)

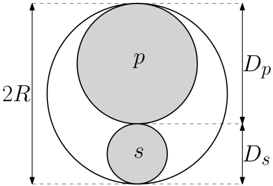
- Relative dynamics:
  $x^T = \left((r_s - r_p)^T, (v_s - v_p)^T\right) \in \mathbb{R}^6$
  $\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \quad t \in [t_0, t_f]$
- Gaussian distribution μ_0, density ρ_0 for x_0
- Collision tube (swept volume):
  $\mathcal{X}_0 = \{x_0 \in \mathbb{R}^6 \mid \exists t \in [t_0, t_f],\ x(t|x_0) \in \mathcal{X}_R\}$
  $\mathcal{X}_R = \{x \in \mathbb{R}^6 \mid \|r\|^2 - R^2 \leq 0\}$

Probability of collision:

$$\mathcal{P}_c = \mu_0(\mathcal{X}_0) = \int_{\mathcal{X}_0} \rho_0(x_0)\mathrm{d}x_0$$

- ρ_0: Gaussian PDF
- $\mathcal{X}_0$: swept-volume

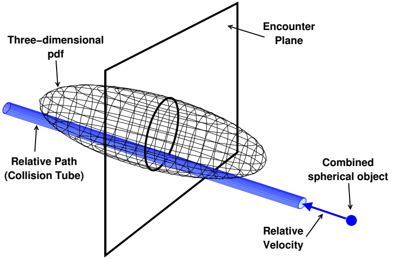




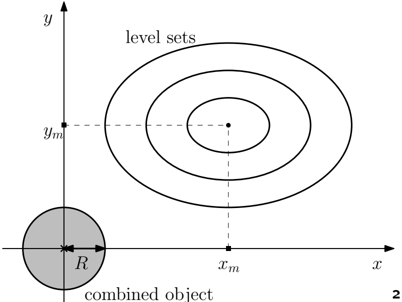
The *swept volume* I

Collision domain: $\mathcal{X}_0 = \{x_0 \in \mathbb{R}^6 \mid \exists t \in [t_0, t_f],\ x(t|x_0) \in \mathcal{X}_R\}$.

The ***swept volume*** is visualized in 3D, fonction of the relative position.



1. Short-term encounters, no velocity uncertainty
   $\rightsquigarrow$ the *swept volume* is a **cylinder** (collision tube).



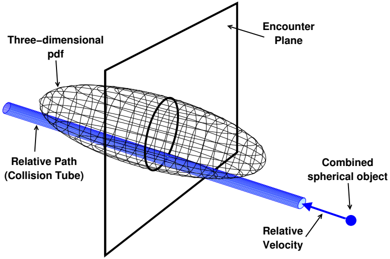
- Analytic method
- Effective a priori error bounds
- Linear complexity
- CNES implemented it and now uses it

The *swept volume* II

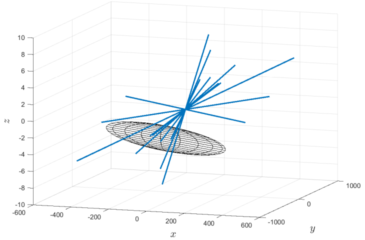
Collision domain: $\mathcal{X}_0 = \{x_0 \in \mathbb{R}^6 \mid \exists t \in [t_0, t_f],\ x(t|x_0) \in \mathcal{X}_R\}$.

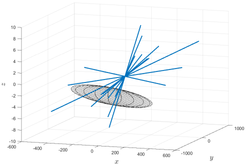
The ***swept volume*** is visualized in 3D, fonction of the relative position.

1. Short-term encounters, no velocity uncertainty
   $\rightsquigarrow$ the *swept volume* is a **cylinder** (collision tube).



2. Linear encounters, with velocity uncertainty
   $\rightsquigarrow$ the *swept volume* is a **cylinder for each fixed velocity vector**.





- The *swept volume* is a **cylinder for each fixed velocity**:

$$\mathcal{X}^0_\infty = \{x_0 \in \mathbb{R}^6 \mid v_0 \in \mathbb{R}^3 \text{ and } ||v_0 \times r_0||_2 \leq R||v_0||_2\}.$$

- 6D Integral to evaluate:

$$\mathcal{P}_c = \frac{1}{(2\pi)^3\sqrt{\det(P_{x_0})}} \int\limits_{\mathcal{X}^0_\infty} \exp\left(-\frac{1}{2}(x_0 - m_{x_0})^T P_{x_0}^{-1}(x_0 - m_{x_0})\right) \mathrm{d}x_0.$$

Current method:

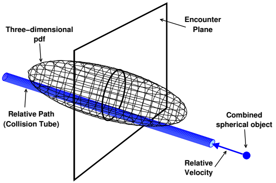
- N Gaussian random velocity samples v_i;
- for each v_i, compute with the super FastRelax 2D algorithm + recent improvements based on saddle-point method;
- do the average.

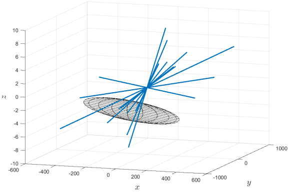
This is an integral of a holonomic function on a semi-algebraic domain.

TODO (1): "half-a-page" algorithm written in C to evaluate this with the required accuracy.

Recap of *swept volume* "types"

3D Projection of ***swept volume***, fonction of relative positions.

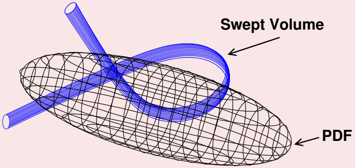




1. Short-term encounters, no velocity uncertainty ⇝ **cylinder**.

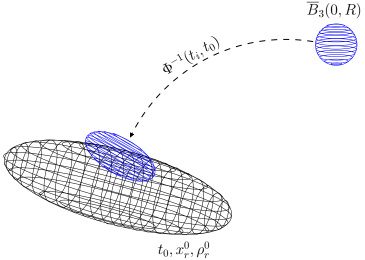
2. Linear encounters, velocity uncertainty ⇝ **cylinder for each fixed velocity**.

3. Non-linear encounters, **with** or without velocity uncertainty
   - What can we do?



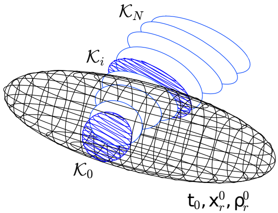
- The set $\bar{B}_3(0,R) \times \mathbb{R}^3$ is *(retro)-propagated* from t_i to t_0

$$\mathcal{K}_i := \Big\{x_0 \in \mathbb{R}^6 : \\ R^2 - x_0^T \Phi(t_i,t_0)^T I_{11} \Phi(t_i,t_0) x_0 \geq 0\Big\},$$



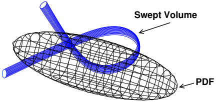
- The sets $\mathcal{K}_i$ are neither disjoint, nor compact in general

$$\bigcup_{i=1}^{N} \mathcal{K}_i \subseteq \mathcal{X}_0 = \Big\{x_0 \in \mathbb{R}^6 : \exists\, t \in [t_0,t_f]\,\text{t.q.} \\ R^2 - x_0^T \Phi(t,t_0)^T I_{11} \Phi(t,t_0) x_0 \geq 0\Big\},$$



$$\mathcal{P}_c([t_0,t_f]) \simeq \int_{\bigcup_{i=1}^{N} \mathcal{K}_i} \rho_0(x_0)\mathrm{d}x_0,$$

$$\tilde{\mathcal{P}}_c \simeq \int_{\bigcup_{i=1}^{N} \mathcal{K}_i} \rho(x)\mathrm{d}x,$$



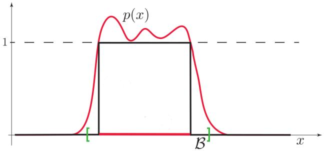
1. Outer-approximate the union $\mathcal{K} := \bigcup_{i=1}^{N} \mathcal{K}_i$ by a polynomial super-level set (PSS) i.e., $p_d \in \mathbb{R}[x]_d$ s.t.
$$\mathcal{K} \subseteq \mathrm{PSS}_{p_d} := \{x : p_d(x) \geq 1\}.$$
2. Compute the integral of a Gaussian on a PSS

Reasons: Visualisation, better numerical behaviour, Gaussian on a PSS is holonomic...

PSS for the *swept volume* I

Volume of a semi-algebraic set*

$$\mathrm{vol}(\mathcal{K}) = \int_{\mathcal{K}} \mathrm{d}\lambda_{\mathcal{B}}(x),$$



Optim. over measures

Duality

Optim. over polynomials up to deg. d

$$w_d^* \downarrow v^* = \mathrm{vol}(\mathcal{K})$$

$$\begin{aligned} v^* = \quad & \sup_{\mu} \mu(\mathcal{K}) \\ & \mu \leq \lambda_{\mathcal{B}}, \\ & \mathrm{supp}(\mu) \subseteq \mathcal{K}, \\ & \mu \geq 0. \end{aligned}$$

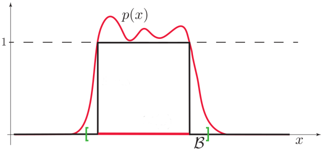
$$\begin{aligned} w_d^* = \quad & \inf_{p \in \mathbb{R}[x]_d} \|p\|_1, \\ & \text{s.t.} \quad \begin{array}{l} p \geq 0 \,\mathrm{on}\, \mathcal{B}, \\ p \geq 1 \,\mathrm{on}\, \mathcal{K}. \end{array} \end{aligned}$$

SOS

SDP Programming

*Lasserre, Henrion; [DabbeneHenrionLagoa]

$$\mathrm{vol}\left(\bigcup_{i=1}^{N}\mathcal{K}_i\right) = \int_{\bigcup_{i=1}^{N}\mathcal{K}_i} \mathrm{d}\lambda_{\mathcal{B}}(x),$$



Optim. over measures

Duality

Optim. over polynomials up to deg. d

$$w_d^* \downarrow v^* = \mathrm{vol}\left(\bigcup_{i=1}^{N}\mathcal{K}_i\right)$$

$$\begin{aligned} v^* = \quad & \sup_{\mu_i} \sum_{i=1}^{N} \mu_i\left(\mathcal{K}_i\right) \\ & \sum_{i=1}^{N} \mu_i \leq \lambda_{\mathcal{B}}, \\ & \mathrm{supp}(\mu_i) \subseteq \mathcal{K}_i, \\ & \mu_i \geq 0. \end{aligned}$$

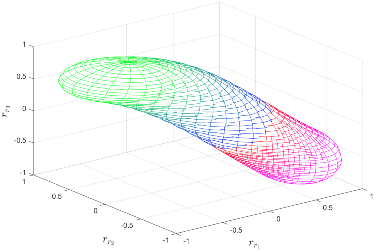
$$\begin{aligned} w_d^* = \quad & \inf_{p\in\mathbb{R}[x]_d} \|p\|_1, \\ \text{s.t.} \quad & p \geq 0 \text{ on } \mathcal{B}, \\ & p \geq 1 \text{ on } \mathcal{K}_i. \end{aligned}$$

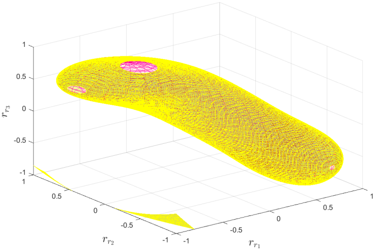
SOS

SDP Programming

*Lasserre, Henrion; [DabbeneHenrionLagoa]

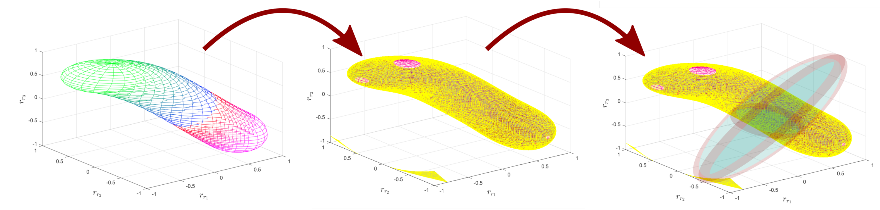
Polynomial Approximations for the *swept volume*





(PSS) of fixed degree d:

- $\rightsquigarrow$ **Polynomial optimization problem, LMI, SDP optimization**
- $\rightsquigarrow$ Prototype Implementation in Matlab
- $\rightsquigarrow$ Tested on some cases [Alfano2009]
- $\rightsquigarrow$ Degree $d = 4, 6, 8$
- $\rightsquigarrow$ Better results in 3D (less overestimation)



Two steps:

1. Implicit Representation of the integration domain by a PSS_{p_d}.
2. Gaussian integration (3D or 6D), on PSS_{p_d} :

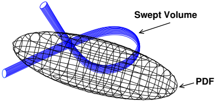
$$\mathcal{P}_c \simeq \frac{1}{\sqrt{(2\pi)^n \det(P)}} \int_{\mathrm{PSS}_{p_d}} e^{-\frac{1}{2}(x-x_m)' P^{-1}(x-x_m)} \mathrm{d}x.$$

- For $n = 3$, adaptive Gauss quadrature for implicit domains [Saye2015] ;
- For $n = 6$, classical Monte-Carlo sampling.

Swept volume as union of semi-algebraic sets

Integration of a Gaussian PDF on a union of semi-algebraic sets: Second Approach*

$$\tilde{\mathcal{P}}_c \simeq \int_{\bigcup_{i=1}^{N} \mathcal{K}_i} \rho_0(x_0) \mathrm{d}x_0,$$



Optim. over measures ↔ Optim. over moment sequences → Optim. over moment seq. up to deg. d

$$\overline{\kappa}^* = \sup_{\mu_i} \sum_{i=1}^{N} \mu_i(\mathcal{K}_i)$$
$$\sum_{i=1}^{N} \mu_i \leq \rho,$$
$$\mathrm{supp}(\mu_i) \subseteq \mathcal{K}_i,$$
$$\mu_i \geq 0.$$

$\overline{\kappa}^*_d \downarrow \overline{\kappa}^* = \tilde{P}_c$

$$\overline{\kappa}^*_d = \sup_{m_{ij}} \sum_{i=1}^{N} m_{i0}$$

Inequality constr

Localization

SDP Mom matrix

$\rightsquigarrow$ Infinite dim. LP

$\rightsquigarrow$ $m_{ij} = \int_{\Omega} x^j \mathrm{d}\mu_i(x)$

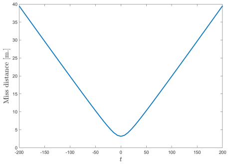
$\rightsquigarrow$ Need to compute the moments of ρ.

↓

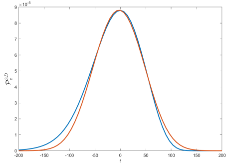
SDP Programming

TODO (2): efficient/reliable computation of moments of Gaussian on balls

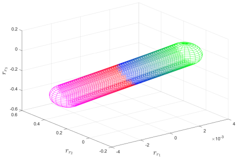
* J.-B. Lasserre & Co-authors



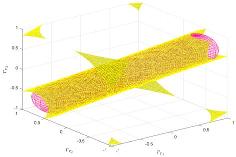
(a) Miss distance (en m)



(b) Instantaneous PoC with (in blue) or without (in red) velocity uncertainty at TCA

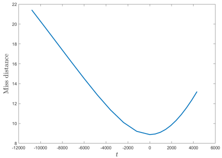


(c) 3D Swept-volume (no velocity uncertainty)

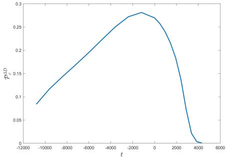


(d) PSS Approximation $d = 4$

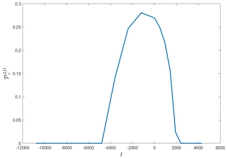
Figure – Test Case Alfano-7: Brute-force-PoC $=0.000158$ (10^6 samples) ; PoC $=0.000165$, with PSS $d = 4$, in 3D.



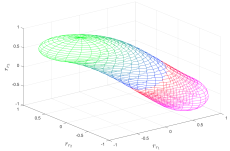
(a) Miss distance (en m)



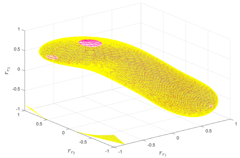
(b) Instantaneous PoC with velocity uncertainty at TCA



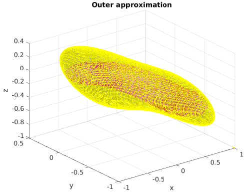
(c) Instantaneous PoC without velocity uncertainty at TCA



(d) 3D Swept-volume (no velocity uncertainty)

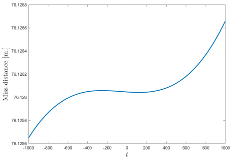


(e) PSS Approximation (without velocity uncertainty) $d = 8$

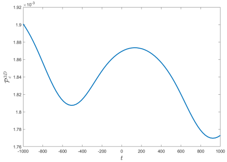


(f) Projection of PSS6D approximation, $d = 8$, for a specific velocity = mean velocity at TCA.

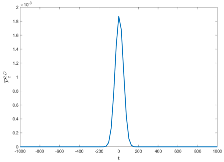
Figure – Test Case Alfano-9; Brute-force-PoC-3D $=0.287322$, (10^6 samples); PoC $=0.2825$, for PSS $d = 8$, in 3D; Brute-force-PoC-6D $=0.36336$ (10^6 samples); PoC $=0.506297$, for PSS $d = 8$, in 6D.



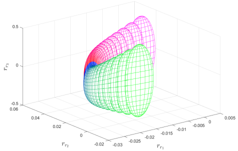
(a) Miss distance (en m)



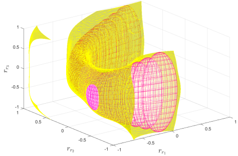
(b) Instantaneous PoC with velocity uncertainty at TCA



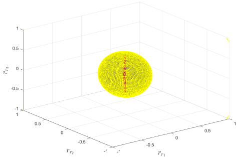
(c) Instantaneous PoC without velocity uncertainty at TCA



(d) 3D Swept-volume (no velocity uncertainty)



(e) PSS Approximation (no velocity uncertainty) $d = 8$



(f) Projection of PSS6D approximation, $d = 8$, for a specific velocity = mean velocity at TCA.

Figure – Test Case Alfano-11; Brute-force-PoC-3D=0.0026, same result for PSS $d = 8$, in 3D; Brute-force-PoC-6D =0.0032 (10^5 samples); PoC =0.06239, PSS $d = 8$, in 6D.

Conclusion

- Generalization of methods for calculating the probability of collision between two spacecraft, for cases where the simplified 2D model is not realistic enough
- Several methods analyzed and proposed by trying to gradually increase both the modeling and computation complexity
- Need for clarification between modeling and calculation method, validation of models and validation of methods
- Significant work needed for the discrimination, analysis and evaluation of each type of encounter
- TODO (1): "half-a-page" algorithm written in C to evaluate this with the required accuracy

$$\mathcal{X}_{\infty}^{0} = \{x_0 \in \mathbb{R}^6 \mid v_0 \in \mathbb{R}^3 \text{ and } ||v_0 \times r_0||_2 \leq R||v_0||_2\}.$$

$$\mathcal{P}_c = \frac{1}{(2\pi)^3\sqrt{\det(P_{x_0})}} \int\limits_{\mathcal{X}_{\infty}^{0}} \exp\left(-\frac{1}{2}(x_0 - m_{x_0})^T P_{x_0}^{-1}(x_0 - m_{x_0})\right) \mathrm{d}x_0.$$

- TODO (2): efficient/reliable computation of moments of Gaussian on balls