6.6 Intruders and Moving Constraints

Intruder behaviour: *Adversarial behaviour* of moving obstacle is trying to destroy avoiding our UAS. The *Intruder* UAS [1] is not trying to hurt our *UAS* actively. The *Adversarial behaviour* is neglected in this work. The non-cooperative avoidance is assumed, it can be relaxed to *cooperative avoidance* in *UTM controlled airspace.*

Intruder information: The *observable intruder information set* for any kind of intruder, obtained trough sensor/C2 line, is following:

1. *Position* - position of intruder in *local* or *global* coordinate frame, which can be transformed into *avoidance grid coordinate frame.*

2. *Heading and Velocity* - intruder heading and linear velocity in avoidance grid coordinate frame.

3. *Horizontal/Vertical Maneuver Uncertainty Spreads* - how much can an *intruder* deviate from *original linear path* in *horizontal/vertical* plane in *Global coordinate Frame.*

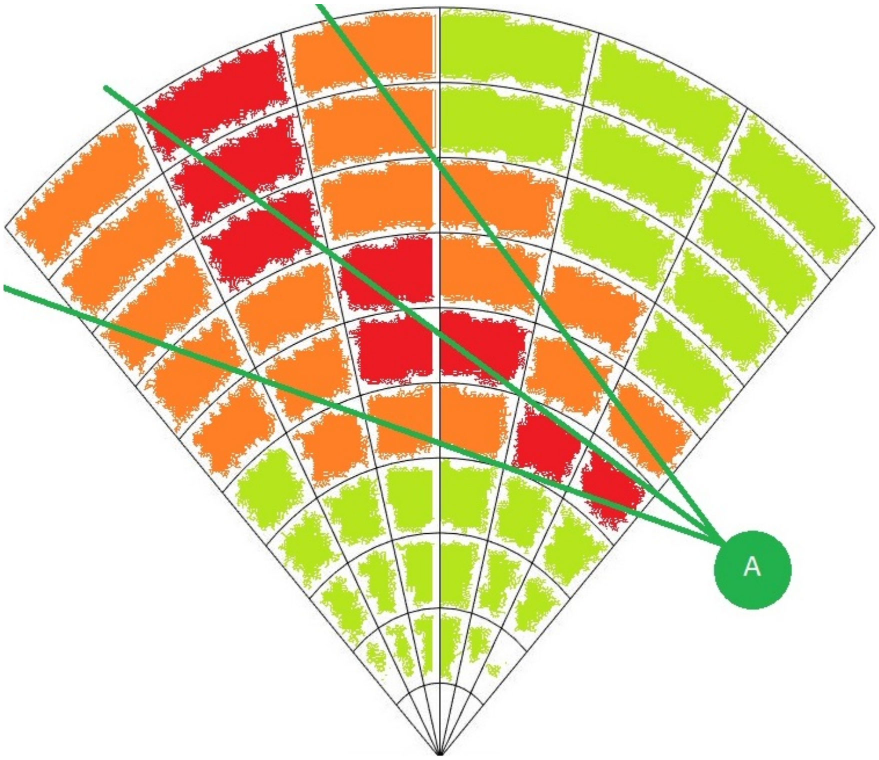

Figure 6.1: Intruder UAS intersection rate along expected trajectory.

Example of Intruder Intersection: Lets neglect the *time-impact* aspect on *intersection.* The *intruder* (black "I" circle) is intersecting one *avoidance grid horizontal slice* (fig. 6.1).

The intruder is moving along linear path approximation based on velocity (middle green line). The *Horizontal Maneuver Uncertainty spread* is in *green line boundary area intruder intersection rating* is denoted as green-orange-red cell fill reflecting intersection severity: red is high rate of intersection, orange is medium rate of intersection and green is low rate of intersection.

Moving Threats: The *UAS* can encounter following threats during the *mission execution*:

1. *Non-cooperative Intruders* - the intruders whom does not implement any approach to ensure mutual avoidance efficiency.

2. *Cooperative Intruders* - the intruders whom actively communicate or follow common agreed behaviour pattern (ex. Rules of the Air).

3. *Moving Constraints* - the constrained portion of *free* space which is shifting its boundary over time (ex. Short term bad weather).

Note. Our approach considers only *UAS* intruders, because *Data Fusion* considers data received trough *ADS-B* messages. The *Intruders* extracted from *LiDAR* scan were not considered (ex. birds). The proposed *intruder intersection models* are reusable for other *intruder sources.*

Approach Overview: The *Avoidance Grid* (def. **??**) is adapted to *LiDAR* sensor. The *euclidean grid intersections* are fairly simple. The *polar coordinates grid* are not. The need to keep *polar coordinates grid* is prevalent, because of fast *LiDAR* reading assessment. There are following commonly known methods to address this issue:

1. *Point-cloud Intersections* - the *threat impact area* is discredited into sufficiently thick point cloud. This point-cloud have *point impact rate* and *intersection time* assigned to each point. The *point-cloud* is projected to *Avoidance Grid.* If *impact point* hits $cell_{i,j,k}$ the cell's impact rate is increased by amount of *point impact rate.* The final *threat impact rate* in $cell_{i,j,k}$ is given when *all* points from point cloud are consumed. Close point problem [2] was solved by application of method [3].

2. *Polygon Intersections* - the *threat impact area* is modeled as polygon, each $cell_{i,j,k}$ in *Avoidance Grid* is considered as *polygon.* There is a possibility to calculate cell space geometrical inclusive intersection. The *impact rate* is then given as rate between *intersection volume* and $cell_{i,j,k}$ volume. The algorithm used for intersection selected based on:[4] the selected algorithm *Shamos-Hoey* [5].

Note. The *Intruder Intersection* models are based on *analytically geometry* for *cones and ellipsoids* taken from [6].

6.6.1 Intruder Behaviour Prediction

Idea: *Intruder Intersection Models* is about space-time intersection of *intruder body* with *avoidance Grid* and *Reach Set*:

1. The *UAS* reach set defines *time boundaries* to *enter/leave* cell in avoidance grid.
2. The *Intruder* behavioral pattern defines *rate* of *space intersection* with cell bounded space in avoidance grid.

The multiplication of *space intersection rate* and *time intersection rate* will give us *intruder intersection* rate for our *UAS* and intruder.

Intruder Dynamic Model: The definition of avoidance grid enforces the most of these methods to be numeric. Let us introduce intruder dynamic model:

$$\partial position/\partial time = velocity \quad \Big| \quad \begin{aligned} position_x(t) &= position_x(0) + velocity_x \times t \\ position_y(t) &= position_y(0) + velocity_y \times t \\ position_z(t) &= position_z(0) + velocity_z \times t \end{aligned} \tag{6.1}$$

Position vector in euclidean coordinates $[x, y, z]$ is transformed into *Avoidance Grid* coordinate frame. Velocity vector for $[x, y, z]$ is *estimated and not changing.* The time is in interval $[entry, leave]$, where *entry* is intruder entry time into avoidance grid and *leave* is intruder leave time from avoidance grid.

Note. If *intruder* is considered, time of entry is marked as $intruder_{entry,k}$ where k is intruder identification, time of leave is marked as $intruder_{leave,k}$ where k is intruder identification.

Cell Entry and Leave Times $UAS_{entry}(cell_{i,j,k})$ and $UAS_{leave}(cell_{i,j,k})$ are depending on intersecting *Trajectories* and *bounded cell space* (eq. **??**). There is *Trajectory Intersection* function from (def. **??**) which evaluates *Trajectory segment* entry and leave time.

The UAS *Cell Entry* time is given as minimum of all *passing trajectory segments* entry times (eq. 6.2), if there is no *passing trajectories* the UAS *entry time* is set to 0.

$$UAS_{entry}(cell_{i,j,k}) = \min \left\{ \begin{aligned} &0, entry(Trajectory, cell_{i,j,k}) : \\ &\qquad Trajectory \in PassingTrajectories \end{aligned} \right\} \tag{6.2}$$

The UAS *Cell Leave* time is given as maximum of all *passing trajectory segments* entry times (eq. 6.3), if there is no *passing trajectories* the UAS *leave time* is set to 0.

$$UAS_{leave}(cell_{i,j,k}) = \max \left\{ \begin{aligned} &0, leave(Trajectory, cell_{i,j,k}) : \\ &\qquad Trajectory \in PassingTrajectories \end{aligned} \right\} \tag{6.3}$$

Time Intersection Rate: The key idea is to calculate how long the *UAS* and *Intruder* spends together in same space portion ($cell_{i,j,k}$). The *Intruder* can spent some time in $cell_{i,j,k}$ bounded by interval of *intruder* entry/leave time.

The *UAS* can spent some time, depending on *selected trajectory* from *Reach Set*. The time spent by UAS is bounded by entry (eq. 6.2) and leave (eq. 6.3).

The intersection duration of these two intervals creates *time intersection rate* numerator, the *maximal duration* of *UAS* stay gives us *denominator*. The *time intersection rate* is formally defined in (eq. 6.4).

$$time\begin{pmatrix} UAS, \\ Intruder, \\ cell_{i,j,k} = \circ \end{pmatrix} = \frac{\left|\begin{matrix} [intruder_{entry}(\circ), intruder_{leave}(\circ)] \\ \cap \\ [UAS_{entry}(\circ), UAS_{leave}(\circ)] \end{matrix}\right|}{|[UAS_{entry}(\circ), UAS_{leave}(cell_{\circ})]|} \tag{6.4}$$

Intruder Intersection Rate: The *Intruder Intersection Rate* (eq. 6.5) is calculated as *multiplication* of *space intersection rate* (defined later) and *time intersection rate* (eq. 6.4).

$$intruder\begin{pmatrix} UAS, \\ Intruder, \\ cell_{i,j,k} \end{pmatrix} = time\begin{pmatrix} UAS, \\ Intruder, \\ cell_{i,j,k} \end{pmatrix} \times space\begin{pmatrix} UAS, \\ Intruder, \\ cell_{i,j,k} \end{pmatrix} \tag{6.5}$$

Note. If there is no information to derive *Intruder* entry/leave time for cells the *time intersection rate* is considered 1.

The *Intruder cell reach* time (eq. 6.6) is bounded to discrete point in intersection model [2, 3]. The intruder *entry/leave time* is calculated similar to *UAS cell entry (eq. 6.2)/leave (eq. 6.3) time.*

$$pointReachTime(Intruder, point) = \frac{distance(Intruder.initialPosition, point)}{|Intruder.velocity} \tag{6.6}$$

Space Intersection Rate: The *Space Intersection Rate* reflects probability of *Intruder* intersection with portion of space bounded by $cell_{i,j,k}$, to be precise with intruder trajectory or vehicle body shifted along the trajectory. The principles for *space intersection rate* calculation are following:

1. *Line trajectory* - *intruder* trajectory is given by linear approximation (eq. 6.1), depending on *intruder size* the intersection with avoidance grid can be:

a. *Simple line* - intersection is going along the trajectory line line defined by intruder model (eq.6.1).

b. *Volume line* - intersection is going along the trajectory line defined by intruder model (eq. 6.1) and intruder‘s *body radius* is considered in intersection.

2. *Elliptic cone* - initial position is considered as the top of a cone, the main cone axis is defined by intruder linear trajectory (eq. 6.1) $time \in [0, \infty]$. The cone width is set by horizontal and vertical spread.

6.6.2 Linear Intersection

Idea: There are *small intruders* which have body *smaller* than average $cell_{i,j,k}$ cell size. Its trajectory will stick to *linear trajectory* prediction with high probability.

Space Intersection Rate: The *Space Intersection Rate* for $cell_{i,j,k}$ is implemented as simple point cloud intersection. Where *sufficiently thick* point cloud is defined along *line* (eq. 6.7):

$$position(time) = position(time_0) + velocity \times time, \quad time \in [0, \infty[\tag{6.7}$$

Then there exist projection function from local euclidean coordinates to local polar coordinates (eq. 6.8. The function projects intruder trajectory (eq. 6.7) to planar coordinates $[distance, horizontal^\circ, vertical^\circ]$ as a set of sufficiently thick point cloud.

$$polarSet : position(t) \rightarrow \{[distance, horizontal^\circ], vertical^\circ\} \tag{6.8}$$

The *space intersection rating* $SpaceIntersection(\circ)$ for line type is given as (eq. 6.9). If there exist non empty intersection of $polarSet \cap cell_{i,j,k}$ there is space intersection rate equal to 1, if intersection $polarSet \cap cell_{i,j,k} = \varnothing$ then the rate is zero.

$$space\begin{pmatrix} Intruder, \\ cell_{i,j,k} \end{pmatrix} = \begin{cases} 1 : & \exists point \in polarSet(eq.6.8) : point \in c_{i,j,k} \\ 0 : & \text{otherwise} \end{cases} \tag{6.9}$$

Note. The *intruder intersection rate* is multiplication of *space intersection rate* and time intersection rate. The *intersection rate* is calculated for *every intruder* and *selected intersection model* separately.

6.6.3 Body-volume Intersection

Idea: The *Intruder* has body volume greater than *average* $cell_{i,j,k}$ volume. The *intruder body* is considered as the ball moving along *intruder position.* The *intersection* of the intruder body is realized as sufficiently thick *point-cloud intersection.*

Space Intersection Rate - Body Volume: The *body volume mass* with center at $position(t)$ is moving along intruder trajectory prediction (eq. 6.10) in time interval $[0, \infty[$:

$$position(time) = position(time_0) + velocity \times time \tag{6.10}$$

The body *Volume ball* $Body(position(t), radius)$ (eq. 6.11) is defined as set of points in $\mathbb{R}^3$ euclidean space. The center is moving along the *position(t).* The body *volume ball* is a set of points sufficiently thick including also inner points. The *thickness* is guaranteed by existence of neighbour point which is close enough.

$$Body(position(t), radius) = \left\{ point \in \mathbb{R}^3 : \begin{array}{r} \|position(t) - point\| \leq radius \\ \forall point_i \exists point_{j \neq i}, \\ distance(point_i, point_j) \leq thickness \end{array} \right\} \tag{6.11}$$

The *polar volume ball polarBody* (eq. 6.12) is projection of body volume ball set $Body(position(t), radius)$ to a set of planar coordinates in avoidance grid coordinate frame:

$$polarBall(t) : Body(position(t), radius) \rightarrow \left\{ \begin{bmatrix} distance, horizontal^\circ, \\ vertical^\circ, intersectionTime \end{bmatrix} \right\} \tag{6.12}$$

The *space intersection rate for vehicle body* $space(Intruder, cell_{i,j,k})$ (eq. 6.13) is calculated as intersection of polar body volume ball and $cell_{i,j,k}$. If intersection is non empty then base probability is one, zero otherwise:

$$space \begin{pmatrix} Intruder, \\ cell_{i,j,k} \end{pmatrix} = \begin{cases} 1: & \exists point \in polarBall(eq.6.12) : point \in c_{i,j,k} \\ 0: & \text{otherwise} \end{cases} \tag{6.13}$$

Intersection Time: The *intersection time* id depending on point cloud (eq. 6.12) where each point *have intersection time* given as *body-center position* time (eq. 6.10).

Note. The *body-volume* intersection model, can insert the *multiple intersection times* into one $cell_{i,j,k}$. the *interval length* considers all of these for intersection rates (eq. 6.4).

6.6.4 Maneuverability Uncertainty Intersection

Idea: The *intruders* are not bullets they are not sticking to predicted linear paths. The *intruder* maneuverability is given as horizontal and vertical spread. Therefore *intruder reach set* will form a *elliptic cone*. This cone can be transformed into *finite discrete* point-cloud, each *point* should have assigned *severity* impact value. The point cloud intersection with *Avoidance Grid* will give us space impact of *uncertain* intruder.

Note. Following section will use condensed notation, due the equation complexity. The *terminology* is consistent with rest of section.

Sprace Intersection Rate - Body Volume Intersection: $P_T(i_k(x_s, v, \theta, \varphi), c_{i,j,k})$ computation is less straight-forward than other space intersection rates. First let us define the linear intruder i_k positions x at time t (eq. 6.14) model, where $x(t)$ defines intruder position in *avoidance grid euclidean coordinate frame* at time t_i, v defines intruder velocity, and t is time offset.

$$x(t) = x_s + v_I.t \tag{6.14}$$

Intruder *horizontal spread* θ and *vertical spread* φ are introduced. These spreads represents intruder deviation limits along from linear trajectory prediction $x(t) \in \mathbb{R}^3$. The example is given by (fig. 6.2) where the intruder starts at point x_s with fixed velocity v, the linear trajectory prediction is outlined by blue line. The *predicted intruder position* at time $t = 10s$ is given by $x(10)$ (blue point). The ellipsoidal space $E(x)$ is projected on the plane $D(x(t))$. The plane D (eq. 6.15) for point $x(t)$ and velocity v is defined as an orthogonal plane to velocity vector $v \in \mathbb{R}^3$ with origin at intruder position $x(t)$.

$$D(x(t), v) = \left\{ a \in \mathbb{R}^3 : (a - x(t)) \perp v, \right\} \tag{6.15}$$

To construct ellipsoidal space boundary on orthogonal plane $D(x(t), v)$ some parameters are defined in (eq. 6.16). The *scalar distance* $d_d x(t)$ is simple euclidean norm, *maximal horizontal offset* $d_\theta(x_t)$ is given as product of sinus of horizontal offset angle θ and scalar distance d_d, and *maximal vertical offset* $d_\varphi(x(t))$ is given a product of sinus of vertical offset angle φ and scalar distance d_d.

$$\begin{aligned} d_d &= d_d(x(t), x_s) = \|x(t) - x_s\|_2 \\ d_{\theta_{\max}} &= d_\theta(x(t)) = \sin\theta(i_k).d_d(x(t)) \\ d_{\varphi_{\max}} &= d_\varphi(x(t)) = \sin\varphi(i_k).d_d(x(t)) \end{aligned} \tag{6.16}$$

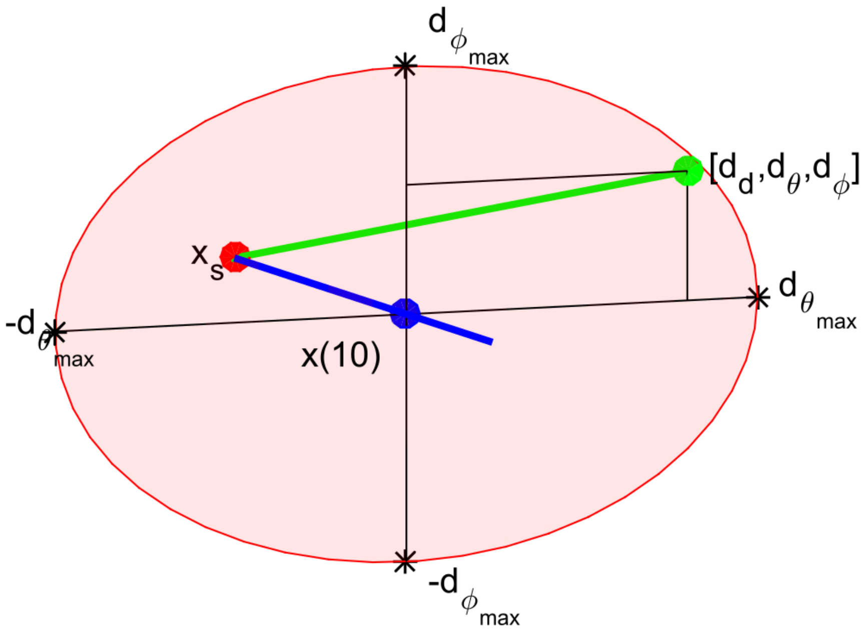

Figure 6.2: One rate position $[d_d, d_\theta, d_\varphi]$ (green). deviated from linear trajectory (blue line) at point $x(10)$.(blue) with initial position x_s (red)

The *Ellipsoid* $E(x(t), v)$ (eq. 6.17) for fixed intruder position $x(t)$ and fixed intruder velocity v is given as constrained portion of orthogonal plane $D(x(t), v)$. The constraint is defined by an internal coordinate frame $p \in \mathbb{R}^2$ which is space reduction of plane $D(x(t), v)$.

The internal coordinate frame $p \in \mathbb{R}^2$ has origin in $x(t) \to \mathbb{R}^2$. The points of plane p are bounded by projection $p = (b - x(t)) \to \mathbb{R}^2$, where $b \in D(x(t), v)$. The point of ellipsoidal p is then given as standard ellipse boundary with vertical span $d_\theta(x(t))$ and horizontal span $d_\varphi(x(t))$.

The 2D *Ellipsoid* $E(x(t), v)$ for specific time $t = 10s$ example is portrayed as red ellipsoid (in fig. 6.2).

$$E(x(t), v) = \left\{ \begin{matrix} b \in \mathbb{R}^3 : b \in D(x(t), v), p = (b - x(t)) \to \mathbb{R}^2, \\ \left(\dfrac{p(1)^2}{d_\theta(x(t))^2} + \dfrac{p(2)^2}{d_\varphi(x(t))^2} \right) \leq 1 \end{matrix} \right\} \tag{6.17}$$

The expected behaviour of an intruder i_k is to stick to predicted linear trajectory $x(t)$ (6.14). The probability of deviation should be decreasing with distance from ellipse center (fig. 6.3.).

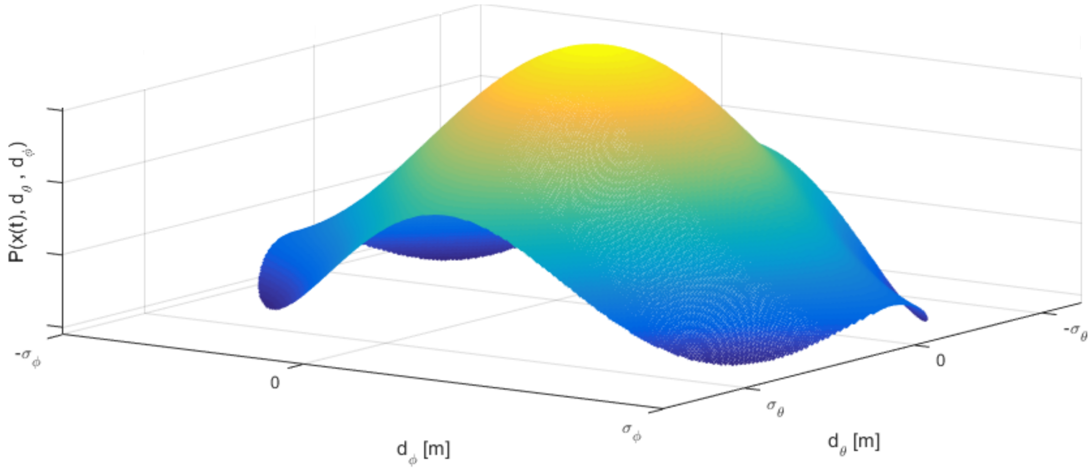

Figure 6.3: Probability of intruder i_k position in ellipsoid $E(x(t), v)$

Probability density function for ellipsoid $E(x(t), v)$defined in (eq. 6.17) is depending on maximal horizontal spread $d_\theta(x(t))$, maximal vertical spread $d_\varphi(x(t))$, defined by (eq. 6.16).

Two standard probabilistic distributions are established $\mathcal{N}(\mu_\theta, \sigma_\theta)$ (eq. 6.18) for horizontal spread $\theta(x(t))$ and $\mathcal{N}(\mu_\varphi, \sigma_\varphi)$ (eq. 6.19) for vertical spread $\varphi(x(t))$. The means μ_θ and μ_φ are set to zero, and internal coordinate frame $p \in \mathbb{R}^2$ where $x(t) \to \mathbb{R}^2$ is frame center. The variances σ_θ and σ_φ are set as maximal distances on horizontal/vertical spread axes $d_\theta(x(t))$ and $d_\varphi(x(t))$.

$$P(x(t), d_\theta) = \mathcal{N}(\mu_\theta, \sigma_\theta) = \mathcal{N}(0, d_\theta(x(t))) \tag{6.18}$$

$$P(x(t), d_\varphi) = \mathcal{N}(\mu_\varphi, \sigma_\varphi) = \mathcal{N}(0, d_\varphi(x(t))) \tag{6.19}$$

The combined *probability density function* for maximal spreads d_θ and d_φ is given by (eq. 6.20). Because probability density function is defined for internal space $p \in \mathbb{R}^2$ and one may need to calculate impact rate for cell space $c_{i,j,k} \in \mathbb{R}^3$.

The reduction from two parameter probability distribution function to scalar rate distribution function is needed. An scalar rate distribution function $P(x(t), d_\theta, d_\varphi)$ over ellipsoid $E(x(t), v)$ is defined as (eq.6.20), where final rate is given as average of two partial probabilities.

Final space intersection rate $P(x(t), d_\theta, d_\varphi)$ needs to be normalized to hold *normal distribution condition* (eq. 6.21). Normal distribution condition value (eq. 6.21) is given as surface integral over ellipsoid $E(x(0), v)$ with rate distribution function $P(x(t), d_\theta, d_\varphi)$.

$$P(x(t), d_\theta, d_\varphi) = \frac{\mathcal{N}(\mu_\theta, \sigma_\theta) + \mathcal{N}(\mu_\varphi, \sigma_\varphi)}{2} \tag{6.20}$$

$$\iint_{E(x(\tau))} P(x(t), d_\theta, d_\varphi)\, \mathrm{d}d_\theta\, \mathrm{d}d_\varphi = 1 \tag{6.21}$$

Final space intersection rate $P(x(t), c_{i,j,k}, \theta, \varphi)$ (space portion, time portion is calculated in (eq.6.5) is given by (eq. 6.23). Its mean value of all intersection rates $P(x(\tau), c_{i,j,k}, \theta, \varphi)$ where $\tau \in [i_e(c_{i,j,k}), i_l(c_{i,j,k})]$ is fixed point in intersection time interval.

An $P(x(\tau), c_{i,j,k}, \theta, \varphi)$ (6.22) is integration of rate density function $P(x(\tau), d_\theta, d_\varphi)$ (eq. 6.20) in surface $E(x(\tau), v)$ to cell $c_{i,j,k}$ volume intersection.

To get a volume integration partial rate in surface intersection must be integrated and normalized in time interval $\tau \in [i_e(c_{i,j,k}), i_l(c_{i,j,k})]$, the *base intersection probability* $P_T(i_k(x_s, v, \theta, \varphi), c_{i,j,k})$ is given by (eq. 6.23). Example of intersection of intruder i_r uncertain ellipsoid cone with avoidance grid $\mathcal{A}(t_i)$ is given in (fig. 6.4).

$$P(x(\tau), c_{i,j,k}, \theta, \varphi) = \iint_{E(x(\tau),v) \cap c_{i,j,k}} P(x(\tau), d_\theta, d_\varphi) \tag{6.22}$$

$$P_T(i_k(x_s, v, \theta, \varphi), c_{i,j,k}) = \frac{\int_{i_e(c_{i,j,k})}^{i_l(c_{i,j,k})} P(x(\tau), c_{i,j,k}, \theta, \varphi)\, \mathrm{d}\tau}{i_l(c_{i,j,k}) - i_e(c_{i,j,k})} \tag{6.23}$$

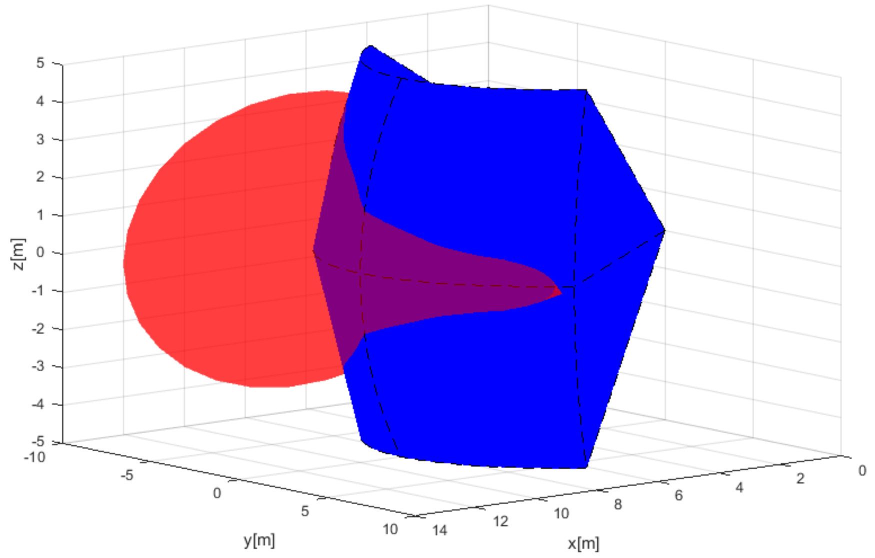

Figure 6.4: Avoidance grid $\mathcal{A}(t_i)$ (blue) intersection with elliptic cone intruder $i_k(x, v, \theta, \varphi)$ (red) example.

An *numeric approximation* of space intersection rate $P_T(i_k(x_s, v, \theta, \varphi), c_{i,j,k})$ is more implementation feasible than symbolic calculation due the multiple intersection constraints and bad intersection algorithm complexity.

Let us define homogeneous discrete subset of real numbers $\mathcal{R}$ which is non empty subset of real numbers $\mathbb{R}$. The set $\mathcal{R}$ (eq. 6.24) is homogeneous, that means for any equal interval $(i, i+1], i \in \mathbb{Z}$ subset the count of members is equal to some positive natural number k. The parameter k can be understand as *unit approximation density.*

Similarly the power sets $\mathcal{R}^2 \subset \mathbb{R}^2$, $\mathcal{R}^3 \subset \mathbb{R}^3$, ... $\mathcal{R}^i \subset \mathbb{R}^i, i \in \mathbb{N}^+$ keeps homogeneous distribution.

$$\mathcal{R} = \left\{ \begin{matrix} a \in \mathbb{R} : \forall i \in \mathbb{Z}, |i < a \leq i+1| = k, k \in \mathbb{N}^+, \\ \forall j \in \mathbb{N}^+ a_{j+1} - a_j = m, m \in \mathbb{R}^+ \end{matrix} \right\}, \mathcal{R} \subset \mathbb{R} \tag{6.24}$$

The orthogonal plane for $x(t), v, t \in \mathbb{R}$ is defined by (eq. 6.15). The orthogonality property is also kept for any subspace $\mathcal{R}^n \in \mathbb{R}^n, n \in \mathbb{N}^+$. Numeric approximation of $D(x(t), v)$ is given as $D_D(x(t), v)$ (eq. 6.25).

The only difference is that discrete approximation is countable $|D_D| = m, m \in \mathbb{N}^+$, but continuous representation $|D| \approx \infty$ is uncountable. Because ellipsoid is subset of orthogonal plane it keep its countability property, therefore E_D is also countable and must contains at-least one member.

$$D_D(x(t), v) = \left\{ a \in \mathcal{R}^3 : (a - x(t)) \perp v, \right\}, t \in \mathcal{R} \tag{6.25}$$

The *base ellipsoid* $E(x(t), v)$ for continuous-space is given by (eq. 6.17). Every element, expect the base of internal projection $\mathcal{R}^2$ and orthogonal plane D_D is same in discrete case $E_D(x(t), v)$ (eq. 6.26).

$$\bar{E}_D(x(t), v) = \left\{ \begin{matrix} b \in \mathcal{R}^3 : b \in D_D(x(t), v), p = (b - x(t)) \rightarrow \mathcal{R}^2, \\ \left(\dfrac{p(1)^2}{d_\theta(x(t))^2} + \dfrac{p(2)^2}{d_\varphi(x(t))^2} \right) \leq 1 \end{matrix} \right\}, t \in \mathcal{R} \tag{6.26}$$

The *numeric calculation disproportion* can occur in case that ellipsoid $\bar{E}_D(x(t), v)$ (6.26) in case of $d_\theta(x(t)) \approx 0$ and $d_\varphi(x(t)) \approx 0$. The count of ellipsoid members can be $|\bar{E}_D(x(t), v)| = 0$, which is in contradiction with assumption $|\bar{E}_D(x(t), v)| \neq 0$.

Let assume for discrete times $\tau = \{t_1, t_2, \ldots, t_i\}$, $i \in \mathbb{N}^+$ there exists ellipsoids $\bar{E}_D(x(t_1), v)$,$\bar{E}_D(x(t_1), v)$, ..., $\bar{E}_D(x(t_i), v)$ which are non empty and in space $\mathcal{R}^2$ in internal coordinate frame and space $\mathcal{R}^3$ in avoidance grid $\mathcal{A}(t_i)$ coordinate frame. The intersection of these partial ellipsoids in both spaces is equal to:

$$\bar{E}_D(x(t_1), v) \cap \bar{E}_D(x(t_2), v) \cdots \cap \ldots \bar{E}_D(x(t_i), v) = \varnothing \tag{6.27}$$

An *empty intersection* enables us to keep homogeneity property of ellipsoids by adding points so it is safe to add specific point $x(t)$ into empty ellipsoid. But only one, because it does not impact probability density functions $\mathcal{N}(\mu_\theta, \sigma_\theta)$ and $\mathcal{N}(\mu_\varphi, \sigma_\varphi)$, neither space intersection rate density function $P(x, d_\theta, d_\varphi)$.

The final ellipsoid used forward $E_D(x(t), v)$(eq. 6.28) is keeping all properties of ellipsoid $E(x(t), v)$ (eq. 6.28).

$$E_D(x(t), v) = \begin{cases} |\bar{E}_D(x(t), v)| = 0 & : \{x(t)\} \\ |\bar{E}_D(x(t), v)| \geq 0 & : \bar{E}_D(x(t), v) \end{cases} \tag{6.28}$$

The normal distribution condition for rate distribution function $P_D(x(t), d_\theta, d_\varphi, p)$, which is instance of to rate density function $P(x(y), d_\theta, d_\varphi)$ (eq. 6.20) is used. This rate distribution must be normalized according to (eq. 6.29).

$$\sum_{p \in E_D(x(t))} P_D(x(t), d_\theta, d_\varphi, p) = 1, \forall t \in \mathcal{R}^+ \tag{6.29}$$

The equations for *space intersection rate* are similar to (eq. 6.22, 6.23). For cell $c_{i,j,k}$ there exist intruder entry time $i_e(c_{i,j,k})$ its the earliest intersection with ellipsoid $E_D(x(i_e(c_{i,j,k}))), v$. Same situation occurs with intruder leave time $i_l(c_i, j, k)$. Because E_D is countable set, it means additional attributes can be attached to each point $p \in E_D$. Based on system dynamic (eq. 6.1) the *Time Of Arrival* (TOA) can be calculated. The example of TOA is given in fig. 6.5.

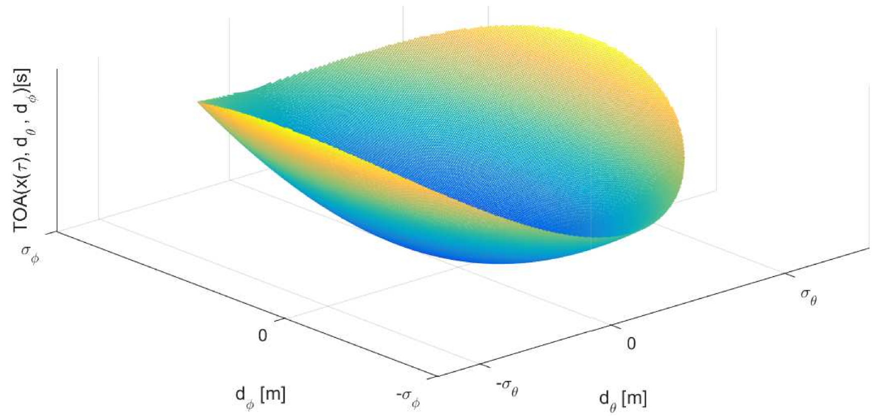

Figure 6.5: Time Of Arrival (TOA) for one ellipsoid $E_D(x(\tau), v)$.

The intersection rate $P_D(x(\tau), c_{i,j,k}, \theta, \varphi)$ for one time sample τ is given by (eq. 6.30), which has similar notation to (eq. 6.22), sums are used instead of integrals and discrete rate density function $P_D(x(\tau), d_\theta, d_\varphi, p)$ for points form ellipse and cell intersection are used as iterator base set $p \in \{E_D(x(\tau), v) \cap c_{i,j,k}\}$.

$$P_D(x(\tau), c_{i,j,k}, \theta, \varphi) = \sum_{p \in \left\{E_D(x(\tau), v) \cap c_{i,j,k}\right\}} P_D(x(\tau), d_\theta, d_\varphi, p) \tag{6.30}$$

The *space intersection rate* $P_{TD}(i_k(x_s, v, \theta, \varphi), c_{i,j,k})$ (eq. 6.31) is given as mean intersection rate of partial intersections $P_D(x(\tau), c_{i,j,k}, \theta, \varphi)$ where step set $T = \{\ i_e(c_{i,j,k}),\ \ldots,\ i_l(c_{i,j,k})\}$ contains all viable intersection times with ellipsoids $E(x(\tau \in T), v)$. The denominator is basically count of samples in sample time set T.

$$P_{TD}(i_k(x_s, v, \theta, \varphi), c_{i,j,k}) = \frac{\sum_{\tau=i_e(c_{i,j,k})}^{i_l(c_{i,j,k})} \sum_{p \in E_D(x(\tau),v)} P_D(x(\tau), c_{i,j,k}, \theta, \varphi, p)}{\sum_{\tau i_l(c_{i,j,k})}^{i_e(c_{i,j,k})} 1} \tag{6.31}$$

An *intersection of intruder cone and cell* $c_{i,j,k}$ cell is defined by (eq. 6.32) The set of point $p \in \mathcal{R}^3$ where condition of intersection between ellipsoids $E_D(x(\tau), v)$ for times $\tau \in \mathcal{R}^+$ and cell space $c_{i,j,k}$ is met.

$$\mathcal{P}(i_k(x_s, v, \theta, \varphi), c_{i,j,k}) = \bigcup_{\forall \tau \in \mathcal{R}^+} \left\{ p \in \mathcal{R}^3 : p \in c_{i,j,k} \cap E_D(x(\tau), v) \right\} \tag{6.32}$$

An *intruder time of entry* $i_e(i_k, c_{i,j,k})$ (eq. 6.33), for intruder i, k and cell $c_{i,j,k}$ is approximated for discrete point set $\mathcal{P}(i_k(x_s, v, \theta, \varphi), c_{i,j,k})$ (eq. 6.32) as minimal time of arrival $t_{TOA}(p)$ of member points p.

$$i_e(i_k, c_{i,j,k}) \approx \min \left\{ t_{TOA}(p) : p \in \mathcal{P}(i_k(x_s, v, \theta, \varphi), c_{i,j,k}) \right\} \tag{6.33}$$

An *intruder time of leave* $i_l(i_k, c_{i,j,k})$ (eq. 6.34), for intruder i, k and cell $c_{i,j,k}$ is approximated for discrete point set $\mathcal{P}(i_k(x_s, v, \theta, \varphi), c_{i,j,k})$ (eq. 6.32) as maximal time of arrival $t_{TOA}(p)$ of member points p.

$$i_l(i_k, c_{i,j,k}) \approx \max \left\{ t_{TOA}(p) : p \in \mathcal{P}(i_k(x_s, v, \theta, \varphi), c_{i,j,k}) \right\} \tag{6.34}$$

Combined intersection model: The *combined intersection model* $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$ is defined for intruder i_k with parameters:

1. *Starting position* x_s - expected position of intruder i_r in 3D space at time of avoidance t_i in avoidance grid frame $\mathcal{A}(t_i)$.

2. *Velocity vector* v - oriented velocity of intruder i_r at time of avoidance t_i in avoidance grid frame $\mathcal{A}(t_i)$.

3. *Horizontal uncertainty spread* θ - defines how much can intruder i_r deviate on horizontal axis of intruder local coordinate frame (if X+ is main axis, then Y is horizontal axis in right-hand euclidean coordinate frame), due the properties of intersection definition, the horizontal uncertainty spread can have following values $\theta \in [0, \pi/2]$.

4. *Vertical uncertainty spread* φ -defines how much can intruder i_r deviate on vertical axis of intruder local coordinate frame (if X+ is main axis in local right-hand euclidean intruder coordinate frame, then Z is horizontal vertical axis), due the intersection definition, the vertical uncertainty spread can have following values $\varphi \in [0, \pi/2]$.

5. *Body volume radius* r - defines the body volume of intruder in meters and it is having $\mathbb{R}^+$ value.

The *flag vector* $l, b, s, \tau \in \{0, 1\}$ is parametrization of rate calculation: l stands for *lined intersection*, b stands for *body intersection*, s stands for *spread intersection*, τ stands for *time account.*

The *space intersection for line* $P_L(i_k, c_{i,j,k})$ is defined as $P_T(i_k(x, v), c_{i,j,k})$, where i_k is intruder with properties of initial position x, velocity vector v and $c_{i,j,k}$ is target cell. (eq. 6.9).

The *space intersection rate for body volume* $P_B(i_k, c_{i,j,k})$ is defined as $P_T(i_k(x, v, r), c_{i,j,k})$ (eq. 6.13), where intruder i_r has additional property of the intruder body volume radius r.

The *space intersection probability for maneuverability uncertainty* $P_S(i_k, c_{i,j,k})$ is defined as $P_{TD}(i_k(x_s, v, \theta, \varphi), c_{i,j,k})$ (eq. 6.31), where intruder properties θ, φ stands for intruder horizontal and vertical uncertainty spread.

The *time intersection rate* $P_{\tau,x}(i_k, c_{i,j,k}) \in [0, 1]$ is defined in (eq. 6.5). This probability has two calculation modes, first is for 1D intersection (line), second is for volume intersection (body volume, spread elliptic cone).

UAS cell entry time t_e and cell leave time t_l time for vehicle in avoidance grid $\mathcal{A}(t_i)$ are given by (eq. 6.2) and (eq. 6.3).

Intruder leave and entry time for 1D intersections is trivial and is omitted in this section. Intruder entry i_e and intruder leave i_l for 3D intersection are given by (eq. 6.33, 6.34).

All partial rates with respective definition references are summarized in (eq. 6.35)

$$\begin{aligned}
P_L(i_k, c_{i,j,k}) &= P_T(i_k(x, v), c_{i,j,k}) && (6.9)\\
P_B(i_k, c_{i,j,k}) &= P_T(i_k(x, v, r), c_{i,j,k}) && (6.13)\\
P_S(i_k, c_{i,j,k}) &= P_{TD}(i_k(x_s, v, \theta, \varphi), c_{i,j,k}) && (6.31)\\
P_{\tau,x}(i_k, c_{i,j,k}) &= \frac{\|[i_e(c_{i,j,k}), i_l(c_{i,j,k})] \cap [t_e, t_l]\|}{\|[t_e, t_l]\|} && (6.5)
\end{aligned} \tag{6.35}$$

With definition of all space and time intersection rates (eq. 6.35) and given flag vector $l, b, s, \tau \in \{0, 1\}$ one can formulate combined intersection rate $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$ (eq. 6.36) for intruder i_k and cell $c_{i,j,k}$. The principle is following: *maximum of selected rates product based on flag vector is final intersection rate of intruder i_k in cell.*

The time-use flag τ is adding time intersection rate $P_{\tau,x}(i_k, c_{i,j,k})$, where time intersection rate is defined by $x = \{L, B, S\}$ for line, body volume, spread ellipse time intersections

$(P_{\tau,L}(i_k, c_{i,j,k}) \neq P_{\tau,B}(i_k, c_{i,j,k}) \neq P_{\tau,B}(i_k, c_{i,j,k})$ for one intruder i_k).

$$P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau) = \begin{cases} \tau = 0 & : \max \left\{ \begin{matrix} P_L(i_k, c_{i,j,k}).l \\ P_B(i_k, c_{i,j,k}).b \\ P_S(i_k, c_{i,j,k}).s \end{matrix} \right\} \\ \tau = 1 & : \max \left\{ \begin{matrix} P_{\tau,L}(i_k, c_{i,j,k}).P_L(i_k, c_{i,j,k}).l \\ P_{\tau,B}(i_k, c_{i,j,k}).P_B(i_k, c_{i,j,k}).b \\ P_{\tau,S}(i_k, c_{i,j,k}).P_S(i_k, c_{i,j,k}).s \end{matrix} \right\} \end{cases} \tag{6.36}$$

6.6.5 Moving Constraints

Idea: The basic ideas is the same as in case *static constraints* (sec. **??**). There is horizontal constraint and altitude constraint outlining the constrained space. The only additional concept is moving of *constraint* on horizontal plane in global coordinate system.

The constraint intersection with *avoidance grid* is done in *fixed decision Time*, for cell in *fixed cell leave time* (eq. 6.3), which means concept from static obstacles can be fully reused.

Definition: The *moving constraint definition* (eq. 6.37) covers minimal data scope for moving constraint, assuming linear constraint movement.

The original definition (eq. **??**) is enhanced with additional parameters to support constraint moving:

1. *Velocity* - velocity vector on 2D horizontal plane.
2. *Detection time* - the time when *constraint* was created/detected, this is the time when *c*enter and boundary points position were valid.

$$\begin{aligned} constraint = \{position, boundary, \dots \\ \dots, velocity, detectionTime, \dots \\ \dots altitude_{start}, altitude_{end}, safetyMargin\} \end{aligned} \tag{6.37}$$

Cell Intersection: The *intersection algorithm* follows (eq. **??**), only shift of the *center and boundary points* is required.

First let us introduce $\Delta time$ (eq. 6.38), which represents difference between the constraint detection time and expected cell leave time (eq. 6.3).

$$\Delta time = UAS_{leave}(cell_{i,j,k}) - detectionTime \tag{6.38}$$

The constraint boundary is shifted to:

$$shiftedBoundary(constraint) = \{newPoint = point + velocity \times \Delta time : \dots \\ \dots \forall point \in constraint.boundary\} \quad (6.39)$$

The constraint center is shifted to:

$$shiftedCenter(constraint) = constraint.center + velocity \quad (6.40)$$

Note. The $\Delta time$ is calculated separately for each $cell_{i,j,k}$, because *UAS* is also moving and reaching cells in different times. The *cell leave time* can be calculated in advance after reach set approximation.

Alternative Intersection Implementation: The alternative used for intersection selected based on polygon intersection algorithms review [4], the selected algorithm is *Shamos-Hoey* [5].

The implementation was tested on *Storm scenario* (sec. **??**) and it yelds same results.

Bibliography

[1] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[2] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Foundations of Computer Science, 1975., 16th Annual Symposium on*, pages 151–162. IEEE, 1975.

[3] Jon Louis Bentley, Bruce W Weide, and Andrew C Yao. Optimal expected-time algorithms for closest point problems. *ACM Transactions on Mathematical Software (TOMS)*, 6(4):563–580, 1980.

[4] Jon Louis Bentley and Thomas A Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on computers*, (9):643–647, 1979.

[5] Michael Ian Shamos and Dan Hoey. Geometric intersection problems. In *17th annual symposium on foundations of computer science*, pages 208–215. IEEE, 1976.

[6] Duncan Mclaren Young Sommerville. *Analytical geometry of three dimensions.* Cambridge University Press, 2016.