

Honeywell International - Prague Laboratory

# Probabilistic approach in data fusion for obstacle avoidance framework based on Reach sets

Research proposal

Alojz Gomola

**Honeywell**

Supervisors: Dr. Pavel Klang  
Dr. Jan Ludvik

July 10, 2017

# Contents

<b>1</b>	<b>Introduction to data fusion</b>	<b>6</b>
1.1	Related work . . . . .	6
1.2	Goals . . . . .	7
1.3	Assumptions . . . . .	7
<b>2</b>	<b>Data fusion problem formulation</b>	<b>9</b>
2.1	Deterministic approach shortcomings . . . . .	9
2.1.1	Changing scanning density of LiDAR . . . . .	9
2.2	Fusion of obstacle map with detected obstacles . . . . .	10
2.2.1	Adversary/Intruder probability spread . . . . .	11
2.2.2	Safety of passing trajectories . . . . .	12
2.2.3	Addressed issues summary . . . . .	12
2.3	Probabilities for trajectory and grid cell . . . . .	14
<b>3</b>	<b>State of art</b>	<b>15</b>
3.1	UAV motion model . . . . .	15
3.1.1	Continuous-time systems . . . . .	15
3.1.2	Discrete-time systems . . . . .	16
3.2	Reach sets . . . . .	16
3.3	Occupied space . . . . .	17
3.4	Movement automaton . . . . .	18
3.4.1	Prediction stability of Movement Automaton . . . . .	19
3.4.2	Lyapunov stability of Movement Automaton . . . . .	21
3.5	LiDAR . . . . .	22
3.6	Complements of algebra . . . . .	23
3.6.1	Transformation between coordinate systems . . . . .	24
3.6.2	Planar surface calculation . . . . .	25
<b>4</b>	<b>Probabilistic approach</b>	<b>27</b>
4.1	Simple plane model . . . . .	27
4.2	Model predictive control . . . . .	28
4.2.1	Model predictive control with movement automaton . . . . .	28
4.2.2	Predictor corrections . . . . .	29
4.3	Movement automaton predictor . . . . .	30
4.4	General algorithm . . . . .	32
4.5	Reduced reach set . . . . .	34
4.5.1	Constrained expansion procedure . . . . .	36
4.5.2	Chaotic reach set approximation . . . . .	36
4.5.3	Harmonic reach set approximation . . . . .	37

4.5.4	Combined reach set approximation . . . . .	38
<b>5</b>	<b>Probabilistic calculations</b>	<b>40</b>
5.1	Intruder intersection model . . . . .	40
5.2	Detected obstacle probability . . . . .	49
5.3	Visibility probability . . . . .	50
5.4	Map obstacle probability . . . . .	53
<b>6</b>	<b>Simulation Results</b>	<b>56</b>
6.1	Mission control . . . . .	56
6.2	Reduced reach set approximation computation . . . . .	57
6.3	Moving obstacles spread models . . . . .	59
6.3.1	Line intersection - Static . . . . .	59
6.3.2	Line intersection - Timed . . . . .	60
6.3.3	Line intersection - Future movements . . . . .	60
6.3.4	Vehicle body volume - timed . . . . .	61
6.3.5	Uncertainty spread - Static . . . . .	63
6.3.6	Uncertainty spread - Timed . . . . .	65
6.4	Static obstacles avoidance . . . . .	65
6.4.1	Probabilistic model evaluation . . . . .	70
6.4.2	Avoidance performance . . . . .	73
6.5	Intruders avoidance . . . . .	76
6.5.1	Intruder avoidance evaluation . . . . .	79
<b>7</b>	<b>Conclusion and Future work</b>	<b>84</b>

# List of Figures

2.1	Different count of LiDAR hits on objects with same size but different distance from vehicle . . . . .	10
2.2	Overshadowed map obstacle by detected obstacles . . . . .	11
2.3	Intruder vehicle collision probability along assumed trajectory with uncertain spread. . . . .	12
2.4	Trajectories leading to same destination with different safety ratings. . . . .	13
3.1	Six spaces of interest [? ] . . . . .	23
3.2	Planar surface calculation notation and plot . . . . .	26
4.1	Trajectories for chaotic reach set approximation method . . . . .	37
4.2	Trajectories for harmonic reach set approximation method . . . . .	38
4.3	Trajectories for combined reach set approximation method . . . . .	39
5.1	One probable position $[d_d, d_\theta, d_\varphi]$ (green). deviated from linear trajectory (blue line) at point $\vec{x}(10)$ .(blue) with initial position $x_s$ (red) . . . . .	43
5.2	Probability of intruder $i_k$ position in ellipsoid $E(\vec{x}(t), \vec{v})$ . . . . .	44
5.3	Avoidance grid $\mathcal{A}(t_i)$ (blue) intersection with elliptic cone intruder $i_k(\vec{x}, \vec{v}, \theta, \varphi)$ (red) example. . . . .	45
5.4	Time Of Arrival (TOA) for one ellipsoid $E_D(\vec{x}(\tau), \vec{v})$ . . . . .	46
5.5	Visibility probability for cell row - free space . . . . .	51
5.6	Visibility probability for cell row - first obstacle . . . . .	52
5.7	Visibility probability for cell row - second obstacle . . . . .	52
5.8	Visibility probability for cell row - third obstacle . . . . .	53
5.9	Map obstacle - undetected by sensory system . . . . .	53
5.10	Map obstacle - detected by sensory system . . . . .	54
5.11	Map obstacle - hidden behind detected obstacle . . . . .	54
6.1	Example of partial movement buffers chaining. . . . .	57
6.2	Obstacle space for static line intersection . . . . .	59
6.3	Reach set approximation for static line intersection - middle layer . . . . .	59
6.4	Obstacle space for timed line intersection . . . . .	60
6.5	Reach set approximation for timed line intersection - middle layer . . . . .	60
6.6	Obstacle space for timed line intersection with future movement . . . . .	61
6.7	Reach set approximation for timed line intersection with future movement - middle layer . . . . .	61
6.8	Obstacle space for timed intruder body volume . . . . .	62
6.9	Reach set approximation for timed vehicle body volume - all layers . . . . .	63
6.10	Obstacle space for static uncertainty spread . . . . .	64
6.11	Reach set approximation for static uncertainty spread - all layers . . . . .	64

6.12	Obstacle space for static uncertainty spread . . . . .	65
6.13	Reach set approximation for static uncertainty spread - all layers . . . . .	65
6.14	First obstacle set $\mathcal{B}_0(1)$ avoidance . . . . .	68
6.15	Second obstacle set $\mathcal{B}_0(2)$ avoidance . . . . .	68
6.16	Third obstacle set $\mathcal{B}_0(3)$ avoidance . . . . .	69
6.17	Vehicle trajectory after mission competition . . . . .	70
6.18	Reachability probability assessment for $\mathcal{B}_0(1)$ avoidance . . . . .	71
6.19	Obstacle probability assessment for $\mathcal{B}_0(1)$ avoidance . . . . .	72
6.20	Visibility probability assessment for $\mathcal{B}_0(1)$ avoidance . . . . .	73
6.21	Crash distance $c_r$ evolution during mission . . . . .	74
6.22	Trajectory tracking performance . . . . .	75
6.23	First intruder detection. . . . .	78
6.24	After all intruders avoidance . . . . .	79
6.25	Obstacle probability after first intruder detection . . . . .	80
6.26	Reachibility probability after first intruder detection . . . . .	81
6.27	Visibility probability after first intruder detection . . . . .	82
6.28	Intruder collision distance $c_d(t)$ evolution. . . . .	83
7.1	Decision frame $\mathcal{D}(0)$ . . . . .	85

# List of Tables

4.1	Base values for movement application, vehicle position difference $x_v, y_v, z_v$ and orientation differences $\alpha_v, \beta_v, \gamma_v$ . . . . .	30
4.2	Base values for movement application, vehicle position difference $x_v, y_v, z_v$ and orientation differences $\alpha_v, \beta_v, \gamma_v$ . . . . .	31
6.1	Reduced reach set computation methods performance . . . . .	58
6.2	Intruder set $\mathcal{I}$ . . . . .	77

# Chapter 1

## Introduction to data fusion

Need for abstract representation of *operative space* independent of used sensed technology arisen from commercial side of UAV users. The universal obstacle avoidance system should have *portability property*. Our previous work *Obstacle avoidance framework based on reach sets* have introduced important concept of obstacle avoidance framework with control interface. *Control interface* has been implemented as *movement automaton MA*, which gave us control interface portability. The original concept was using simple thresholding to cell status interpretation, which was hardwired to LiDAR technology. The new demand is to incorporate concepts of *visibility*, *intruder* and *multiple obstacle sources* into obstacle avoidance concept. The probabilistic approach, when the state of cells in avoidance is changed into set of ratings is introduced into this work. The key features introduced in this work are:

1. *Data fusion interface* - interface to fuse sense data from various online, offline, cooperative, non-cooperative sources.
2. *Intruder concept* - concept of moving obstacle and its uncertainty spread mapping into avoidance grid.

### 1.1 Related work

Avionics sensor fusion has been proposed by Ramsay in [? ]. Next generation avoidance concept [? ] is introducing concept of higher level sensor fusion called *data fusion*. The uncertainty of remotely piloted systems have been discussed in [? ]. The work provided concept of various performance ratings like visibility and obstacle rating, more details have been given in [? ]. This ratings were modeled only for operator decision making [? ], results are usable for automated decision making and space assessment. *Probabilistic trajectory assessment* has been firstly proposed in [? ] where trajectory was tracking *safety measurements* along. Probabilistic approach from view of game theory was firstly used in [? ]. Probabilistic path planning using safety zones similar to cell classification of this work have been used in [? ]. Probabilistic path search similar to our reach set representation using rapidly exploring path trees have been used in [? ? ]. Relationship between classic grid search and probabilistic lattice search have been established in [? ]. A probabilistic approach for trajectory estimation via reduced lattice search is known from 1986 from work of Gessel [? ] lattice paths were enumerated via movement sequences and similar technique is used in our reach set estimation method using movement automaton.

Overall concepts of probabilistic sets have been given by Hirota in [? ]. Free flight safety rating similar to our reachability concept have been presented in [? ].

## 1.2 Goals

Main goal is to develop a conceptual approach data fusion and create interface to sensor system, so given obstacle avoidance approach given by our previous work can be both *platform and sensor system* independent. The independence should be solved via interface feed where sensor reading will be represented as *space portion and trajectory point* probabilities. To achieve this goal following sub-goals must be achieved:

1. *Define universal interface for vehicle sensors* - identify probabilities bounded to trajectory  $\mathcal{T}(x_0, B)$  or visibility grid cell  $c_{i,j,k} \in \mathcal{A}(t_i)$  and define their meaning independent on sensor equipment on vehicle.
2. *Derive probabilistic model for avoidance grid* - for identified probabilities define abstract calculation mechanism to derive final values for given bounded to trajectory  $\mathcal{T}(x_0, B)$  or visibility grid cell  $c_{i,j,k} \in \mathcal{A}(t_i)$ .
3. *Propose data fusion procedure for ADS-B and LiDAR with obstacle MAP* - define data fusion procedure for identified probabilities as extension of abstract model.
4. *Compare probabilistic and deterministic model performance* - compare probabilistic and deterministic approach performance in common area of *static obstacle avoidance*. Choose demonstrative example of mission with static obstacles and compare *Crash distance* function performance.

## 1.3 Assumptions

The precision of *obstacle map* must be sufficient for intersections with avoidance grid cells. Intersection with grid cells is mandatory for map obstacle inclusion into space assessment algorithm. This condition is summarized in assumption 1.

**Assumption 1.** *Obstacle map is given with sufficient precision*

*Data fusion* is main concern of this work. *Sensor fusion* approach how to filter and estimate various positional parameters of intruders or static obstacles have been summarized in book by *Frederik Gustafsson: Statistical Sensor Fusion* [? ]. The assumption that readings from sensory system are pre-processed up to best possible quality is summarized in assumption 2.

**Assumption 2.** *Sensory readings are filtered and processed via sensor fusion*

The moving intruders are new addition to the *avoidance grid* concept, therefore it is necessary to start with the simplest conditions possible. The standard intruder model is given by position and velocity vector at detection time. For this case there will be no position nor velocity vector update and the intruder will have tolerated conic deviation with horizontal and vertical parameters. This assumptions about moving intruders is summarized in assumption 3.

**Assumption 3.** *Moving intruders velocity and position vector are given with known noise and are static during intruder life time, the maximal horizontal and vertical deviation from trajectory is given.*

When intruder is detected with given position, velocity and spread, there is at least one avoidance trajectory to avoid intruder. This is given by assumption 4.

**Assumption 4.** *Moving intruders are avoidable with existing vehicle maneuverability*

The given assumptions 1-4 are still far away from non segregated airspace, but more relaxed than assumptions for *deterministic approach*.

# Chapter 2

## Data fusion problem formulation

For each point  $\vec{p} \in \mathbb{R}^3$  in space there is set of questions which can be asked:

1. *Is point visible ?* - how good can be point or space partition observed during vehicle flight before decision.
2. *Is there obstacle ?* - how sure I can be that there is observation of obstacle.
3. *Is point reachable ?* - how is maximum safety of any route leading to given point.

To address these issues it is necessary to identify related ratings and bound them to space representation artifacts. In this case we have two space representation merged together in one avoidance grid  $\mathcal{A}(t_i)$ :

1. *Cells*  $c_{i,j,k}$  belongs to planar space  $c_{i,j,k} \in \mathbb{R}^3$  and represents a set of points bounded by cell, the ratings bounded to cells are representing *visible space assessment*.
2. *Trajectories*  $\mathcal{T}(\vec{x}_0, B)$  belongs to vehicle system state space  $\mathcal{T}(\vec{x}_0, B) \in \mathbb{R}^n$ , where  $n \in \mathbb{N}^{3+}$  and there exist projection function  $\mathcal{P} : \mathbb{R}^n \rightarrow \mathbb{R}^3$  which projects vehicle trajectory  $\mathcal{T}(\vec{x}_0, B)$  to visible space  $\mathbb{R}^3$ , the ratings bounded to trajectories are representing *state space assessment*

*State space assessment* and *visible space assessment* are interconnected and the ratings are impacting each other between spaces. The final value of the particular rating is given by *general algorithm*.

### 2.1 Deterministic approach shortcomings

This subsection contains identified shortcomings of *deterministic approach*. The identified shortcomings varies trough different topics and their impact is reviewed in terms of probabilistic approach.

#### 2.1.1 Changing scanning density of LiDAR

LiDAR is scanning in conic section given by  $d_r, \theta_r, \varphi_r$ , where  $d_r$  is distance range,  $\theta = [s, e], s, e \in [0, 2\pi]$  is horizontal rotation range, and  $\varphi = [s, e]s, e \in [0, \pi/2]$  is vertical offset range. Let say that  $d\theta, d\varphi$  is unitary angle offset in which one LiDAR send and return is executed. The surface of area given by some distance  $d$ , and unitary offsets  $d\theta, d\varphi$  is

changing with  $d$ . The area of thresholding object surface is not changing. This fact has an impact on count of the hits on some surface. The example is given in fig. 2.1 where we have two identical objects (red circle) in distances 5 and 10 meters. The closer object consumes 5 LiDAR beam hits and the farther object consumes only 3 LiDAR beam hits. The probability of obstacle encounter is remaining the same for closer and farther object. *Probabilistic approach must handle this issue* and return the same probability of obstacle collision for objects with same scanned surface (with different LiDAR beam hit count).

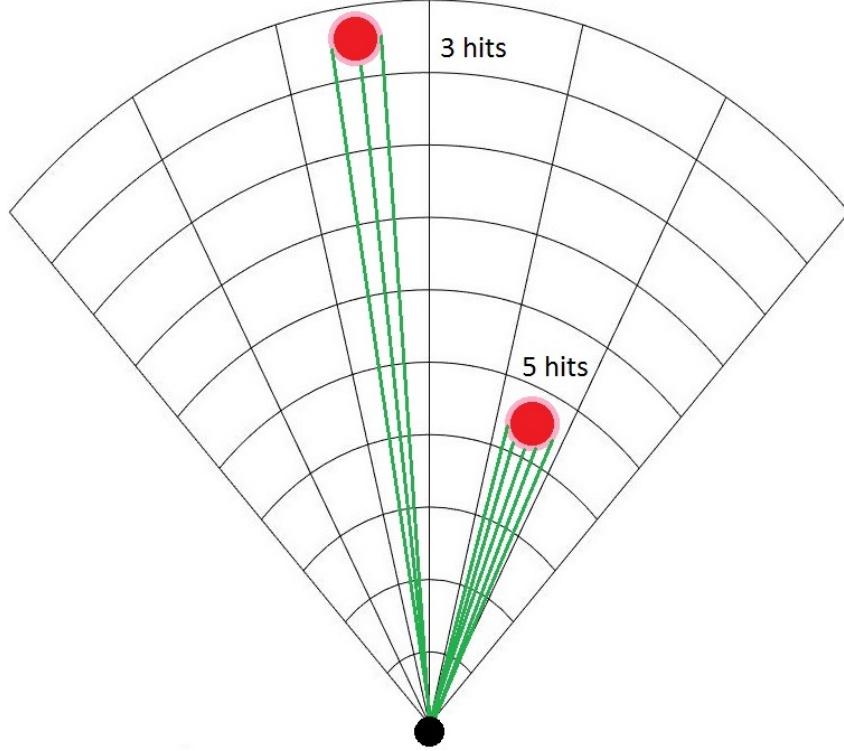


Figure 2.1: Different count of LiDAR hits on objects with same size but different distance from vehicle

## 2.2 Fusion of obstacle map with detected obstacles

The concept of *offline/online obstacle map* is mandatory in modern obstacle avoidance systems and increases the safety of trajectory planning. *Deterministic* concept was considering only LiDAR reading or *real-time reading* in general. The fusion of real time reading and obstacle map (prior knowledge) is required. Data fusion of these two sources is strongly depending on visibility property, because there are three basic scenarios:

1. *Dual detection* - the obstacle is marked on the map and detected by sensory system at some point of the time (deterministic concept works).
2. *Hindered vision* - the detected obstacles are hindering vision to map obstacle therefore map obstacle uncertainty arises (deterministic concept fails).
3. *False-positive map* - map obstacle occupied space is visible by sensory system, but negative detection is returned. Therefore the map is giving *false-positive* information.

The second case is given in fig. 2.2, where map obstacle (blue circle) is overshadowed by three scanned obstacles (red circle). The visible space is denoted by green fill, the invisible space is denoted by gray fill.

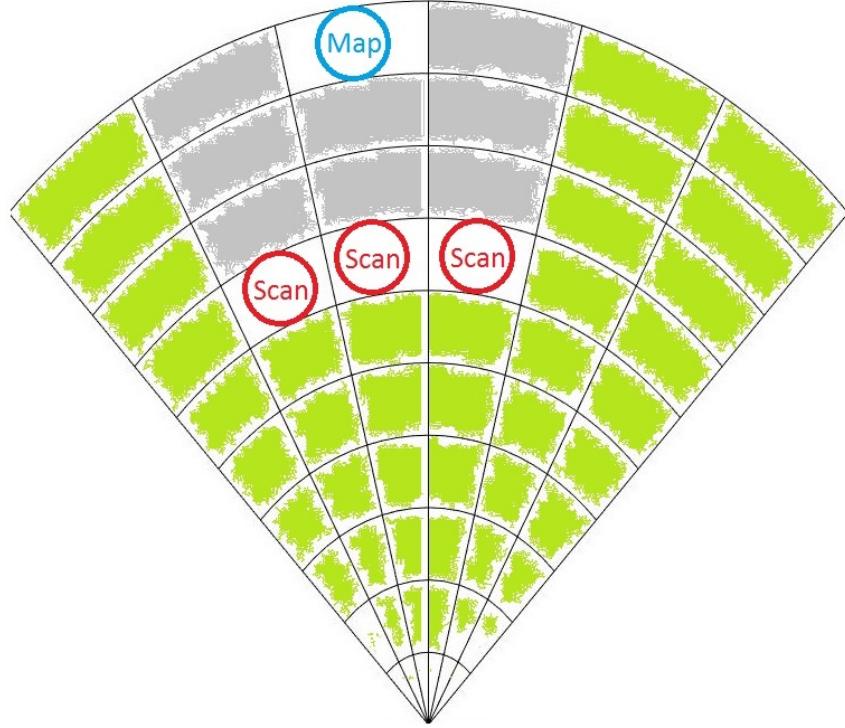


Figure 2.2: Overshadowed map obstacle by detected obstacles

### 2.2.1 Adversary/Intruder probability spread

*Adversarial behaviour* is first incorporation of moving obstacle. The Adversary is trying to destroy avoiding vehicle. Intruder concept will be used instead of adversary concept in this work. The *intruder* vehicle (for short intruder) is in non-cooperative mode and is not trying to hurt our vehicle. The non-cooperative premise is given by current standard behaviour of small UAV vehicles. *Minimal data set* which can extracted for any intruder consist from:

1. *Position*  $\vec{p} \in \mathbb{R}^3$  - position of intruder in any transferable vehicle coordinate frame can be extracted and converted into *avoidance coordinate frame*
2. *Heading (velocity)*  $\vec{v} \in \mathbb{R}^3$  - heading with time parameter or velocity of intruder can be read or estimated trough various means. It must be converted into *avoidance coordinate frame*.
3. *Uncertainty spreads*  $\theta, \varphi, \dots$  - various spreads of uncertain intruder maneuverability.

Example of intruder intersection without *time of intersection* consideration is given in fig. 2.3. The intruder initial position  $\vec{p}$  at time of avoidance  $t_i$  is denoted as green circle with 'A' letter. The intruder linear path approximation based on velocity  $\vec{v}$  is denoted as middle green line. The spread of intruder maneuverability is given as outer green lines. The probability of intersection is denoted as green-orange-red cell fill, meaning is the red

is high probability of intersection, orange is medium probability of intersection and green is low probability of intersection.

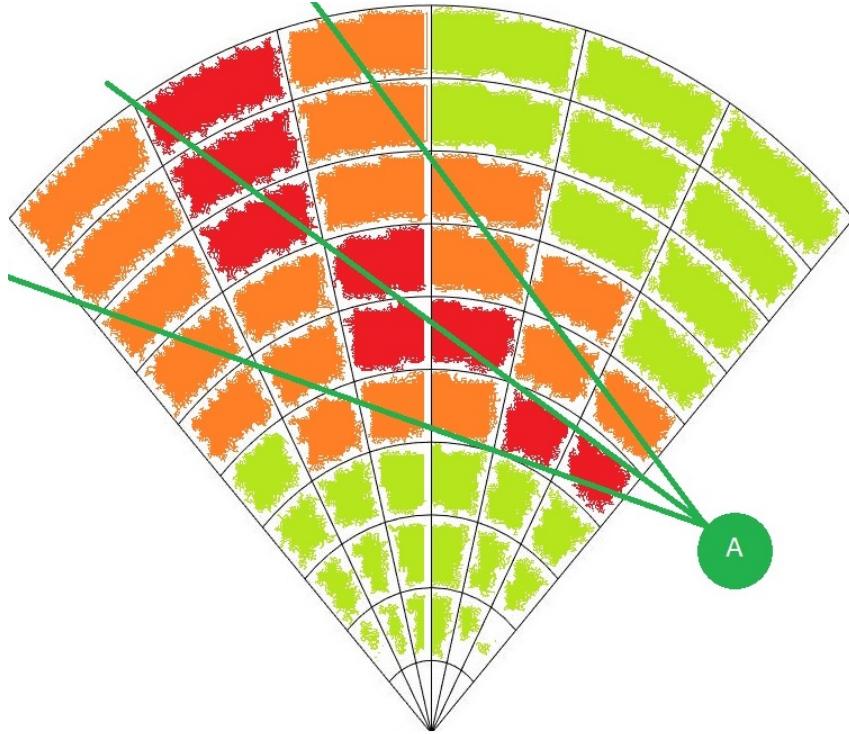


Figure 2.3: Intruder vehicle collision probability along assumed trajectory with uncertain spread.

### 2.2.2 Safety of passing trajectories

Safety of passing trajectories is given by combined safety of passing cells. Each trajectory  $\mathcal{T}(\vec{x}, B)$  incorporated in reach set approximation  $\mathcal{R}(\vec{x}, t_i, t_{i+1})$  is passing through ordered set of cells  $\mathcal{C} \subset \mathcal{A}(t_i)$ . The base safety rating of single cell  $c_{i,j,k}\mathcal{C}$  can be expressed as product of cell visibility rating and inverted cell obstacle rating. Simply if cell is visible and probability of obstacle is low, cell is safe. *Safe trajectory* is trajectory which have safe space along the path. The example of trajectories safety is given in fig. 2.4. The safety ratings of cell are given by red-orange-pink-green fills, where red denotes very low safety, orange denotes low safety, pink denotes medium safety, green denotes high safety. Blue trajectory is passing through many unsafe cells and therefore overall safety is dangerous. Yellow trajectory is passing through few cells with low and medium safety, therefore the choice of this trajectory is questionable. Green trajectory is passing only through one cell with medium (pink) cell otherwise it passes through all very safe cells, therefore its safe to chose this trajectory. The obstacle which caused such cell safety assessment is denoted as black filled circle with 'O' markings.

### 2.2.3 Addressed issues summary

Following issues have been extracted of previously mentioned situations. they have been generalized in context of previous *deterministic* approach, offline obstacle map, and intruder behaviour model.

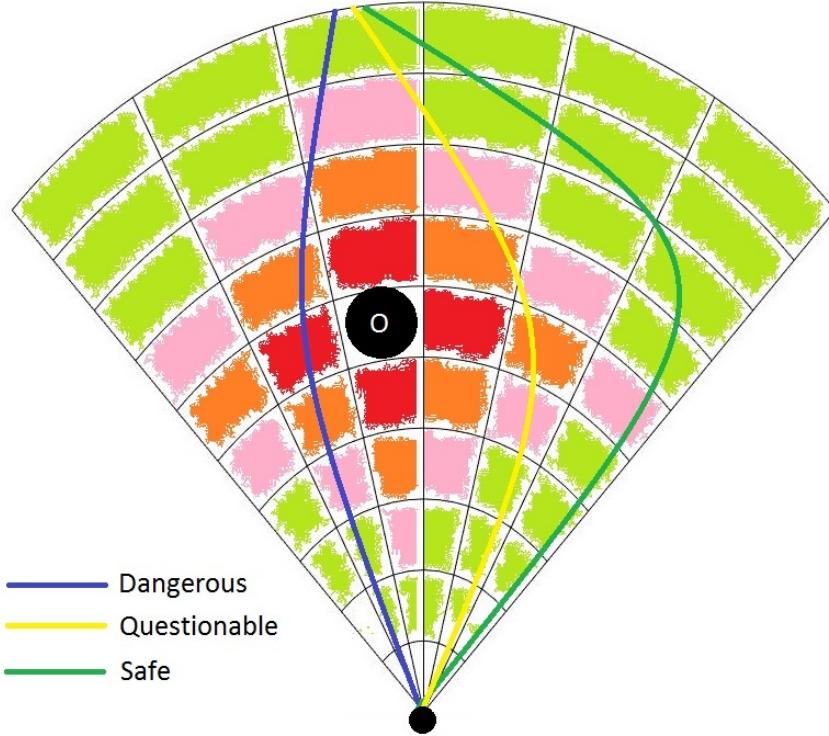


Figure 2.4: Trajectories leading to same destination with different safety ratings.

1. *Model for intruder passing* - intruder can pass through vehicle's FOV at different times with different probability. There are multiple intruder behaviours which have not been accounted in previous approach, like intruder speed and heading uncertainty, intruder own maneuverability etc ...
2. *Multiple obstacle sources* - concept of visibility is mandatory for map and sensor reading obstacle fusion. The introduction of visibility needs to be done prior multiple source data fusion.
3. *Reachability differentiation* - some cells in avoidance grid  $\mathcal{A}(t_i)$  are more reachable than others, for example cell with one feasible trajectory has same reachability than cell with multiple leading trajectory.
4. *Decision making support* - some trajectories leading to same goal have different execution cost and different safety rating, this can be used as additional decision factor. For example framework can choose more expensive but safer trajectory to avoid.
5. *Obstacle size impact* - obstacle size can have impact on cell safety rating, in deterministic approach cell with one small obstacle have same safety rating than cell with big obstacle.

## 2.3 Probabilities for trajectory and grid cell

The preliminary analysis of scenarios and identified behaviours and impacts of probabilistic approach gives us an opportunity to identify specific track-able ratings for cell and trajectory artifacts. Each cell  $c_{i,j,k}$  at avoidance time  $t_i$  must have assessed this minimal set of probabilities:

1. *Obstacle probability*  $P_O$  - determines the probability of obstacle encounter in given cell, obstacle probability is combination of various sources. Initial value of obstacle probability  $P_O$  is assumed as 100 %.
2. *Visibility*  $P_V$  -determines visibility of cell at time of avoidance  $t_i$ , portion of sight into cell space can be hindered by obstacles in front of cell.
3. *Reachability*  $P_R$  - determines the maximum reachability like-hood for given cell based on passing trajectories.

Each trajectory  $\mathcal{T}(x_0, B) \in \mathcal{R}(t_0, t_1, x_0)$  for each trajectory point  $p_{\mathcal{T}} \in \{\mathcal{T}(x_0, B) \rightarrow \mathbb{R}^3\}$  must have assessed this minimal set of properties:

1. *Reachability*  $P_R$  - determines reachability rating of given point  $p_{\mathcal{T}}$ .
2. *Feasibility of decision*  $P_D$  - determines next decision feasibility rating.

# Chapter 3

## State of art

Cooperative and non cooperative Sense and Avoid (SAA) systems are key enablers for Unmanned Aircraft (UAV) to routinely access non-segregated airspace [? ]. Both cooperative and non-cooperative SAA systems are being developed to address this integration requirement. The SAA capability is defined as the automatic detection of possible conflicts by the UAV platform under consideration and performing avoidance maneuver tasks to prevent the identified collisions. An analysis of the available SAA candidate technologies and the associated sensors for both cooperative and non-cooperative SAA systems is presented in [? ]. Non-cooperative Collision Detection and Resolution (CD&R) for UAV is considered as one of the major challenges that needs to be addressed [? ] for the insertion of UAVs in non-segregated air space. As a result, a number of non-cooperative sensors for the SAA system have been adopted. Light Detection and Ranging (LIDAR) is used for detecting, warning and avoiding obstacles for low-level flying [? ].

An approach to the definition of encounter models and their applications to SAA strategies is presented in [? ] for both cooperative and non-cooperative scenarios.

Since 2014, there is a visible strong political support for developing rules on drones but regulations are not harmonized yet. The European Aviation Safety Agency (EASA) has been tasked to develop a regulatory framework for drone operations and proposals for the regulation of "low-risk" UAV operations. In achieving this, EASA is working closely with the Joint Authorities for Regulation of Unmanned Systems (JARUS) [? ].

### 3.1 UAV motion model

This section strongly follows [? ].

#### 3.1.1 Continuous-time systems

Consider a class of systems given by functions:

$$\begin{aligned} S : \vec{u}(t) &\rightarrow \vec{x}(\vec{x}_0, t) \\ \vec{u}(t) &: [0, T] \rightarrow \mathbb{R}^p \\ \vec{u}(t) &\in \mathbb{R}^p, \vec{x}(t) \in \mathbb{R}^n \end{aligned} \tag{3.1}$$

where  $\vec{u}(t)$  and  $\vec{x}(\vec{x}_0, t)$  are sets of continuous-time signals. These are often called continuous-time systems because they operate on continuous-time signals. Frequently, such systems can be defined by differential equations that relate the input signal to the

output signal. A prototypical description of a controlled (there is a control input signal) continuous-time system is:

$$\dot{x}(t) = f(t, x(t), u(t)), u(t) \in U(t) \quad (3.2)$$

where  $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$  satisfies the conditions for existence and uniqueness of the ordinary differential equation and  $u$  is our control[? ].

### 3.1.2 Discrete-time systems

Consider another class of systems given by functions

$$\begin{aligned} S : \vec{u}(k) &\rightarrow \vec{x}(k), \\ k &\in \{0, t_s, 2.t_s, 3.t_s, \dots i.t_s\}, i \in \mathbb{N}^+ \\ \vec{u}(k) &\in \mathbb{R}^p, \vec{x}(k) \in \mathbb{R}^n \end{aligned} \quad (3.3)$$

where  $\vec{u}(k), \vec{x}(k)$  is a set of discrete-time signals. They can be represented by a function  $f$  like  $f : \{0, t_s, 2.t_s, 3.t_s, \dots i.t_s\} \rightarrow \mathbb{R}^n, i \in \mathbb{N}^+$  where  $t_s$  is sampling time and  $i$  is discrete step [? ].

## 3.2 Reach sets

The reach set of a system described by a differential equation is the set of all states that can be reached from an initial state within a given time interval. For general case consider the system described by equation (3.2).

**Definition 1** (Reach set starting at a given point). *Suppose the initial position and time  $(\vec{x}_0, t_0)$  are given. The reach set  $\mathcal{R}[\tau; t_0, \vec{x}_0]$  of system (3.2) at time  $\tau \geq t_0$ , starting at position and time  $(\vec{x}_0, t_0)$  is given by:*

$$\mathcal{R}[\tau, t_0, \vec{x}_0] = \bigcup \{\vec{x}(\tau) : \vec{u}(s) \in U(s), s \in (t_0, \tau]\} \quad (3.4)$$

Reach set starting at given set can be used to determine reach set in case of hybrid system input control switch and it is defined as follow:

**Definition 2** (Reach set starting at a given set). *The reach set at time  $\tau > t_0$  starting from set  $X_0$  is defined as:*

$$\mathcal{R}[\tau, t_0, X_0] = \bigcup \{\mathcal{R}[\tau, t_0, \vec{x}_0] : \vec{x}_0 \in X_0\} \quad (3.5)$$

Reach set for adversarial behavior can be used to calculate possible escape routes from pursuer and it is defined as follow:

**Definition 3** (Reach set under adversarial behavior). *Consider now the case of adversarial behavior, for system  $\dot{x} = u, u \in \mathbb{B}$ . where  $u(t)$  is our control and  $v(t)$  is adversary control which is independent of  $u(t)$ , let  $w(t) = u(t) - \arg_{v(t) \in V(t)} \sup_{x \in x(t)} v(t)$ , which represents worst possible input change in given state and time, then reach set for system is represented as:*

$$\mathcal{R}[\tau; t_0, \vec{x}_0] = \bigcup \{\vec{x}(\tau) : \vec{w}(s) \in W(s), s \in (t_0, \tau]\} \quad (3.6)$$

Reach set under constraints are usable to define state constrained systems in terms of dynamics and technical capabilities.

**Definition 4** (Reach set under state constraints). *Suppose the initial position and time  $(\vec{x}_0, t_0)$  and  $x$  constraints are given  $x(t) \in \mathbb{A} \subset \mathbb{R}^n, \dot{x}(t) \in \mathbb{B} \subset \mathbb{R}^n$ . The reach set  $\mathcal{R}[\tau, t_0, \vec{x}_0]$  of system (3.2) at time  $\tau \geq t_0$ , starting at position and time  $(\vec{x}_0, t_0)$  is given by:*

$$\mathcal{R}[\tau, t_0, \vec{x}_0] = \bigcup \{\vec{x}(\tau) : \forall s \in (t_0, \tau], x(s) \in \mathbb{A}, \dot{x}(s) \in \mathbb{B}, \exists u(s) \in U(s)\} \quad (3.7)$$

### 3.3 Occupied space

Occupied space representation is crucial in obstacle avoidance, this section introduces models and notations. Analytical geometry structures of sphere and ellipsoid are used to determine closed occupied spaces.

**Definition 5.** Unit sphere (Unit ball)  $\mathcal{B}(\vec{p}, r)$  denotes occupied space in point  $\vec{p} = [x_p, y_p, z_p]^T$  with radius  $r$ . Where point  $\vec{b} = [x_b, y_b, z_b]$  belongs to sphere  $\mathcal{B}(\vec{p}, r)$  if and only if:

$$(x_b - x_p)^2 + (y_b - y_p)^2 + (z_b - z_p)^2 \leq r \quad (3.8)$$

Definition of sphere  $\mathcal{B}(\vec{p}, r)$  is usually used to denote safety margin of vehicle  $s_m$ , where  $s_m$  represents maximum radius to vehicle matter point from vehicle mass center.

**Definition 6.** Spherical coating  $\mathcal{C}(\vec{f}(\cdot), r)$  denotes occupied space of object which surface can be approximated by function  $f(\cdot)$ , with points of surface  $\vec{p} = [x_p, y_p, z_p]^T \in \mathbb{R}^3$  and object inner points  $\vec{i} = [x_i, y_i, z_i]^T \in \mathbb{R}^3$ . Then spherical coating  $\mathcal{C}(\vec{f}(\cdot), r)$  is defined as follows:

$$\mathcal{C}(\vec{f}(\cdot), r) = \left\{ \vec{b} \in \mathbb{R}^3 : \vec{b} \in \bigcup_{\forall \vec{p} \in \vec{f}(\cdot)} \mathcal{B}(\vec{p}, r), \#\vec{i} = \vec{b} \right\} \quad (3.9)$$

One can say that spherical coating is a closed, not compact set of points  $\vec{b} = [x_b, y_b, z_b]$  where distance to closest surface point  $\vec{p}$  is lesser or equal to coating radius  $r$ .

Calculation of spherical coating can be time consuming especially when surface function  $\vec{f}(\cdot)$  is not smooth [? ]. Closest point problem has been formulated by Shamos [? ]. Closest point estimation with moving vehicle and with thick data flow can be solved by time optimal and deterministic approach, one of them have been presented by Bentley in [? ]. Approach is based on closest point search in local planar coordinates which may be ideal for steam-line LiDAR data. Estimation of spherical coating is used in potential field avoidance methods. One most notable study was using spherical approximation based on obstacle center and mass distribution in space [? ]. Potential field have their limitations in partially known environment, because it is hard to determine center of mass and mass distribution of obstacle from partial information [? ]. Main source of object proportional estimation can be reused from camera based solutions like [? ]. Other possible solution sources from fast clustering algorithms from Geographical Information Systems (GIS) for example [? ].

### 3.4 Movement automaton

Movement automaton used as proxy between discrete command chain (movement chain) and control signal ( $u(t)$ ). Movement automation is based on hybrid automaton.

**Definition 7.** Hybrid automaton  $\mathcal{H}(Q, \mathbb{R}^n, f, \varphi, \rho)$  is well defined system representation consisting from following components.

$$\begin{aligned} Q &\equiv \text{set of discrete states} \\ \mathbb{R}^n &\equiv \text{continuous state-space} \\ f : Q \times \mathbb{R}^n &\rightarrow \mathbb{R}^n \equiv \text{vector field} \\ \varphi : Q \times \mathbb{R}^n &\rightarrow Q \equiv \text{discrete transition} \\ \rho : Q \times \mathbb{R}^n &\rightarrow \mathbb{R}^n \equiv \text{reset map} \end{aligned} \tag{3.10}$$

Discrete state  $q_i \in Q$  represents system current state and impacts system behavioural equation and state. Continuous state  $[x(t), u(t)] \in \mathbb{R}^n$  represents system state from physical viewpoint (values of state variables and inputs in continuous space  $\mathbb{R}^n$ ). Vector field  $f$  assign to each discrete state  $q_i \in Q$  system behavioral function  $\dot{x} = f(x, u)$ . Discrete transition  $\varphi$  defines conditions to transit between two different states. Reset map  $\rho$  defines system state or input change on reset conditions.

**Definition 8.** Trajectory primitive  $\hat{t}(x_0, t_0, t_1)$  is defined on given time interval  $(t_0, t_1]$  for system  $\dot{x} = f(x, u)$  with initial state  $x_0$  at  $t_0$  and final state  $x_1$  at  $t_1$  as follow:

$$\hat{t}(x_0, t_1, t_1) = \{x \in \mathbb{R}^n : x = \Phi(t_0, \tau, x_0), \tau \in (t_0, t_1]\} \tag{3.11}$$

Trajectory primitive can be viewed as ordered set of system trajectory positions, this set is infinite continuous and flat.

**Definition 9.** Trajectory equivalence. Let  $\mathcal{T}$  be trajectory defined as ordered sequence of countable trajectory primitives:

$$\mathcal{T} = \left\{ \bigcup_{i=0}^n \hat{t}_i(x_i, t_i, t_{i+1}) \right\} \tag{3.12}$$

Trajectory  $\mathcal{T}$  is time independent and contains at least one trajectory primitive. Trajectory  $\mathcal{T}_\alpha$  consist from  $m \geq 1$  trajectory primitives and its smooth and continuous. Trajectory  $\mathcal{T}_\beta$  consist from  $n \neq m$  primitives and its smooth and continuous. Trajectories  $\mathcal{T}_\alpha \equiv \mathcal{T}_\beta$  if and only if:

$$\forall x_i \in \mathcal{T}_\alpha \exists y_i \in \mathcal{T}_\beta : x_{i+1} \equiv y_{i+1}, i \in 0 \dots k \tag{3.13}$$

Where  $x_i$  and  $y_i$  is  $i$ -th point of trajectories in state space  $\mathbb{R}^n$ .

Trajectory primitives and trajectory equivalence have been defined, therefore movement primitives and movement can be defined.

**Definition 10.** Movement primitive  $p$  for system  $\dot{x} = f(x, u)$  and time interval  $(t_i, t_{i+1}]$  there is defined continuous input signal  $u(t)$ .

$$p_i = u(t), t \in (t_i, t_{i+1}] \tag{3.14}$$

Movement primitives can be chained to give smooth input signal. if there is movement primitive  $p_1$  and movement primitive  $p_2$  they can be chained on time  $\tau$  when  $u_1(\tau) = u_2(\tau)$ , therefore system state can be also chained  $x_1(\tau) = x_2(\tau)$ .

**Definition 11.** Movement  $m(t)$  is defined as chain of movement primitives  $\{p_1, p_2, \dots, p_n\}$ . Movement is smooth function with existing derivation.

Movement is main building block of movement automaton, movement is defined by movement type and its duration, movement switching is possible when execution time allows movement chaining. Movements can be separated into two categories:

1. *Stationary movement* - movement signal is constant during time of movement execution
2. *Dynamic movement* - movement signal is evolving during time of movement execution.

**Definition 12.** Movement automaton  $\mathcal{MA}$  for system defined by dynamics  $\dot{x} = f(x, u)$  is a structure defined as follow:

$$\begin{aligned} M &\equiv \text{set of movements} \\ u : M \times \mathbb{R} &\rightarrow \mathbb{R}^m \equiv \text{input function evolution} \\ \varphi : M \times M \times \mathbb{R} &\equiv \text{movement transition map} \\ B = M \times \mathbb{R}^2 &\equiv \text{movement buffer} \end{aligned} \tag{3.15}$$

Movement  $m_i \in M$  can be stationary or dynamic. Each movement in movement buffer has assigned duration  $(t_i, t_{i+1}] \in \mathbb{R}^2$ . Input function evolution  $u(B, t)$  defines input evolution in given execution time  $t = \tau + t_0, \tau \in (t_0, t_1]$ . Each movement  $m_i$  in movement automaton  $\mathcal{MA}$  is compliant with following rules:

1. Each movement  $m_i(t_i, t_{i+1})$  has non zero duration.
2. At switching time  $\tau$  between movements  $m_i$  and  $m_{i+1}$ , input function  $u_i(\tau + t_i) = u_{i+1}(0)$ .
3. Each dynamic movement should be linking two static movement and vice-versa.

### 3.4.1 Prediction stability of Movement Automaton

Because Movement automaton  $\mathcal{MA}$  is abstract level control realized trough *open loop hybrid automaton*, infinite receding horizon is questionable. This fact impacts the prediction reliability. This subsection formulates *prediction horizon of movement automaton*.

**Definition 13.** Movement automaton  $\mathcal{MA}$  prediction stability For system  $\dot{x} = f(x, \mathcal{MA})$  and predictor  $\dot{\hat{x}} = f(\hat{x}, \mathcal{MA})$  with separable nonlinear state  $h(\cdot)$  and input transformation  $g(\cdot)$  functions. With initial condition  $x(t_0) = \hat{x}(t_0) = x_0$  and finite execution time  $t \in [t_0, t_1]$ . Prediction deviation  $x(t) - \hat{x}(t)$  stays in unit ball defined by:

$$\mathcal{B}(x(t) - \hat{x}(t)) = \{y \in \mathbb{R}^n : y = x(t) - \hat{x}(t), \|y\| \leq \rho\} \tag{3.16}$$

*Proof.* Let  $\dot{x} = f(x, \mathcal{MA})$  be continuous time system, with state  $x \in \mathbb{R}^n$  and control with movement automaton  $\mathcal{MA}$ . Input function  $u(t)$  for movement buffer  $B = \{m_1(t_1), \dots, m_i(t_i)\}$  is interpreted as chained input function:

$$u(t) = \begin{cases} u(m_0, m_1, t_0, t_1) & : \text{for } m_1(t_1) \\ \vdots \\ u(m_{i-1}, m_i, t_{i-1}, t_i) & : \text{for } m_i(t_i) \end{cases} \quad (3.17)$$

Input function  $u(t)$  (3.17) is smooth and differentiable as given by movement automaton  $\mathcal{MA}$  definition (def. 12.). Let  $w(t)$  be state noise function with boundary  $\xi_w \in \mathbb{R} - \{\infty\}$ ,  $v(t)$  input noise function with boundary  $\xi_v \in \mathbb{R} - \{\infty\}$  and  $\hat{x}(t)$  predicted state for movement chain  $B = \{m_1(t_1), \dots, m_i(t_i)\}$ . We can define function  $V(t)$  as follow:

$$V(t) = \frac{1}{2}(x(t) - \hat{x}(t))^T I(x(t) - \hat{x}(t)) = (x(t) - \hat{x}(t))^2 \quad (3.18)$$

Given function can be derived by time  $t$ :

$$\dot{V}(t) = x(t) - \hat{x}(t) \quad (3.19)$$

Controlled system with state  $w(t)$  and input noise  $v(t)$ , where input and state are separable:

$$\dot{x} = f(x, \mathcal{MA}) = g(x(t) + w(t)) + h(u(t) + v(t)) \quad (3.20)$$

Predicted system equation:

$$\dot{\hat{x}} = f(\hat{x}, \mathcal{MA}) = g(\hat{x}(t)) + h(u(t)) \quad (3.21)$$

Therefore  $\dot{V}$  is equal to:

$$\begin{aligned} \dot{V}(t) &= (g(x(t) + w(t)) + h(u(t) + v(t))) - (g(\hat{x}(t)) + h(u(t))) \\ &= g(w(t)) + h(v(t)) \end{aligned} \quad (3.22)$$

Member  $g(x(t) - \hat{x}(t))$  is equal to zero, because of the premise that predicted  $\hat{x}$  and real state  $x$  are equal without state  $w(t)$  and input noise  $v(t)$  noise. Let us bound  $g(w(t))$ , boundary is straightforward in this case, because state noise  $w(t)$  is bounded by  $\xi_w$ . Boundary is given as:

$$\|g(w(t))\| \leq \|g(\sup(w(t)))\| \leq \|g(\xi_w)\| \leq \rho_w, \quad \rho_w \in \mathbb{R}^+ - \{\infty\} \quad (3.23)$$

Bounding member  $h(v(t))$  will be problematic due to mapping function (3.17). Introducing input noise to movement automaton mapping function will cause input noise to be characterized like this:

$$u(t) - \hat{u}(t) = \left( \begin{array}{c} u(m_0, m_1, t_0, t_1) + v_1(t) \\ \vdots \\ u(m_{i-1}, m_i, t_{i-1}, t_i) + v_i(t) \end{array} \right) - \left( \begin{array}{c} \hat{u}(m_0, m_1, t_0, t_1) \\ \vdots \\ \hat{u}(m_{i-1}, m_i, t_{i-1}, t_i) \end{array} \right) \quad (3.24)$$

This can be simplified to:

$$\sum_{l=0}^i v_l(t)^l, \quad i \in \mathbb{R}^+ - \{0, \infty\} \quad (3.25)$$

So for case of input noise  $v(t)$  member  $h(v(t))$  of  $\dot{V}(t)$  can be bounded like follow:

$$\|h(v(t))\| \leq \left\| h_{sup} \sum_{l=0}^i v_l(t)^l \right\| \leq \|h(\xi^i)\| \leq \rho_v \quad \rho_v \in \mathbb{R}^+ - \{\infty\} \quad (3.26)$$

Where  $i$  is finite count of movements in movement automaton  $\mathcal{MA}$  buffer  $B$ ,  $i = |B|$ . This parameter is cumulative which is given by member  $\left\| h_{sup} \sum_{l=0}^i v_l(t)^l \right\|$ , but it is bounded by  $\|h(\xi^i)\|$ . One can define boundary as follow:

$$\|\dot{V}(t)\| \leq \rho_w + \rho_v = \rho, \quad \rho \in \mathbb{R}^+ - \{\infty\} \quad (3.27)$$

With boundary defined for derivation of  $V(t)$  one can define boundary for  $V(t)$  like follow for time of prediction  $t_0$  and time of movement buffer execution end  $t_i$ :

$$\|V(t)\| \leq \int_{t_0}^{t_i} \|\dot{V}(t)\| dt \leq \int_{t_0}^{t_i} \rho dt = \rho(t_i - t_0) \quad (3.28)$$

□

### 3.4.2 Lyapunov stability of Movement Automaton

It has been proven that system with movement automaton is stable in terms of finite receding horizon prediction. There is still open issue of *Lyapunov stability*. To prove Lyapunov stability one needs to construct tracking problem of waypoint. Let say that system initial state  $x(t_0) \in \mathcal{B}_s \subset \mathbb{R}^n$  is unit ball where system  $\dot{x} = g(x) + h(u)$  is trying to reach equilibrium point  $\hat{x}(t_i)$ . Equilibrium point is predicted state  $\hat{x}$  at final time of movement automaton  $\mathcal{MA}$  buffer execution  $t_i$ . Then region of attraction  $\mathcal{B}(\hat{x}(t_i), \rho)$ , where  $\hat{x}(t_i)$  is center of unit ball and  $\rho$  is prediction deviation boundary at time  $t_i$  given by definition (def. 13). Let us define system  $y$  for tracking problem with equilibrium point  $x(t_i)$  and initial state  $x(t_0) \in \mathcal{B}_s$  for finite time period  $t \in [t_0, t_i]$

$$y = x(t) - \hat{x}(t_i), \quad x(t_0) \in \mathcal{B}_s \subset \mathbb{R}^n, \quad (3.29)$$

It is obvious that region of attraction  $\mathcal{B}(\hat{x}(t_i), \rho)$ , is subset of  $\mathcal{B}_s$ . Let us define simple Lyapunov function  $V(y)$  like follow:

$$V(y) = \frac{1}{2} y^T y = \frac{1}{2} \sum_{l=1}^n (x_l(t)^2 - \hat{x}_l(t_i)^2) \quad (3.30)$$

With derivation in time given by equation:

$$\dot{V}(y) = \sum_{l=1}^n \frac{\partial V(y)}{\partial y_l} \quad (3.31)$$

Now lets check second Lyapunov method stability conditions for  $y$ :

1.  $V(0) = 0$  - this condition is valid, because Lyapunov function is 0 at equilibrium point  $\hat{x}(t_i)$ .

2.  $V(y) < 0, \forall y \in \mathcal{B}_s - \{0\}$  - this condition is valid, because  $\frac{1}{2} \|y\| > 0, \forall y \in \mathcal{B}_s - \{0\}$ .
3.  $\dot{V}(y) < 0$  or  $\dot{V}(y) \leq 0$  - convergence criterion is fully satisfied in case of no existing obstacle, because avoidance framework converges to goal waypoint exponentially fast.

Convergence condition  $\dot{V}(y) < 0$  is not guaranteed in term of obstacle avoidance, because in some cases  $\dot{V}(y) > 0$  while avoiding enormous obstacle. Movement automaton  $\mathcal{MA}$  is exponentially or at least asymptotic stable in terms of Lyapunov stability, within region of attraction  $\mathcal{B}(\hat{x}(t_i), \rho)$ . Obstacle avoidance framework can be or can not be Lyapunov stable depending on obstacle set  $\mathcal{O}$ . Reachability of waypoint  $\hat{x}(t_i)$  is determined by system reachability set  $\mathcal{R}$ .

### 3.5 LiDAR

LiDAR (Light Detection And Ranging) is active form of remote sensing: information is obtained from a signal which is sent from a transmitter and reflected by a target, and detected by a receiver back at the source. Following types of information can be obtained:

1. *Range to target* - topographic LiDAR or laser altimeter.
2. *Chemical properties of target* - differential absorption LiDAR.
3. *Velocity of target* - Doppler LiDAR.

Chemical properties of target are out of scope. Velocity of target seems as interesting property to investigate, but this type of LiDAR is usually used for meteorological measurements of wind currents [? ]. Extended research in LiDAR as obstacle detection sensor has been executed by research group around Sabatini [? ] and Ramasy [? ].

LiDAR output is represented as point cloud it is described by following definition.

**Definition 14** (Scanned point and Point-cloud). *Consider viewpoint  $v$  as origin of  $\mathbb{R}^3$  space in Let point  $p \in P$  be defined in polar coordinates:*

$$p = [d, \theta, \varphi, t]^T \quad (3.32)$$

*Where  $d$  is distance to scanner,  $\theta$  is horizontal angle from origin,  $\varphi$  is vertical angle to origin,  $t$  is time of retrieval.*

*Point-cloud is set of points scanned in small enough time-frame, based on processing raw point data it can have following representations:*

1. *Local point-cloud* - position of sensor is used as origin of space and points can be represented in orthogonal or planar representation.
2. *Global point-cloud* - global position of sensor is used as reference to calculate global position of points.

Point-cloud is usually addressed as *raw point-cloud* in case if its represented in Local planar coordinates. Other forms of point cloud require further processing and they are not feasible for real time obstacle detection and avoidance [? ].

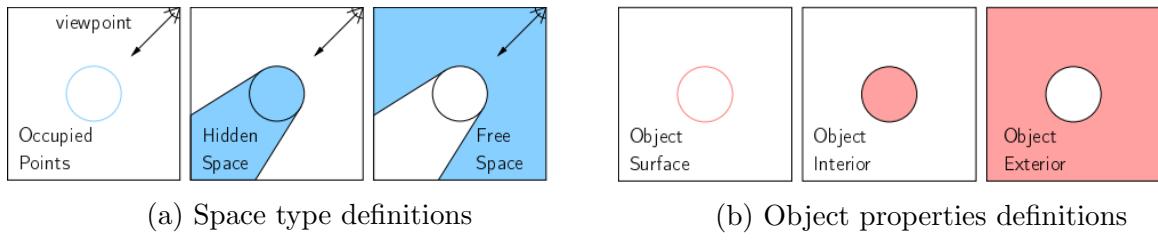


Figure 3.1: Six spaces of interest [? ]

Because of real-time obstacle avoidance it is necessary to introduce following terminology:

1. *Occupied points* - points which have been detected by LiDAR (also addressed as visible points).
  2. *Hidden space* - space which is hidden behind occupied points, from viewpoint it is uncertain what is in that space.
  3. *Free space* - space which is visible from viewpoint and it is not occupied by known objects.
  4. *Object surface* - detected and undetected object surface
  5. *Object interior* - occupied space by object.
  6. *Object exterior* - free space around known objects.

Existing method for space segregation [? ] can yield to following definition.

**Definition 15** (Accessible space). Consider known space  $S$  as space explored by sensor (it can have different viewpoint along previous 3D trajectory). Intersection between object exterior  $S_E$  and free space  $S_F$  gives us Accessible space.

$$S_A = S_E \cap S_F \quad (3.33)$$

Accessible space  $S_A$  (15) is our bordering limitation for reachable space of system  $R(\tau, t_0, \vec{x}_0)$  (def. 1.);

### 3.6 Complements of algebra

3D Cartesian space gives us an x, y, and z axis (describing position based on horizontal placement, vertical placement, and depth respectively). The coordinates for any point within this space are shown as a tuple  $(x, y, z)$ . Coordinate system used this work will be viewed as right-handed system (thumbs points at positive direction of x-axis, index finger is pointing to positive direction of y-axis, then positive direction of z axis is shown by remaining fingers).

Euler have worked on universal rotation theorem which was presented in [? ]. Rigid body dynamics and rotation matrices have been proposed in book by Schaub [? ]. Local coordinates are used, where plane center of gravity is used as center point and plane heading is oriented to X axis. This local coordinate system is called Euler Normalized Unit-frame (ENU). Rotation matrices are used to transform between two displaced coordinate systems. Roll angle  $\alpha$  rotation is defined around X-axis by matrix (3.34) on

YZ-plane. Pitch angle  $\beta$  rotation matrix is defined around Y-axis by matrix (3.35) on XZ plane. Yaw angle  $\gamma$  rotation matrix is defined around Z axis by matrix (3.36) on XY plane.

$$R_{YZ} = R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (3.34)$$

$$R_{XZ} = R_\beta = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (3.35)$$

$$R_{XY} = R_\gamma = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.36)$$

Final rotation matrix in XYZ ordered space is given by equation(3.37).

$$\begin{aligned} R_{XYZ} = R_{\alpha\beta\gamma} &= R_{XY} * R_{XZ} * R_{YZ} = R_\gamma * R_\beta * R_\alpha \\ &= \begin{bmatrix} \cos \beta \cos \gamma & \cos \gamma \sin \alpha \sin \beta - \cos \alpha \sin \gamma & \sin \alpha \sin \gamma + \cos \alpha \cos \gamma \sin \beta \\ \cos \beta \sin \gamma & \cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & \cos \alpha \sin \beta \sin \gamma - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \alpha \cos \beta \end{bmatrix} \end{aligned} \quad (3.37)$$

To keep solution numerically stable and rotations numerically stable gimbal lock prevention is necessary [? ]. Gimbal lock occurs when one of matrices (3.34, 3.35, 3.36) is singular or final matrix for XYZ rotation is singular (3.37). Gimbal lock leads to loose of one or more degree of freedom, depending on rank and space dimension of singular matrix. To prevent gimbal lock it is necessary to introduce mechanism to check if rotation matrix is regular. For this purpose normative reset function is introduced:

$$[\alpha, \beta, \gamma]^T = f(t, \alpha^-, \beta^-, \gamma^-), \quad \text{norm}(R_{\alpha\beta\gamma}) = 3 \quad (3.38)$$

Function resets yaw  $\gamma$  or roll  $\alpha$  angle to initial position to keep degree of rotation matrix. Simpler but not fault tolerant solution is to keep angles  $\alpha, \beta, \gamma, \in (-\pi, \pi]$  range.

### 3.6.1 Transformation between coordinate systems

Polar coordinate system represents point in form of vector [*distance*, *angle<sub>horizontal</sub>*, *angle<sub>vertical</sub>*], which is ideal for representation of LiDAR scanned point, because usually total point distance and pair of angles are returned. Using most common LiDAR with horizontal rotation  $\theta$  and vertical mirror inclination  $\varphi$ , one can define planar coordinate  $p_{planar} = [d_{xyz}, \theta, \varphi]$  which is dual to Cartesian coordinate  $p_{cartesian} = [x, y, z]$ . If rotation angles are forced to stay as  $\theta, \varphi \in (-\pi, \pi]$  transformation function is bijection.

Transformation from planar to Cartesian representation is defined by following series of functions (3.39, 3.40, 3.41, 3.42).

$$d_{xy} = \cos \varphi \cdot d_{xyz} \quad (3.39)$$

$$z = \sin \varphi \cdot d_{xyz} \quad (3.40)$$

$$y = \sin \theta \cdot d_{xy} \quad (3.41)$$

$$x = \cos \theta \cdot d_{xy} \quad (3.42)$$

Transformation from Cartesian to planar representation is defined by following series of functions (3.43, 3.44, 3.45, 3.46).

$$d_{xyz} = \sqrt{x^2 + y^2 + z^2} \quad (3.43)$$

$$d_{xy} = \sqrt{x^2 + y^2} \quad (3.44)$$

$$\theta = \arctan \frac{y}{x} \quad (3.45)$$

$$\varphi = \arctan \frac{z}{d_{xy}} \quad (3.46)$$

**Definition 16.** Global coordinate system  $\mathcal{X}_S$  takes as center  $c_{S0}$  well known point (for example center of geo-reference model in GNSS systems) every reference distance, plane or angle is calculated taking this center to mind.

**Definition 17.** Local coordinate system  $\mathcal{X}_L$  takes as center  $c_{L0}$  frame of vehicle and can be changing position and orientation in global coordinate frame  $\mathcal{X}_S$ .

**Definition 18.** Global position of planar obstacle  $o_i \in \Omega_{3D}$ . Let  $o_i = [d_o, \theta_o, \varphi_o]^T$  be planar position of obstacle  $o_i$  in local coordinate frame of vehicle with global Cartesian position  $[x_v, y_v, z_v]^T$  and normalized orientation angles  $[\alpha_v, \beta_v, \gamma_v]$ .

Then Cartesian position of obstacle  $o_i$ ,  $[x_o, y_o, z_o]^T$  in local coordinate frame is given by transformation functions  $x_o$  (3.42),  $y_o$  (3.41),  $z_o$  (3.40).

Global position of planar obstacle  $o_i$ ,  $[x_g, y_g, z_g]^T$  is given by following equation:

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \left[ R_{XYZ}(\alpha_v, \beta_v, \gamma_v) \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} \right] \quad (3.47)$$

**Definition 19.** Local position of global coordinate  $[x_g, y_g, z_g]^T \in \mathbb{R}^3$ . Let there be vehicle with global Cartesian position  $[x_v, y_v, z_v]^T$  and normalized orientation angles  $[\alpha_v, \beta_v, \gamma_v]$ . in global coordinate frame  $\mathcal{X}_X$ .

Then local Cartesian coordinate position  $[x_l, y_l, z_l]^T$  of point  $[x_g, y_g, z_g]^T$  is given by following equation:

$$\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \left[ R_{XYZ}(-\alpha_v, -\beta_v, -\gamma_v) \left( \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} - \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} \right) \right] \quad (3.48)$$

Local planar position is given as  $[d_l, \theta_l, \varphi_l]$ , where  $d_l$  is given by (3.43),  $\theta_l$  is given by (3.45).  $\varphi_l$  is given by (3.46), where  $[x_l, y_l, z_l]$  are used as local coordinates.

### 3.6.2 Planar surface calculation

The problem is to calculate intersected surface  $dA$  of ball subsurface defined by radius  $r$ , horizontal span  $\phi$  and vertical span  $\theta$ . From classical mechanics one can formulate problem as given by figure 3.2a. The intersection plot is in figure 3.2b.

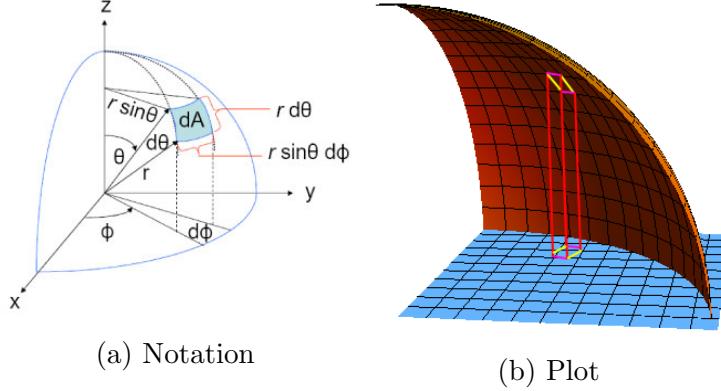


Figure 3.2: Planar surface calculation notation and plot

One can use first fundamental form to determine the surface area element. Recall that this is the metric tensor, whose components are obtained by taking the inner product of two tangent vectors in planar space  $g_{i,j} = X_i \cdot X_j$ , for tangent vectors  $X_i, X_j$ . Following identification for the components of metric tensor will be used:

$$g_{ij} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \quad (3.49)$$

Where  $E = \langle X_u, X_u \rangle$ ,  $F = \langle X_u, X_v \rangle$ , and  $G = \langle X_v, X_v \rangle$ . Lagrange's identity can be used , which tells us that the squared area of a parallelogram in space is equal to the sum of the squares of its projections onto the Cartesian plane:

$$|X_u \times X_v|^2 = |X_u|^2 |X_v|^2 - (X_u \cdot X_v)^2 \quad (3.50)$$

Given example is displayed in figure 3.2b. The area element is given as:

$$\begin{aligned} dA &= |X_u \times X_v| \, du \, dv \\ &= \sqrt{| |X_u|^2 |X_v|^2 - (X_u \cdot X_v)^2} \, du \, dv \\ &= \sqrt{EG - F^2} \, du \, dv \end{aligned} \quad (3.51)$$

We will find tangent vectors via the usual parametrization which give,  $X(\phi, \theta) = [r \cos \phi \sin \theta, r \sin \phi \sin \theta, r \cos \theta]$ , so that tangent vectors are simply defined as:

$$\begin{aligned} X_\phi &= [-r \sin \phi \cos \theta, r \cos \phi \sin \theta, 0] \\ X_\theta &= [-r \cos \phi \sin \theta, r \sin \phi \cos \theta, -r \sin \theta] \end{aligned} \quad (3.52)$$

Computing the elements of the first fundamental form gives us:

$$E = r^2 \cos^2 \theta, \quad F = 0, \quad G = r^2 \quad (3.53)$$

Thus final difference is given as:

$$dA = \sqrt{r^4 \cos^2 \theta} \, d\theta \, d\phi = r^2 \cos \theta \, d\theta \, d\phi \quad (3.54)$$

# Chapter 4

## Probabilistic approach

Given a partial obstacle map, a mission plan, and a small UAV equiped with ADS-B and LiDAR sensor a safe approach to real-time obstacle avoidance with a small computational footprint compatible with the low computational power available in small UAVs. The main application consists of terrain and intruder avoidance for low altitude flights. Following assumption holds:

1. There is a obstacle database charting some known obstacles prior the flight.
2. There is possibility of non-adversarial and non-cooperative intruders.
3. Estimates of the state of the UAV are available.
4. The operational space does not contain 'traps' such as caves. 'Traps' arise because the field of view of the LiDAR does not include the whole space surrounding the UAV.
5. There is a mission plan for the UAV consisting of waypoints which are reachable with vehicle dynamics.

### 4.1 Simple plane model

For avoidance theorem formulation in three dimensional space simplified rigid body kinematic model will be used. This model have decoupled roll, yaw and pitch angles which enables to provide simpler and more clean control (e.g movements can be simplified).

$$\vec{x} = [x_v, y_v, z_v, \alpha_v, \beta_v, \gamma_v]^T \quad (4.1)$$

State vector (4.1) defined as positional state in euclidean position in right-hand euclidean space, where  $x_v, y_v, z_v$  states, for latitude, longitude and altitude. Orientation angles for vehicle are  $\alpha, \beta, \gamma$  for roll, pitch, yaw angle.

$$\vec{u} = [v, \omega_{\alpha_v}, \omega_{\beta_v}, \omega_{\gamma_v}]^T \quad (4.2)$$

Input vector (4.2) is defined as frontal velocity of vehicle  $v$ , orientation change in main axes as angular speed  $\omega_{\alpha_v}, \omega_{\beta_v}, \omega_{\gamma_v}$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = R_{XYZ}(\alpha_v, \beta_v, \gamma_v) \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f_{v_x}(v, \alpha_v, \beta_v, \gamma_v) \\ f_{v_y}(v, \alpha_v, \beta_v, \gamma_v) \\ f_{v_z}(v, \alpha_v, \beta_v, \gamma_v) \end{bmatrix} = \begin{bmatrix} v \cos(\beta_v) \cos(\gamma_v) \\ v \cos(\beta_v) \sin(\gamma_v) \\ -v \sin(\beta_v) \end{bmatrix} \quad (4.3)$$

Velocity distribution function (4.3) is defined through standard rotation matrix (3.37) and frontal velocity  $v$ , final distributed velocity is time depending function with values  $v_x, v_y, v_z$  given by functions  $f_{v_x}(\dots), f_{v_y}(\dots), f_{v_z}(\dots)$ . Final nonlinear model which have been derived from reference model [?] is defined by (4.4).

$$\begin{aligned}\dot{x}_v &= v_x = f_{v_x}(v, \alpha_v, \beta_v, \gamma_v) = v \cos(\beta_v) \cos(\gamma_v) \\ \dot{y}_v &= v_y = f_{v_y}(v, \alpha_v, \beta_v, \gamma_v) = v \cos(\beta_v) \sin(\gamma_v) \\ \dot{z}_v &= v_z = f_{v_z}(v, \alpha_v, \beta_v, \gamma_v) - v \sin(\beta_v) \\ \dot{\alpha}_v &= \omega_{\alpha_v} \\ \dot{\beta}_v &= \omega_{\beta_v} \\ \dot{\gamma}_v &= \omega_{\gamma_v}\end{aligned}\tag{4.4}$$

## 4.2 Model predictive control

To employ MPC scheme with movement automaton some adjustments needs to be made to original discrete control scheme. These changes are on formal level, because the step time  $t_s$  of discrete predictor is similar to fixed movement time  $t_i$ . So if constraint is satisfied:

$$t_s = t_{i+1} - t_i, \forall i \in B \in \mathcal{MA} \tag{4.5}$$

MPC problem can be solved as discrete time nonlinear MPC problem with fixed discrete time step  $t_s = t_{i+1} - t_i$ . When movement chain  $m_1(t_1), \dots, m_n(t_n)$ . In our case movement automaton  $\mathcal{MA}$  is used as input control and state space is constrained by obstacle space  $\mathcal{O}$ . One can define optimal control problem as follow:

$$\text{Minimize } \sum_{k=0}^{N-1} f_0(k, x(k), u(k)) + \Phi(x(T))$$

Subject to:

$$\text{Dynamics: } x(k+1) = f(k, x(k), u(k)), \quad k = 0, 1, \dots, T-1 \tag{4.6}$$

$$\text{Initial conditions: } x(0) = x_0$$

$$\text{Control constraints: } u(k) \in \mathcal{MA}_i, \quad k = 0, 1, \dots, T-1$$

$$\text{State space constraints: } x(k) \in \{\mathbb{X} - \mathcal{O}\}, \quad k = 0, 1, \dots, T.$$

Where  $\sum_{k=0}^{N-1} f_0(k, x(k), u(k))$  is dynamic control cost functional,  $\Phi(x(T))$  is terminal control cost functional. Dynamics of system is given by  $f(k, x(k), u(k))$ . Initial conditions  $x_0$  are considered for initial prediction time 0 up to finite prediction horizon  $H_T$  at time  $T$ . Control input at time  $k$  is considered as time of movement  $m(t_i)$  execution time  $t_i$ . State  $x(k)$  is constrained as  $x(k) \notin \mathcal{O}$ . for any time  $k = 0, \dots, T$ . This problem is modified problem of dynamic programming with constraints.

### 4.2.1 Model predictive control with movement automaton

This section describes method used for predictor. The book [?] addresses state space modeling of time series. Time series are long term statistical study method. Methods used in time series can be abused to obtain predictor for movement automaton  $\mathcal{MA}$ . Predictor uses observed state  $x$  and known space assessment of avoidance grid  $\mathcal{A}(t_i)$  to

predict future state chain  $\hat{x}$  and movement chain  $\hat{B}$ . Therefore predictor function is given as follows:

$$f : \vec{x} \times \mathcal{A}(t_i) \rightarrow \hat{x} \times \hat{B} \quad (4.7)$$

The main advantage of movement automaton  $\mathcal{MA}$  control is separability of input  $u$  from state  $\hat{x}$ . Input signal  $u(t)$  is generated as interpretation of movement automaton chain  $u(t) = \mathcal{I}(m_1(t_1), \dots, m_i(t_i))$ . The system is given by discrete time equation:

$$x^+ = f(x, \mathcal{I}(m_1(t_1), \dots, m_i(t_i))) \quad (4.8)$$

$\mathcal{I}$  is movement interpretation function. Predictor must therefore satisfy following equation:

$$\hat{m}_i(t_i) = \mathcal{I}^{-1}(\hat{x}(t_i), \mathcal{I}(m_1(t_1), \dots, m_{i-1}(t_{i-1})), \hat{x}(0) = x(t_p)) \quad (4.9)$$

Where  $\mathcal{I}^{-1}$  is inverse movement interpretation functor,  $\hat{x}(t_i)$  is expected predicted state at time  $t_i$  and  $m_1(t_1), \dots, m_{i-1}(t_{i-1})$  is previously executed movement chain, which can be partially predicted movement chain. Therefore prediction window or receding horizon is variable. The time series are using application of discrete events in state space models, which have been partially linearized. Therefore standard linear model for discrete time  $x^+ = f(x, u)$  where  $u$  is some discrete event set can be abused as movement predictor function  $\mathcal{I}^{-1}$ . Implementation example can be found in section 4.3.

### 4.2.2 Predictor corrections

For this part assume that system operates in infinite time frame  $t \in [t_0, \infty)$ . The predictor is giving predicted output  $\hat{x}(t)$  and predicted input  $\hat{u}(t)$  and is given by following equation for system (4.4):

$$\hat{x}(t_d) f_p(\hat{x}(t_d), \hat{u}(t_d), w(t_d), v(t_d)) \quad (4.10)$$

where  $t_d$  is prediction time. Predicted input function  $\hat{u}(t_d)$  is disturbed by input disturbance  $v(t_d)$  and state disturbance  $w(t_d)$ . The difference between predicted state and real state increases with time. Main purpose of control is tracking, therefore the tracking deviation should be employed as marginal error function  $e_d$ .

$$e_d = \sqrt{\left\| \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} - \begin{bmatrix} \hat{x}_v \\ \hat{y}_v \\ \hat{z}_v \end{bmatrix} \right\|^2} + \sqrt{\left\| \begin{bmatrix} \beta_v \\ \gamma_v \end{bmatrix} - \begin{bmatrix} \hat{\beta}_v \\ \hat{\gamma}_v \end{bmatrix} \right\|^2} \quad (4.11)$$

This error is accumulating prediction  $\hat{x}(t_c), \hat{u}(t_c)$  deviation from real input  $u(t_c)$  and state vector at time of comparison  $t_c$ . The marginal error function  $e_m$  is in respective notation of system (4.4). To compare deviation some well established parameter needs to be used. The ideal candidate is safety margin  $s_m$ , because some partition of safety margin is prediction error, therefore the hard condition for correction is  $\frac{1}{12}s_m$ . Time of comparison is possible at any execution time  $t$ , but the correction is possible only at movement automaton switching time when movements  $m_{i-1}(t_{i-1}), m_i(t_i)$  are switching. Let us define time of prediction  $t_{p1}$  and time of correction  $t_{p2}$  with following constraints  $t_{p1} \leq t_c \leq t_{p2}$ . then movement chain at time of original prediction  $B(t_{p1}) = \{m_1^{p1}(t_{1,p1}), \dots, m_\infty^{p1}(t_{t_{p1}+\infty})\}$  and new chain for time of new prediction  $t_{p2}$  to be chained after time of correction  $t_c$  given as  $B(t_{p2}) = \{m_1^{p2}(t_{1,p2}), \dots, m_\infty^{p2}(t_{t_{p2}+\infty})\}$ . Then the final controller input  $B$  is given as following equation:

$$B_{\mathcal{MA}} = \bigcup_{p=\{p_1(t_1), \dots, p_N(t_n)\}}^{c=\{t_{c_1}, \dots, t_{c_{N-1}}\}} \{m_{1,p}(t_{1,p}), \dots, m_{c,p}(t_{c,p}), \dots, m_{k_p,p}(t_{k_p,p})\} \quad (4.12)$$

Where  $p = \{p_1(t_1), \dots, p_N(t_n)\}$  is chain of predictions  $c = \{t_{c_1}, \dots, t_{c_{N-1}}\}$  is chain of corrections. The prediction  $p_i$  is followed by correction  $c_i$  if necessary. Movement chain  $m_{1,p}(t_{1,p}), \dots, m_{c,p}(t_{c,p}), \dots, m_{k_p,p}(t_{k_p,p})$  denotes that some movements from previous prediction  $p_i$  are executed after correction  $c_i$  and then new prediction  $p_{i+1}$  is employed. The duration of prediction sequences may differ and its denoted by  $k_p$  which marks last executed movement executed from prediction  $p$ . Movement  $m_{c,p}(t_{c,p})$  denotes movement executed from chain  $p_i$  at time of correction  $c_i$ . The goal of predictive control quality is to  $\lim_{t \rightarrow \infty} k_p = \infty$ . This has been achieved in simulation. Unfortunately no disturbances were present to employ this approach. Moving receding horizon is known from literature [?], it has been modified for movement automaton  $\mathcal{MA}$  as shown in this section.

### 4.3 Movement automaton predictor

Vehicle system is given by model from section 4.1. Vehicle control is defined as movement automation  $\mathcal{MA}$ . Movement predictor needs to be developed in order to obtain control signal  $u(t)$ . It is possible to predict movement buffer  $B_{\mathcal{MA}}$ ,

Vehicle state  $x(t_0)$  at start of mission execution is known. Vehicle state in discrete time  $x(t)$  is given by equation 4.13. Where  $x_v(t), y_v(t), z_v(t)$  is vehicle position in local coordinate frame and  $\alpha_v(t), \beta_v(t), \gamma_v(t)$  is vehicle orientation angles.

$$x(t) = [x_v(t), y_v(t), z_v(t), \alpha_v(t), \beta_v(t), \gamma_v(t)]^T; \quad (4.13)$$

Base movement table for movements  $m_i(1) \in M$  have been measured on system model (4.1). This table represents vehicle position and orientation differences after execution of movement. Vehicle initial state was set at center of local coordinate frame with narrow orientation  $[x_0, y_0, z_0] = [0, 0, 0]$ . Vehicle initial orientation was aligned with main frame axis X, therefore initial orientation angles are  $[\alpha_0, \beta_0, \gamma_0] = [0, 0, 0]$ . Vehicle velocity was set to constant value  $v_v = 1ms^{-1}$ . Vehicle position after movement execution is given by parameters  $x_b, y_b, z_b$  at time  $t + 1$ . Vehicle orientation is given by parameters  $\alpha_b, \beta_b, \gamma_b$ . Given parameters are also absolute shifting, because initial state is  $[x_0, y_0, z_0, \alpha_0, \beta_0, \gamma_0]$  set to  $\vec{0}$ .

$v_x/m_i$	Straight	Down	Up	Left	Right
$x_b[m]$	1.00	0.98	0.98	0.98	0.98
$y_b[m]$	0	0	0	0.13	-0.13
$z_b[m]$	0	-0.13	0.13	0	0
$\alpha_b[rad]$	0	0	0	0	0
$\beta_b[rad]$	0	0.2	-0.26	0	0
$\gamma_b[rad]$	0	0	0	0.26	-0.26

Table 4.1: Base values for movement application, vehicle position difference  $x_v, y_v, z_v$  and orientation differences  $\alpha_v, \beta_v, \gamma_v$ .

$v_x/m_i$	Down-Left	Down-Right	Up-Left	Up-Right
$x_b[m]$	0.76	0.76	0.76	0.76
$y_b[m]$	-0.13	0.13	0.13	-0.13
$z_b[m]$	-0.13	-0.13	0.13	0.13
$\alpha_b[rad]$	0	0	0	0
$\beta_b[rad]$	-0.26	-0.26	0.26	0.26
$\gamma_b[rad]$	0.26	-0.26	0.26	-0.26

Table 4.2: Base values for movement application, vehicle position difference  $x_v, y_v, z_v$  and orientation differences  $\alpha_v, \beta_v, \gamma_v$ .

With defined base movement tables (tab. 4.1., 4.2.) with defined shifting for movement at time  $t_0+1$ , *rotation* (4.14) and *shifting* (4.15) equations must be defined in order to obtain predicted position  $\hat{x}(t+1)$ . Position after movement execution  $[\hat{x}(t+1), \hat{y}(t+1), \hat{z}(t+1)]$  is depending on vehicle orientation before movement execution  $[\alpha_v(t), \beta_v(t), \gamma_v(t)]$ . Rotation function rotates vehicle position  $[x_b(t_0), y_b(t_0), z_b(t_0)]$  according to vehicle orientation angles  $[\alpha_v(t), \beta_v(t), \gamma_v(t)]$ . For rotation standard rotation matrix  $R_{XYZ}(\alpha, \beta, \gamma)$  (3.37) is used. Final position offset vector  $[\tilde{x}_b(t+1), \tilde{y}_b(t+1), \tilde{z}_b(t+1)]$  at predicted time  $t+1$  is given by equation 4.14.

$$\begin{bmatrix} \tilde{x}_b \\ \tilde{y}_b \\ \tilde{z}_b \end{bmatrix} = R_{XYZ}(\alpha_v(t), \beta_v(t), \gamma_v(t)) \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad (4.14)$$

Predicted state vector  $\hat{x}(t+1)$  at time  $t+1$  is obtained by combining position offset vector  $[\tilde{x}_b(t+1), \tilde{y}_b(t+1), \tilde{z}_b(t+1)]$ , orientation offset vector  $\alpha_b(t+1), \beta_b(t+1), \gamma_b(t+1)$ , as given in equation 4.15.

$$\begin{aligned} \hat{x}(t+1) &= [\hat{x}_v(t+1), \hat{y}_v(t+1), \hat{z}_v(t+1), \hat{\alpha}_v(t+1), \hat{\beta}_v(t+1), \hat{\gamma}_v(t+1)]^T \\ \hat{x}_v(t+1) &= x_v(t) + \tilde{x}_b \\ \hat{y}_v(t+1) &= y_v(t) + \tilde{y}_b \\ \hat{z}_v(t+1) &= z_v(t) + \tilde{z}_b \\ \hat{\alpha}_v(t+1) &= \alpha_v(t) + \alpha_b \\ \hat{\beta}_v(t+1) &= \beta_v(t) + \beta_b \\ \hat{\gamma}_v(t+1) &= \gamma_v(t) + \gamma_b \end{aligned} \quad (4.15)$$

Movement automaton  $\mathcal{MA}$  has chaining property, therefore movement prediction chaining is also possible via recursive procedure for discrete execution time  $t_i$ . There exist predicted movement buffer  $\hat{B}$  with ordered movement sequence  $m_1(1), \dots, m_i(1)$ , vehicle state after applying movement  $m_{i+1}$  at time  $t_i$  can be obtained via equation 4.16.

$$\hat{x}(t_i+1) = f(\hat{x}(t_i), m_{i+1}(1)) \quad (4.16)$$

Prediction function  $f(\hat{x}(t_i), m_{i+1}(1))$  is in this case function defined by equation 4.15, where parameters  $[x_b, y_b, z_b, \alpha_b, \beta_b, \gamma_b]$  are selected based on movement  $m_{i+1}(1)$  type from look-up tables 4.1., 4.2.

## 4.4 General algorithm

General algorithm describes how final probabilities for cell and trajectory are calculated. It shows how time continuum impacts various partial ratings. This section is situated at one specific time of avoidance  $t_i$ .

*Obstacle probability*  $P_O \in [0, 1]$  for cell  $c_{i,j,k}$  have been identified during analysis. Later on partial probabilities of *map obstacle*  $P_{O_M}$ , *detected obstacle*  $P_{O_D}$ , and *aggregated intruders probability*  $P_{O_I}$  will be introduced. The final obstacle probability  $P_O$  for one cell  $c_{i,j,k}$  is given as:

$$P_O = 1 - \{(1 - P_{O_I}) \times (1 - P_{O_M}) \times (1 - P_{O_D})\} \quad (4.17)$$

The obstacle probability  $P_O$  for cell  $c_{i,j,k}$  is given as inverted product of *intruder obstacle probability*  $P_{O_I}$  (sec. 5.1), *detected obstacle probability*  $P_{O_D}$  (sec. 5.2), and *Map obstacle probability*  $P_{O_M}$  (sec. 5.4).

*Visibility probability*  $P_V$  for cell  $c_{i,j,k}$  is defined in section 5.3. The calculation principle is to count LiDAR beams passed through cell  $c_{i,j,k}$  to count of possibly passing LiDAR beams. The meaning of this rating is to show how good was visibility of space portion contained by cell  $c_{i,j,k}$ .

*General algorithm* inputs and initialization is given by eq. 4.19. The set of intruders  $\mathcal{I}$  is set of detected intruders in time frame  $[t_{i-1}, t_i]$ , where  $t_{i-1}$  is previous passed avoidance (decision) time and  $t_i$  upcoming decision time. The initial position of intruders is projected to upcoming decision time  $t_i$ .

*Map obstacle set*  $\mathcal{O}_m$  is loaded from obstacle database via *radius select* to expected vehicle position  $\hat{x}(t_i) \rightarrow \mathcal{F}_{3D}$ .

*Detected obstacle set*  $\mathcal{O}_d$  is gathered from vehicle sensor systems in time interval  $[t_{i-1}, t_i]$ . The obstacles are steam-lined to avoidance grid  $\mathcal{A}(t_i)$  at expected vehicle position  $\hat{x}(t_i) \rightarrow \mathcal{F}_{3D}$ .

*Reach set*  $\mathcal{R}$  (set of trajectories) is integral part of avoidance grid  $\mathcal{A}(t_i)$  and contains trajectories bounded by  $\mathcal{A}(t_i)$  from expected vehicle position  $\hat{x}(t_i) \rightarrow \mathcal{F}_{3D}$ . Each trajectory has ordered set of passing cells  $\mathcal{C}(\mathcal{T}(\vec{x}, B))$ .

*Avoidance grid*  $\mathcal{A}(t_i)$  is partitioned into cells  $c_{i,j,k}$ ,  $i \in 1..a$ ,  $j \in 1..b$ ,  $k \in 1..c$ ,  $a, b, c \in \mathbb{N}^+$ . There exist homogeneous partition of field of vision and cell space is given by:

$$c_{i,j,k} \rightarrow \mathbb{R}^3 = \left\{ \vec{p} = [d, \theta, \varphi] \in \mathbb{R}^3 : \begin{array}{l} d_i \leq d \leq d_{i+1} \\ \theta_i \leq \theta \leq \theta_{i+1} \\ \varphi_i \leq \varphi \leq \varphi_{i+1} \end{array} \right\} \quad (4.18)$$

The cell space is non-inclusive space in local coordinates with origin at vehicle state  $\hat{x}(t_i) \rightarrow \mathbb{R}^3$  and with main axis (X) at vehicle heading in right hand coordinate frame. The cell space is same as in deterministic approach. Each cell has *passing trajectory set*  $\mathcal{T}(c_{i,j,k})$  containing passing trajectories.

*Initialization* phase is executed every avoidance time  $t_i$ , the initialization is executed in following manner:

1. *Grid cell*  $\forall c_{i,j,k} \in \mathcal{A}(t_i)$  - visibility probability  $P_V$  is set to 1, because its assumed that all of avoidance grid  $\mathcal{A}(t_i)$  is visible, obstacle probability  $P_O$  is set to 0 because its expected that all space contained by avoidance grid  $\mathcal{A}(t_i)$  is free.
2. *Trajectory*  $\forall \mathcal{T}(\vec{x}, B) \in \mathcal{R}(\vec{x}, t_i, t_{i+1})$  - For each trajectory segment  $s \subset B$  passing through cell set  $\mathcal{C}(\mathcal{T}(\vec{x}, s))$  the *reachability probability*  $P_R(\mathcal{T}(\vec{x}, s))$  is initialized to 1, due the properties of space contained in  $\mathcal{C}$  (visible, free of obstacles).

$$\begin{aligned}
\textbf{Inputs:} \quad & \text{Intruders} & \mathcal{I} = \{I_1, \dots, I_A\} \\
& \text{Map obstacles} & \mathcal{O}_m = \{o_{m,1}, \dots, o_{m,B}\} \\
& \text{Detected obstacles} & \mathcal{O}_d = \{o_{d,1}, \dots, o_{d,C}\} \\
& \text{Reach set (trajectories)} & \mathcal{R} = \{\mathcal{T}_1, \dots, \mathcal{T}_D\} \\
& \text{Avoidance grid} & \{c_{1,1,1}, \dots, c_{I,J,K}\} \in \mathcal{A}(t_i) \\
\textbf{Initialize:} \quad & \forall c_{i,j,k} \in \mathcal{A}(t_i) & P_O = 0, \quad P_V = 1 \\
& \forall \mathcal{T}_d \in \mathcal{R} & P_R = 1
\end{aligned} \tag{4.19}$$

1. Propagate intruders  $\mathcal{I}$ , map obstacles  $\mathcal{O}_m$ , detected obstacles  $\mathcal{O}_d$ , through avoidance grid  $\mathcal{A}(t_i)$  so each cell  $c_{i,j,k}$  has obstacle  $P_O$  and visibility  $P_V$  probability set.
2. For each trajectory  $\mathcal{T}_d \in \mathcal{R}$  and for each point of trajectory, calculate reachability probability  $P_R$  given by (4.22). (4.20)
3. For each cell  $c_{i,j,k}$  in avoidance grid, there exists set of passing trajectories  $\mathcal{T} \in c_{i,j,k} = \{\mathcal{T}_f \in \mathcal{R} | (\mathcal{T}_f \rightarrow \mathbb{R}^3) \cap c_{i,j,k} \neq \emptyset\}$ .
4. Reachability probability  $P_R$  calculated for each cell  $c_{i,j,k}$  is given by (4.23).

*General algorithm* is given by eq. 4.20. the outputs of algorithm the final reachability probability of trajectories  $\mathcal{T}(\vec{x}, B)$  in reach set approximation  $\mathcal{R}(\vec{x}, t_i, t_{i+1})$  is given by definition 20. When *reachability probability* for all trajectories at all intersection points are assessed, it is possible to assess *reachability probability*  $P_R(c_{i,j,k})$  for all cells  $c_{i,j,k}$  in avoidance grid  $\mathcal{A}(t_i)$ . *Cell reachability probability*  $P_R(c_{i,j,k})$  in avoidance grid  $\mathcal{A}(t_i)$  is given by definition 21.

**Definition 20.** *Reachability probability of trajectory  $\mathcal{T}(\vec{x}, B)$  in reach set approximation  $\mathcal{R}(\vec{x}, t_i, t_{i+1})$  in avoidance grid  $\mathcal{A}(t_i)$  at time of avoidance  $t_i$ . There exists ordered set of passing cells  $\mathcal{C}$  (4.21).*

$$c_{i_1,j_1,k_1}, c_{i_2,j_1,k_1}, \dots, c_{i_n,j_n,k_n} = \mathcal{C} \subset \mathcal{A}(t_i) \tag{4.21}$$

*Passing cells set  $\mathcal{C}$  is ordered and hierarchical  $i_1 \leq i_2 \leq \dots \leq i_n$ ,  $i_1, \dots, i_n \in \mathbb{N}^+, n > 1$ . Each cell  $c_{i,j,k} \in \mathcal{C}$  has assessed obstacle  $P_O(c_{i,j,k})$  and visibility probability  $P_V(c_{i,j,k})$ . Then for trajectory  $\mathcal{T}(\vec{x}, B)$  is probability of reachability  $P_R(\mathcal{T})$  given as (4.22).*

$$P_R(\mathcal{T}) = \prod_{c_{i,j,k} \in \mathcal{C}(\mathcal{T})} (1 - P_O(c_{i,j,k}).P_V(c_{i,j,k})) \tag{4.22}$$

**Definition 21.** *Cell reachability probability  $P_R(c_{i,j,k})$  in avoidance grid  $\mathcal{A}(t_i)$  at time of avoidance  $t_i$  have set of passing trajectories  $\mathcal{T} \in c_{i,j,k}$  with assessed trajectory reachability probability  $P_R(\mathcal{T}(\vec{x}, B))$  (def. 20). Then the final cell reachability probability  $P_R(c_{i,j,k})$ , which is base in cell goal determination, is given as maximum of belonging trajectory reachability probabilities. If there is no belonging trajectory in cell  $c_{i,j,k}$  then final reachability probability is zero.*

$$P_R(c_{i,j,k}) = \begin{cases} |\mathcal{T} \in c_{i,j,k}| > 0 : & \max_{\mathcal{T}_f \in \mathcal{T}} P_R(\mathcal{T}_f) \\ |\mathcal{T} \in c_{i,j,k}| = 0 : & 0 \end{cases} \tag{4.23}$$

## 4.5 Reduced reach set

Common reach set implementation have  $\sum n^k nodes$  where  $n$  is count of possible movements in  $M \subset \mathcal{MA}$  and  $k$  is maximal count of passing layers in avoidance grid  $\mathcal{A}(t_i)$ . Because of *general algorithm* (4.19,4.20), the count of nodes of reach set approximation  $\mathcal{R}$  are impacting the rating calculation performance. The fact that reach set is pre-calculated for prior the mission calculation does not change the complexity of reach set.

*Count of nodes* in trajectory tree approximation is not significant during standard pruning procedure in *deterministic approach*, but it has significant impact on performance in case of tree expanding (rating propagation) calculations. The answer to this is constraint propagation between avoidance grid  $\mathcal{A}(t_i)$  layers  $l_1, l_2, \dots, l_k$ ,  $k \in \mathbb{N}^+$ . The example of constraint propagation in path exploration have been introduced by Shawn in [? ]. Where only nodes with some properties Were expanded in tree search.

The best search results are achieved via *harmonic trees* [? ]. Harmonic tree have medium coverage and its generates smooth trajectories. Cost model for given search have been derived from [? ] work, where the high dimensional space was used. In fact our approach tries to be platform independent therefore this method will fit.

Before various expansion method are defined, the measurement criterion or coverage function must be defined. Our goal is to cover maximum of avoidance capabilities in grid  $\mathcal{A}(t_i)$ . If we have one trajectory  $\mathcal{T}_1$  and we want to define other trajectory  $\mathcal{T}_2$  and give some avoidance capabilities, the trajectory  $\mathcal{T}_2$  should go trough different space segment. It is natural that  $\mathcal{T}_1 \neq \mathcal{T}_2$ . But the passing cell sets  $\mathcal{C}_1(\mathcal{T}_1)$  can be equal to  $\mathcal{C}_2(\mathcal{T}_2)$ . Therefore *trajectory footprint* is defined in 22.

**Definition 22.** *Trajectory footprint* For trajectory  $\mathcal{T}(x_0, B)$ , and avoidance grid  $\mathcal{A}(t_i)$ , while trajectory projection to planar coordinates  $\mathcal{T}_P = \mathcal{T} \rightarrow \mathbb{R}^3$  is bounded by avoidance grid  $\mathcal{A}(t_i)$  space:

$$\begin{aligned} \forall T_P \in \mathcal{T}_P, \quad T_P &= [d, \theta_{T_P}, \varphi_{T_P}], [d_s, d_e, \theta_s, \theta_e, \varphi_s, \varphi_e] \subset \mathcal{A}(T_I) \\ d_s \leq d_{T_P} \leq d_e, \quad \theta_s \leq \theta_{T_P} \leq \theta_e, \quad \varphi_s \leq \varphi_{T_P} \leq \varphi_e \end{aligned} \quad (4.24)$$

Then there exist membership function  $Q$ , which maps trajectory  $\mathcal{T}(x_0, B)$  to finite order of passing cells:

$$Q : \mathcal{T}(x_0, B) \rightarrow \{c_{i_1, j_1, k_1}, \dots, c_{i_n, j_n, k_n}\}, n \in \mathbb{N} \quad (4.25)$$

*Corollary.* Depending movement automaton unitary movements variety, avoidance grid cell size one footprint  $\{c_{i_1, j_1, k_1}, \dots, c_{i_n, j_n, k_n}\}, n \in \mathbb{N}$  can be shared by multiple trajectories.

**Definition 23.** *Trajectory footprints set* For reach set  $\mathcal{R}$  containing all trajectories  $\mathcal{T}_d(x_0, B) \in \mathcal{R}$ , set of unique trajectory footprints can be defined as follow:

$$\mathcal{Q}(\mathcal{R}) = \{Q(\mathcal{T}(x_0, B)) : \mathcal{T}(x_0, B) \in \mathcal{R}\} \quad (4.26)$$

Reach set trajectories footprint is redundant in complete reach set. New reach set calculation method needs to be introduced in order to:

1. Reduce complexity - remove nodes from reduced reach set approximation to keep high value of coverage.
2. Reduce redundancy - remove equal trajectories in terms of trajectory footprint.

Full reach set approximation complexity is  $n + n^2, n^3, \dots, n^m$  nodes where  $m$  is total count of layers in avoidance grid  $\mathcal{A}(t_i)$ .

**Definition 24.** *Coverage ratio  $C_R$  For system  $\dot{x} = f(x, u)$  controlled via movement automaton  $\mathcal{MA}$  with movement set  $M = \{m_1, \dots, m_i\}$ , there exists full reach set  $\mathcal{R}_F$  which covers all movement possibilities inside avoidance grid  $\mathcal{A}(t_i)$ .*

*Let  $\mathcal{Q}_F(\mathcal{R}_F)$  be trajectory footprints set for  $\mathcal{R}_F$ . There exists reduced reach set with  $\mathcal{R}_R$  which does not contain all trajectories.  $\mathcal{Q}_R(\mathcal{R}_R)$  be trajectory footprint set for  $\mathcal{R}_R$ . Then coverage ratio  $C_R$  can be defined as follow:*

$$C_R = \frac{|\mathcal{Q}_R|}{|\mathcal{Q}_F|} \in [0, 1] \quad (4.27)$$

Goal is to minimize count of node of reduced reach set  $\mathcal{R}_R$  while keeping high coverage ratio  $C_R$ . The other factor is *smoothness of trajectory* - how much is trajectory curved and changed during a execution.

**Definition 25.** *Trajectory  $\mathcal{T}(\vec{x}, B)$  smoothness. is defined in context of movement automaton  $\mathcal{MA}$ . The movement set  $M \in \mathcal{MA}$  can be divided into two complement subsets  $S \cup C = M$ ,  $S \cap C = \emptyset$ :*

1. Smooth movement set  $S \subset M$  - non empty set of movements which are considered smooth (fly straight).
2. Chaotic movement set  $C \subset M$  - non empty set of movement which are considered chaotic (sharp left turn).

*Then smoothness rating  $P_S$  of trajectory  $\mathcal{T}(\vec{x}, B)$  is given by following equation:*

$$P_S(\mathcal{T}(\vec{x}, B)) = \frac{\sum_{m_i \in B} \begin{cases} m_i \in S & : 1 \\ m_i \in C & : 0 \end{cases}}{\sum_{m_i \in B} 1} \quad (4.28)$$

*Where nominator is basically count of smooth movements in trajectory movement buffer and denominator is count of all movements. The definition of smoothness rating is flexible to some extent. If trajectory smoothness rating  $P_S$  is high, trajectory  $\mathcal{T}$  is considered smooth.*

### 4.5.1 Constrained expansion procedure

The goal of this section is to introduce constrained expansion procedure in terms of tree expansion in avoidance grid  $\mathcal{A}(t_i)$ .

*Grid layer* is a subset of grid cells  $l_l \in \mathcal{A}(t_i)$ , where all cells  $c_{i_l,j,k} \in l_l$  have same starting  $d_s$  and ending distance  $d_s$ . Layer index  $l \in 1, \dots, a, a \in N^+$  defines order of layer from vehicle perspective.

*Wave-front propagation* is propagation from layer with some index to layer with same index or following index, formally  $l_i \rightarrow (l_i, l_{i+1})$ . Initial vehicle position  $\hat{x}(t_i)$  in avoidance grid  $\mathcal{A}(t_i)$  is considered as layer with index 0 and with 0 cells.

*Tree node*  $\mathcal{N}(\mathbf{x} \subset \mathbb{R}^n, B, J, \mathcal{C})$  is defined as passed trajectory  $\mathbf{x} \subset \mathbb{R}^n$  which is subset of state space  $\mathbb{R}^n$ ,  $B$  is executed set of movements,  $J$  is criterion value,  $\mathcal{C}$  is set of passing cells.

*Root node*  $\mathcal{N}_{root}(\hat{x}(t_i), B = \emptyset, J = 0, \mathcal{C} = \emptyset)$  is defined for initial vehicle state  $\hat{x}(t_i)$  at time of avoidance  $t_i$ , with no executed movements  $B = \emptyset$ , zero initial cost  $J = 0$  and empty passing cells set  $\mathcal{C} = \emptyset$ .

*Simple expansion* of movement buffer  $B_C = \{B_p, m\}, m \in M$  causes expansion of *parent node*  $\mathcal{N}_p(\mathbf{x}_p, B_p, J_p, \mathcal{C}_p)$  to child node  $\mathcal{N}_c(\mathbf{x}_c, B_c, J_c, \mathcal{C}_c)$  (4.29). Main assumptions in expansion are:

1. *Contained trajectories* - Trajectories defined by buffers  $B_p, B_c$  are contained therefore  $\mathcal{T}(\hat{x}, B_p) \cup \mathcal{A}(t_i) = \mathcal{A}(t_i)$  and  $\mathcal{T}(\hat{x}, B_c) \cup \mathcal{A}(t_i) = \mathcal{A}(t_i)$ .
2. *Trajectory uniqueness* - All trajectories starting in root node  $\mathcal{N}_{root}$  are unique, therefore  $\forall B_k \in \mathcal{N}$  where  $\mathcal{N}_{root}$  is root  $\forall i \neq j, B_i \cap B_j \neq \emptyset$ .

$$\begin{aligned} \mathbf{x}_c &= \mathcal{T}_c(\hat{x}(t_i), B_c) = \{\mathbf{x}_p, \mathcal{T}_c(\hat{x}(t_i), B_c) - \mathcal{T}_p(\hat{x}(t_i), B_p)\} \\ B_c &= \{B_p, m\}, m \in M \\ J_c &= J_p + J(\mathcal{T}_c(\hat{x}(t_i), B_c) - \mathcal{T}_p(\hat{x}(t_i), B_p)) \\ \mathcal{C}_c &= \{\mathcal{C}_p, \mathcal{C}(\mathcal{T}_c(\hat{x}(t_i), B_c) - \mathcal{T}_p(\hat{x}(t_i), B_p))\} \end{aligned} \quad (4.29)$$

At the beginning, the root node is initialized  $\mathcal{N}_{root}(\hat{x}(t_i), B = \emptyset, J = 0, \mathcal{C} = \emptyset)$  and expanded for each movement  $m$  in movement set  $M$ . *Propagation algorithm* is executed for each consecutive layer  $l_1 + \dots + l_n$  in following manner:

1. For each cell  $c_{i_l,j,k} \in l_l$  select count of candidates to be expanded via the criterion function select  $\min(J)$  or  $\max(j)$ .
2. Set of selected nodes  $\mathcal{N}_1, \dots, \mathcal{N}_k$  is then expanded for each movement  $m$  in movement set  $M$ .
3. Prune leftover leafs not belonging to layer  $l_l$ . Pruning leftover leafs have been discussed in [? ].

### 4.5.2 Chaotic reach set approximation

Main intent of chaotic reach set computation is to burst coverage ration  $C_R$  (4.27). The expansion cost functional  $J$  (4.30) is considering distance function  $d(\dots)$ , between last passed cell center  $c.center$  and trajectory final point  $\mathcal{T}(\hat{x}, B) \rightarrow \mathbb{R}^3$  as base. From definition of avoidance grid  $\mathcal{A}(t_i)$  its obvious that; when trajectory leads near cell border it has change to significantly increase amount of covered cells after expansion. (Proof of this

claim is omitted). From definition of smoothness function (4.28) the inverted value can be used as penalization coefficients, because sharp movements (chaotic movements  $m \in C$  (def. 25)).

$$J(\mathcal{N}(\mathbf{x}), B, \mathcal{C}) = d(c.\text{center}, \mathcal{T}(\hat{x}, B) \rightarrow \mathbb{R}^3) \times \frac{1}{P_S(\mathcal{T}(\vec{x}, B))} \quad (4.30)$$

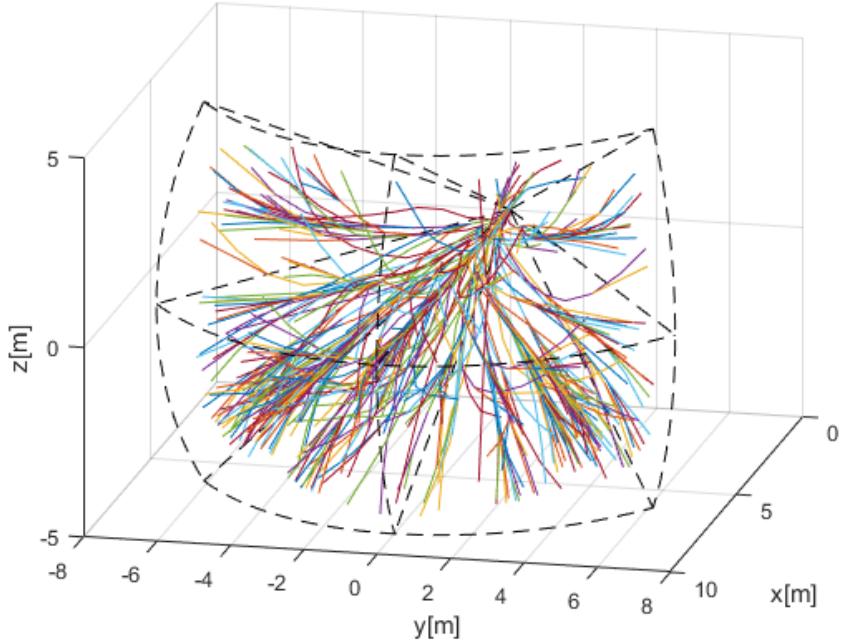


Figure 4.1: Trajectories for chaotic reach set approximation method

The *tuning parameter*  $c$  is spread ratio, which defines how many nodes should be selected in each cell  $c_{i,j,k}$ . Example of reduced reach set approximating is given in fig. 4.1. *Overall method performance* for various avoidance grid sizes:

1. Don't keep trajectories smoothness.
2. Has high coverage.
3. Has high count of nodes.

#### 4.5.3 Harmonic reach set approximation

*Main intention* is to create reduced reach set approximation  $\mathcal{R}_R$ . The *smoothness* of generated trajectories is main priority. It is known that trajectory going through cell center has higher probability to be smooth. The cost functional  $J$  (4.31) is product of smoothness function  $P_S$  (4.28) and inverted distance of last trajectory point  $\mathcal{T}(\hat{x}, B) \rightarrow \mathbb{R}^3$  to last passing cell center  $c.\text{center}$ .

$$J(\mathcal{N}(\mathbf{x}), B, \mathcal{C}) = \frac{1}{d(c.\text{center}, \mathcal{T}(\hat{x}, B) \rightarrow \mathbb{R}^3)} \times P_S(\mathcal{T}(\vec{x}, B)) \quad (4.31)$$

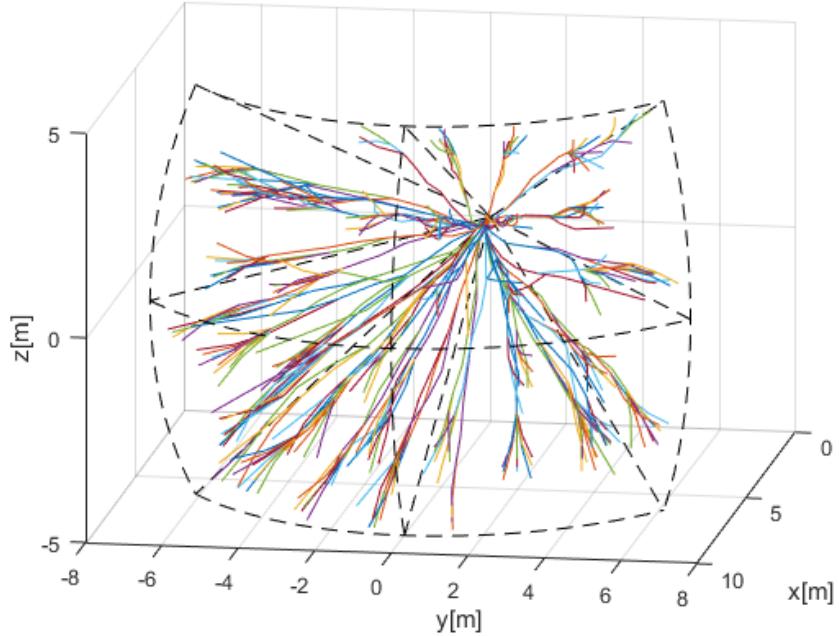


Figure 4.2: Trajectories for harmonic reach set approximation method

*Tuning parameter  $h$*  decides how many nodes  $\mathcal{N}$  in one cell  $c_{i_l,j,k}$  in layer  $l_l \subset \mathcal{A}(t_i)$  will be expanded. The example of harmonic reach set is given in fig. 4.2. One can see that almost all trajectories have long straight (smooth) segments. Overall method performance for various avoidance grid sizes:

1. Keep trajectories smoothness.
2. Has low coverage.
3. Has low count of nodes.

#### 4.5.4 Combined reach set approximation

*Chaotic reach set approximation* (sec. 4.5.2) lacks smoothness of trajectories for navigation without obstacles, it has great coverage ration  $C_R$ . *Harmonic reach set approximation* (sec. 4.5.3) is great for navigation in unconstrained environment, but its coverage capabilities are low.

Because of existence of tuning parameters for *chaotic reach set approximation*  $c$  and *harmonic reach set approximation*  $h$  *Combined method* can be proposed. Using following approach harmonic tree with expansion tuning parameters  $h, c$  can be achieved:

1. Create identical root nodes  $\mathcal{N}_h(\mathbf{x} = \hat{x}(t_i), B = \emptyset, J = 0, \mathcal{C} = \emptyset)$  for harmonic expansion and  $\mathcal{N}_c(\mathbf{x} = \hat{x}(t_i), B = \emptyset, J = 0, \mathcal{C} = \emptyset)$  for chaotic expansion.
2. Expand harmonic root  $\mathcal{N}_h$  using maximum criterion  $J$  (4.31) and tuning parameter  $h$ .
3. Expand chaotic root  $\mathcal{N}_c$  using maximum criterion  $J$  (4.30) and tuning parameter  $c$ .
4. Merge chaotic tree with root  $\mathcal{N}_c$  and harmonic tree with root  $\mathcal{N}_h$ . Merge is possible because both tree structures are compatible and using same structures except expansion criterion (proof omitted). Example of merging multiple trees can be found in [? ].

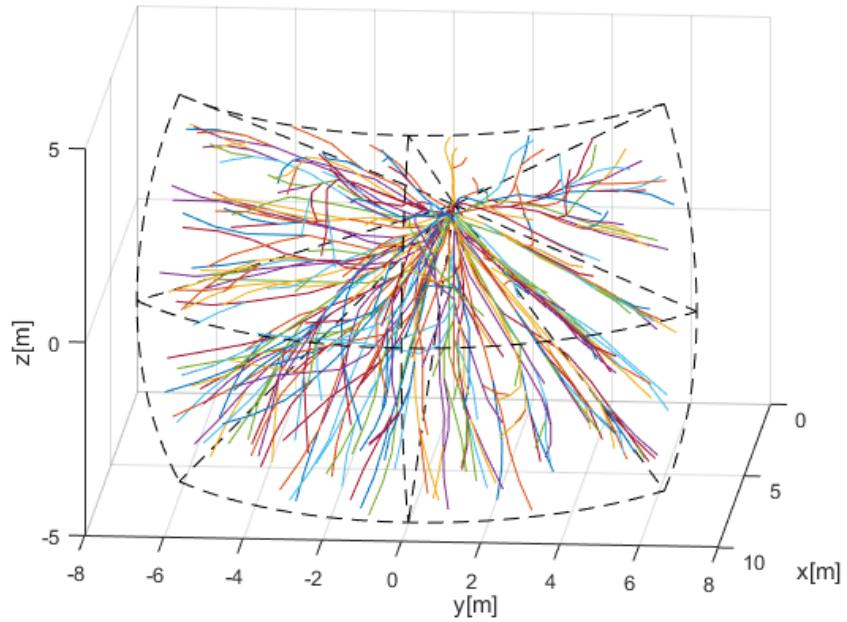


Figure 4.3: Trajectories for combined reach set approximation method

Example of *harmonic reach set approximation* is portrayed in fig. 4.3. Overall method performance for various avoidance grid sizes:

1. Keep trajectories smoothness.
2. Has high coverage.
3. Has medium count of nodes.

# Chapter 5

## Probabilistic calculations

### 5.1 Intruder intersection model

**Intruder model** is basically about intersection of 3D objects, because of definition of avoidance grid  $\mathcal{A}(t_i)$  most of these methods are numeric. Let us introduce intruder model  $\dot{\vec{x}}_I = \vec{f}(\vec{x}_I, \vec{u}_I)$  is given by following equation:

$$\begin{aligned}\dot{p}_x &= v_x & p_x(t) &= p_x(0) + v_x \cdot t \\ \dot{\vec{x}}_I = \dot{p}_y &= v_y; \quad \vec{x}_I(0) = \begin{bmatrix} p_x(0) \\ p_y(0) \\ p_z(0) \end{bmatrix} & p_y(t) &= p_y(0) + v_y \cdot t \\ \dot{p}_z &= v_z & p_z(t) &= p_z(0) + v_z \cdot t\end{aligned}\quad (5.1)$$

Position vector  $[p_x, p_y, p_z]^T$  ( $m, m, m \in \mathbb{R}^3$ ) is given in local coordinate frame of avoidance grid  $\mathcal{A}(t_i)$ . Velocity vector  $[v_x, v_y, v_z]$  ( $m.s^{-1}, m.s^{-1}, m.s^{-1}$ ) is *estimated and not changing*. The time  $t$  is in interval  $[t_e, t_l]$ , where  $t_e$  is intruder entry time into avoidance zone and  $t_l$  is intruder leave time from avoidance zone, if vehicle is considered, time of entry is marked as  $i_{e,k}$  where  $k$  is intruder identification, time of leave is marked as  $i_{l,k}$  where  $k$  is intruder identification.

**Cell entry and leave times**  $t_e(c_{i,j,k})$  and  $t_l(c_{i,j,k})$  are given by trajectory  $\mathcal{T}(x_0, B)$  and cell space  $c_{i,j,k} \in \{x \in \mathbb{R}^3 : d_s \leq x(1) \leq d_e, \theta_s \leq x(2) \leq \theta_e, \varphi_s \leq x(3) \leq \varphi_e\}$ . From trajectory definition there exist projection  $f : \mathcal{T}(x_0, B) \rightarrow (\mathbb{R}^3) \times \mathbb{R}$  to avoidance grid coordinate space  $x \in \mathbb{R}^3$  and avoidance grid time  $t \in \mathbb{R}$ . Then intersection of trajectory and cell with time is given as:

$$(\mathcal{T}(x_0, B) \rightarrow (\mathbb{R}^3) \times \mathbb{R}) \cap (c_{i,j,k} \times (-\infty, \infty)) = \{\vec{x}_e, \dots, \vec{x}_l\} \times \{t_e, \dots, t_l\} \quad (5.2)$$

Where  $t_e$  is cell  $c_{i,j,k}$  entry time given by function  $f_{t_e}(c_{i,j,k}, \mathcal{T}(x_0, B))$ ,  $t_l$  is cell leave time given by function  $f_{t_l}(c_{i,j,k}, \mathcal{T}(x_0, B))$ . Entry time  $t_e$  and leave time  $t_l$  for cell  $c_{i,j,k}$  where  $\mathcal{T}(c_{i,j,k})$  is set of trajectories are given by eq. (5.3) and (5.4).

$$t_e = f_{t_e}(c_{i,j,k}) = \begin{cases} -\infty & : |\mathcal{T}(c_{i,j,k})| = 0 \\ \min f_{t_e}(c_{i,j,k}, \mathcal{T}(x_0, B)), \forall \mathcal{T}(x_0, B) \in \mathcal{T}(c_{i,j,k}) & : |\mathcal{T}(c_{i,j,k})| \neq 0 \end{cases} \quad (5.3)$$

$$t_l = f_{t_l}(c_{i,j,k}) = \begin{cases} \infty & : |\mathcal{T}(c_{i,j,k})| = 0 \\ \max f_{t_l}(c_{i,j,k}, \mathcal{T}(x_0, B)), \forall \mathcal{T}(x_0, B) \in \mathcal{T}(c_{i,j,k}) & : |\mathcal{T}(c_{i,j,k})| \neq 0 \end{cases} \quad (5.4)$$

**Intruder intersection probability**  $P_{OI}(i_k)$  for intruder  $i_k$  intersection with vehicle in cell  $c_{i,j,k}$  is given by eq. (5.5).

$$P_{OI}(i_k) = \frac{\|[i_e(c_{i,j,k}), i_l(c_{i,j,k})] \cap [t_e, t_l]\|}{\|[t_e, t_l]\|} \cdot P_T(i_k, c_{i,j,k}) \quad (5.5)$$

Base intersection probability  $P_T(i_k, c_{i,j,k})$  shows how probable is to meet intruder  $i_k$  in cell  $c_{i,j,k}$ , enumerator  $\|[i_e(c_{i,j,k}), i_l(c_{i,j,k})] \cap [t_e, t_l]\|$  represents time duration which intruder  $i_k$  and our vehicle spends together in cell  $c_{i,j,k}$ , where  $i_e$  is intruder entry,  $i_l$  is intruder leave,  $t_e$  is vehicle entry,  $t_l$  is vehicle leave, denominator  $\|[t_e, t_l]\|$  is maximum duration of vehicle passing through cell  $c_{i,j,k}$ . The main principle is to calculate probability of staying in same portion of space for vehicle and intruder  $i_k$ . In case the numeric approximation is used and the result is set of intersection times/position pairs  $\{(t_1, \vec{p}_1), \dots, (t_k, \vec{p}_k)\}$  where position set  $\{\vec{p}_1, \dots, \vec{p}_k\} \in c_{i,j,k}$  then (5.5) is re-written as:

$$P_{OI}(i_k) = \frac{|\{t_1, \dots, t_k\} \cap [t_e, t_l]|}{k} \cdot P_T(i_k, c_{i,j,k}), \quad \{t_1, \dots, t_k\}, k \in \mathbb{N}^+ \quad (5.6)$$

Base intersection probability  $P_T(i_k, c_{i,j,k})$  stays same it is not impacted. The time intersection probability is calculated differently, enumerator  $|\{t_1, \dots, t_k\} \cap [t_e, t_l]|$  is resulting into count of how many numeric hits landed vehicle entry/leave interval  $[t_e, t_l]$ . Denominator is simple count of hits which landed into cell  $c_{i,j,k}$ . The final ratio shows how probable is to encounter intruder  $i_k$  in cell  $c_{i,j,k}$ . The principle of (5.6) is same as (5.5) but instead of interval intersection, set intersection is used.

**Base intersection probability**  $P_T(i_k, c_{i,j,k})$  reflects probability of vehicle intersection with portion of space occupied by cell  $c_{i,j,k}$ , to be precise with vehicle trajectory or vehicle body shifted among that trajectory. This work considers two trajectories representation:

1. *Direct line* - vehicle trajectory is represented as basic linear model (5.1), depending on average cell volume and safety margin condition direct line can have two intersection modes:
  - a. *simple line* - intersection is approximated along line defined by intruder model (5.1).
  - b. *volume line* - intersection is approximated along line defined by intruder model (5.1) and intruder radius  $r_i$  [m] is considered in intersection.
2. *Elliptic cone* - initial position  $\vec{x}_i(0)$  is considered as top of the cone, main cone axis is given by line (5.1)  $t \in [-\infty, \infty]$ . The cone is defined by horizontal spread parameter  $\theta$  and vertical spread  $\varphi$ .

**Base intersection probability - simple line type**  $P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k})$  is simple linear intersection with line defined as expected intruder  $i_r$  trajectory (5.7). Intruder trajectory is defined by starting point  $\vec{x}_s \in \mathbb{R}^3$  and linear velocity vector  $\vec{v} \in \mathbb{R}^3$ .

$$\vec{x}(t) = \vec{x}_s + \vec{v}_I \cdot t \quad (5.7)$$

Then there exist projection function from local euclidean coordinates to local planar coordinates  $p(t)$  (5.8). Which projects trajectory of the system  $\vec{x}(t) \in \mathbb{R}^3$  to planar

coordinates  $[d(t), \theta(t), \varphi(t)]$ . Where  $d$  is distance,  $\theta$  is horizontal offset angle from main axis, and  $\varphi$  is vertical offset angle from main axis.

$$\vec{p}(t) : \vec{x}(t) \rightarrow [d(t), \theta(t), \varphi(t)] \quad (5.8)$$

The *base intersection probability*  $P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k})$  for line type is given as eq. 5.9. If there exist non empty intersection of  $p(t) \cap c_{i,j,k}$  there is direct collision probability, if intersection  $p(t) \cap c_{i,j,k} = \emptyset$  then probability is zero.

$$P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k}) = \begin{cases} 1 & \exists d(t), \theta(t), \varphi(t), [d(t), \theta(t), \varphi(t)]^T \in c_{i,j,k} \\ 0 & \text{otherwise} \end{cases} \quad (5.9)$$

**Base intersection probability - volume line type**  $P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k})$  is defined for linear intersection model  $\vec{x}(t)$  (5.10) where initial position  $\vec{x}_s$  is given at expected time of avoidance  $t_i$  in avoidance grid  $\mathcal{A}(t_i)$  local coordinate frame, velocity vector  $\vec{v}$  is given in same reference frame.

$$\vec{x}(t) = \vec{x}_s + \vec{v}.t \quad (5.10)$$

*Volume ball*  $\mathcal{B}(\vec{x}(t), r)$  (5.11) is defined as set of points in  $\mathbb{R}^3$  space with defined center at intruder  $i_r$  position  $\vec{x}(t)$ . The inclusion criterion is set to a full ball. Volume ball  $\mathcal{B}(\vec{x}(t), r)$  represents approximation of vehicle body expected position at time  $t$ .

$$\mathcal{B}(\vec{x}(t), r) = \{\vec{a} \in \mathbb{R}^3 : \exists t, \|\vec{a} - \vec{x}(t)\|_2 \leq r\} \quad (5.11)$$

*Planar volume ball*  $\mathcal{P}_{\mathcal{B}}$  (5.12) is projection of volume ball  $\mathcal{B}(\vec{x}(t))$  to a set of planar coordinates  $\{[d, \theta, \varphi]\}$  in avoidance grid  $\mathcal{A}(t_i)$  local coordinate frame for expected avoidance time  $t_i$ .

$$\mathcal{P}_{\mathcal{B}} = \mathcal{B}(\vec{x}(t), r) \rightarrow \{[d, \theta, \varphi]\} \quad (5.12)$$

*Base intersection probability for vehicle body*  $P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k})$  5.13 is calculated as intersection of planar volume ball  $\mathcal{P}_{\mathcal{B}}$  and cell  $c_{i,j,k}$ . If intersection  $\mathcal{P}_{\mathcal{B}} \cap c_{i,j,k} \neq \emptyset$  is non empty then base probability is one, zero otherwise.

$$P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k}) = \begin{cases} 1 & \mathcal{P}_{\mathcal{B}} \cap c_{i,j,k} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (5.13)$$

**Base intersection probability - elliptic cone type**  $P_T(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  computation is less straight-forward than other base intersection probabilities. First let us define linear intruder  $i_k$  positions  $\vec{x}$  at time  $t$  (5.14) model, where  $\vec{x}(t)$  defines vehicle position in *vehicle local coordinate frame* at time of avoidance,  $\vec{v}$  defines intruder velocity, and  $t$  is time offset.

$$\vec{x}(t) = \vec{x}_s + \vec{v}_I.t \quad (5.14)$$

Intruder *horizontal spread*  $\theta$  and *vertical spread*  $\varphi$  are introduced. These spreads represents intruder deviation limits from original trajectory  $\vec{x}(t) \in \mathbb{R}^3$ . The example is given by fig. 5.1 where intruder starts at point  $\vec{x}_s$  with some velocity  $v$ , the linear trajectory approximation is given by blue line. *Intruder position* at time  $t = 10s$  is given by  $\vec{x}(10)$  and is denoted blue point. The ellipsoidal space  $E(\vec{x})$  is projected on plane  $D(\vec{x}(t))$ . The plane  $D$  (5.15) for point  $\vec{x}(t)$  and velocity  $\vec{v}$  is defined as orthogonal plane to velocity  $\vec{v} \in \mathbb{R}^3$  with origin at intruder position  $\vec{x}(t)$ .

$$D(\vec{x}(t), \vec{v}) = \{\vec{a} \in \mathbb{R}^3 : (\vec{a} - \vec{x}(t)) \perp \vec{v}, \} \quad (5.15)$$

To introduce ellipsoidal space on orthogonal plane  $D(\vec{x}(t), \vec{v})$  some parameters are introduced in eq. (5.16). The *scalar distance*  $d_d(\vec{x}(t))$  is simple euclidean norm, *maximal horizontal offset*  $d_\theta(\vec{x}(t))$  is given as product of sinus of horizontal offset angle  $\theta$  and scalar distance  $d_d$ , and *maximal vertical offset*  $d_\phi(\vec{x}(t))$  is given a product of sinus of vertical offset angle  $\varphi$  and scalar distance  $d_d$ .

$$\begin{aligned} d_d &= d_d(\vec{x}(t), \vec{x}_s) = \|\vec{x}(t) - \vec{x}_s\|_2 \\ d_{\theta_{\max}} &= d_\theta(\vec{x}(t)) = \sin \theta(i_k) \cdot d_d(\vec{x}(t)) \\ d_{\phi_{\max}} &= d_\phi(\vec{x}(t)) = \sin \varphi(i_k) \cdot d_d(\vec{x}(t)) \end{aligned} \quad (5.16)$$

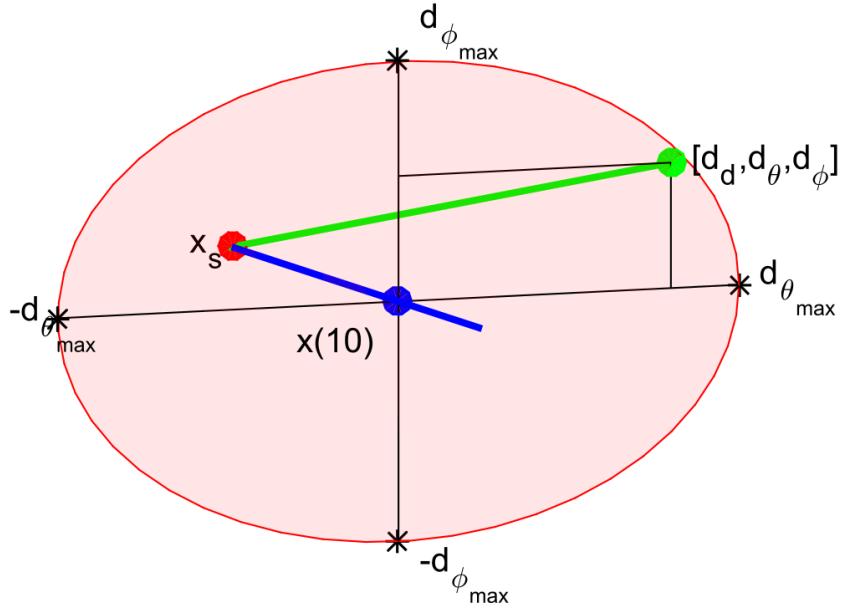


Figure 5.1: One probable position  $[d_d, d_\theta, d_\phi]$  (green). deviated from linear trajectory (blue line) at point  $\vec{x}(10)$ .(blue) with initial position  $x_s$  (red)

*Ellipsoid*  $E(\vec{x}(t), \vec{v})$  (5.17) for some intruder position  $\vec{x}(t)$  and intruder velocity  $\vec{v}$  is given as constrained subspace of orthogonal plane  $D(\vec{x}(t), \vec{v})$ . The constraint is defined by internal coordinate frame  $\vec{p} \in \mathbb{R}^2$  which is space reduction of plane  $D(\vec{x}(t), \vec{v})$ . Internal coordinate frame  $\vec{p} \in \mathbb{R}^2$  has origin in  $\vec{x}(t) \rightarrow \mathbb{R}^2$ . The points of plane  $\vec{p}$  are bounded by projection  $\vec{p} = (\vec{b} - \vec{x}(t)) \rightarrow \mathbb{R}^2$ , where  $\vec{b} \in D(\vec{x}(t), \vec{v})$ . The point of ellipsoidal  $\vec{p}$  is then given as standard ellipse boundary with vertical span  $d_\theta(\vec{x}(t))$  and horizontal span  $d_\phi(\vec{x}(t))$ . *Ellipsoid*  $E(\vec{x}(t), \vec{v})$  for specific time  $t = 10s$  example is portrayed as red elipsoind in fig. 5.1.

$$E(\vec{x}(t), \vec{v}) = \left\{ \vec{b} \in \mathbb{R}^3 : \vec{b} \in D(\vec{x}(t), \vec{v}), \vec{p} = (\vec{b} - \vec{x}(t)) \rightarrow \mathbb{R}^2, \left( \frac{p(1)^2}{d_\theta(\vec{x}(t))^2} + \frac{p(2)^2}{d_\phi(\vec{x}(t))^2} \right) \leq 1 \right\} \quad (5.17)$$

The expected behaviour of intruder  $i_k$  is to stick to linear trajectory  $\vec{x}(t)$  (5.14). The probability of deviation should be decreasing with distance from ellipse center (fig. 5.2.).

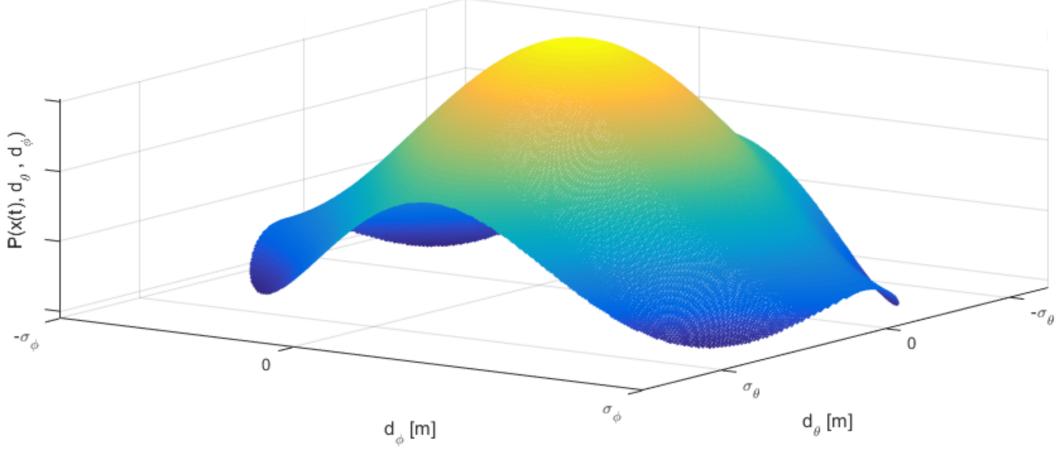


Figure 5.2: Probability of intruder  $i_k$  position in ellipsoid  $E(\vec{x}(t), \vec{v})$

*Probability density function* for ellipsoid  $E(\vec{x}(t), \vec{v})$  defined in (5.17) is depending on maximal horizontal spread  $d_\theta(\vec{x}(t))$ , maximal vertical spread  $d_\varphi(\vec{x}(t))$ , defined by eq. 5.16. Two standard probabilistic distributions are established  $\mathcal{N}(\mu_\theta, \sigma_\theta)$  (5.18) for horizontal spread  $\theta(\vec{x}(t))$  and  $\mathcal{N}(\mu_\varphi, \sigma_\varphi)$  (5.19) for vertical spread  $\varphi(\vec{x}(t))$ . The means  $\mu_\theta$  and  $\mu_\varphi$  are set to zero, and internal coordinate frame  $\vec{p} \in \mathbb{R}^2$  where  $\vec{x}(t) \rightarrow \mathbb{R}^2$  is frame center. The variances  $\sigma_\theta$  and  $\sigma_\varphi$  are set as maximal distances on horizontal/vertical spread axes  $d_\theta(\vec{x}(t))$  and  $d_\varphi(\vec{x}(t))$ .

$$P(\vec{x}(t), d_\theta) = \mathcal{N}(\mu_\theta, \sigma_\theta) = \mathcal{N}(0, d_\theta(\vec{x}(t))) \quad (5.18)$$

$$P(\vec{x}(t), d_\varphi) = \mathcal{N}(\mu_\varphi, \sigma_\varphi) = \mathcal{N}(0, d_\varphi(\vec{x}(t))) \quad (5.19)$$

Combined *probability density function* for maximal spreads  $d_\theta$  and  $d_\varphi$  is given by eq. 5.20. Because probability density function is defined for internal space  $\vec{p} \in \mathbb{R}^2$  and one may need to calculate probability for for cell space  $c_{i,j,k} \in \mathbb{R}^3$ , the reduction from two parameter probability distribution function to scalar probability distribution function is needed. Scalar probabilistic distribution function  $P(\vec{x}(t), d_\theta, d_\varphi)$  over ellipsoid  $E(\vec{x}(t), \vec{v})$  is introduced (5.20), where final probability is given as average of two partial probabilities. Final  $P(\vec{x}(t), d_\theta, d_\varphi)$  needs to be normalized to hold *normal distribution condition* (5.21). Normal distribution condition value (5.21) is given as surface integral over ellipsoid  $E(\vec{x}(0), \vec{v})$  with probability density function  $P(\vec{x}(t), d_\theta, d_\varphi)$ .

$$P(\vec{x}(t), d_\theta, d_\varphi) = \frac{\mathcal{N}(\mu_\theta, \sigma_\theta) + \mathcal{N}(\mu_\varphi, \sigma_\varphi)}{2} \quad (5.20)$$

$$\iint_{E(\vec{x}(\tau))} P(\vec{x}(t), d_\theta, d_\varphi) dd_\theta dd_\varphi = 1 \quad (5.21)$$

Final intersection probability  $P(\vec{x}(t), c_{i,j,k}, \theta, \varphi)$  (static part, timed is calculated in (5.5) is given by eq. 5.23. Its mean value of all intersection probabilities  $P(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi)$  where  $\tau \in [i_e(c_{i,j,k}), i_l(c_{i,j,k})]$  is fixed point in intersection time interval.  $P(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi)$  (5.22) is integration of probability density function  $P(\vec{x}(\tau), d_\theta, d_\varphi)$  (5.20) in surface  $E(\vec{x}(\tau), \vec{v})$  to cell  $c_{i,j,k}$  volume intersection. To get volume integration partial probability 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*  $P_T(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  is given by eq. 5.23. Example of

intersection of intruder  $i_r$  uncertain ellipsoid cone with avoidance grid  $\mathcal{A}(t_i)$  is given in fig. 5.3.

$$P(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi) = \iint_{E(\vec{x}(\tau), \vec{v}) \cap c_{i,j,k}} P(\vec{x}(\tau), d_\theta, d_\varphi) \quad (5.22)$$

$$P_T(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k}) = \frac{\int_{i_e(c_{i,j,k})}^{i_l(c_{i,j,k})} P(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi) d\tau}{i_l(c_{i,j,k}) - i_e(c_{i,j,k})} \quad (5.23)$$

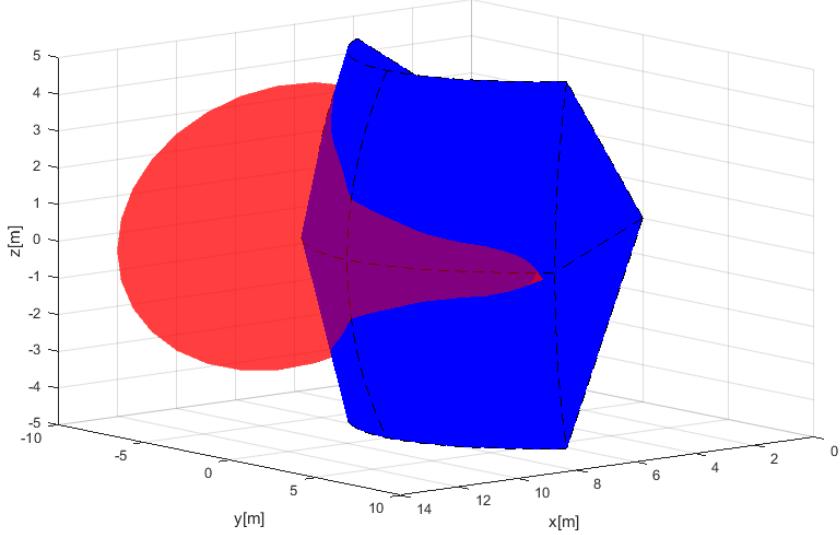


Figure 5.3: Avoidance grid  $\mathcal{A}(t_i)$  (blue) intersection with elliptic cone intruder  $i_k(\vec{x}, \vec{v}, \theta, \varphi)$  (red) example.

*Numeric approximation* of  $P_T(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  is more 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  $\mathcal{R}$  (5.24) 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  $\mathcal{R}^2 \subset \mathbb{R}^2, \mathcal{R}^3 \subset \mathbb{R}^3, \dots \mathcal{R}^i \subset \mathbb{R}^i, i \in \mathbb{N}^+$  keeps homogeneous distribution.

$$\mathcal{R} = \left\{ a \in \mathbb{R} : \forall i \in \mathbb{Z}, |i < a \leq i + 1| = k, k \in \mathbb{N}^+, \right. \\ \left. \forall j \in \mathbb{N}^+ a_{j+1} - a_j = m, m \in \mathbb{R}^+ \right\}, \mathcal{R} \subset \mathbb{R} \quad (5.24)$$

Orthogonal plane for  $\vec{x}(t), \vec{v}, t \in \mathbb{R}$  is defined by eq. 5.15. The orthogonality property is also kept for any subspace  $\mathcal{R}^n \in \mathbb{R}^n, n \in \mathbb{N}^+$ . Numeric approximation of  $D(\vec{x}(t), \vec{v})$  is given as  $D_D(\vec{x}(t), \vec{v})$  (5.25). 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.

$$D_D(\vec{x}(t), \vec{v}) = \{ \vec{a} \in \mathcal{R}^3 : (\vec{a} - \vec{x}(t)) \perp \vec{v}, \} , t \in \mathcal{R} \quad (5.25)$$

*Base ellipsoid*  $E(\vec{x}(t), \vec{v})$  for continuous-space is given by eq. 5.17. Every aspect, except the base of internal projection  $\mathcal{R}^2$  and orthogonal plane  $D_D$  is same in discrete case  $E_D(\vec{x}(t), \vec{v})$  (5.26).

$$\bar{E}_D(\vec{x}(t), \vec{v}) = \left\{ \begin{array}{l} \vec{b} \in \mathcal{R}^3 : \vec{b} \in D_D(\vec{x}(t), \vec{v}), \vec{p} = (\vec{b} - \vec{x}(t)) \rightarrow \mathcal{R}^2, \\ \left( \frac{p(1)^2}{d_\theta(\vec{x}(t))^2} + \frac{p(2)^2}{d_\varphi(\vec{x}(t))^2} \right) \leq 1 \end{array} \right\}, t \in \mathcal{R} \quad (5.26)$$

*Numeric disproportion* can occur in case that ellipsoid  $\bar{E}_D(\vec{x}(t), \vec{v})$  (5.26) in case of  $d_\theta(\vec{x}(t)) \approx 0$  and  $d_\varphi(\vec{x}(t)) \approx 0$ . The count of ellipsoid members can be  $|\bar{E}_D(\vec{x}(t), \vec{v})| = 0$ , which is in contradiction with assumption  $|\bar{E}_D(\vec{x}(t), \vec{v})| \neq 0$ . Let assume for discrete times  $\tau = \{t_1, t_2, \dots, t_i\}$ ,  $i \in \mathbb{N}^+$  there exists ellipsoids  $\bar{E}_D(\vec{x}(t_1), \vec{v}), \bar{E}_D(\vec{x}(t_1), \vec{v}), \dots, \bar{E}_D(\vec{x}(t_i), \vec{v})$  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:

$$\bar{E}_D(\vec{x}(t_1), \vec{v}) \cap \bar{E}_D(\vec{x}(t_2), \vec{v}) \cdots \cap \dots \bar{E}_D(\vec{x}(t_i), \vec{v}) = \emptyset \quad (5.27)$$

*Empty intersection* enables us to keep homogeneity property of ellipsoids by adding points so it is safe to add specific point  $\vec{x}(t)$  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 combined probability density function  $P(\vec{x}, d_\theta, d_\varphi)$ . The final ellipsoid used forward  $E_D(\vec{x}(t), \vec{v})$  (5.28) is keeping all properties of ellipsoid  $E(\vec{x}(t), \vec{v})$  (5.28).

$$E_D(\vec{x}(t), \vec{v}) = \begin{cases} |\bar{E}_D(\vec{x}(t), \vec{v})| = 0 & : \{\vec{x}(t)\} \\ |\bar{E}_D(\vec{x}(t), \vec{v})| \geq 0 & : \bar{E}_D(\vec{x}(t), \vec{v}) \end{cases} \quad (5.28)$$

Normal distribution condition for probability function  $P_D(\vec{x}(t), d_\theta, d_\varphi, \vec{p})$ , which is instance of to probability density function  $P(\vec{x}(y), d_\theta, d_\varphi)$  (5.20) is used. This probabilistic distribution must be normalized according to eq. 5.29.

$$\sum_{\vec{p} \in E_D(\vec{x}(t))} P_D(\vec{x}(t), d_\theta, d_\varphi, \vec{p}) = 1, \forall t \in \mathcal{R}^+ \quad (5.29)$$

Equations for *base intersection probability* are similar to eq. 5.22, 5.23. For cell  $c_{i,j,k}$  there exist intruder entry time  $i_e(c_{i,j,k})$  its the earliest intersection with ellipsoid  $E_D(\vec{x}(i_e(c_{i,j,k}))), \vec{v}$ . 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  $\vec{p} \in E_D$ . Based on system dynamic (5.1) the *Time Of Arrival* (TOA) can be calculated. The example of TOA is given in fig. 5.4.

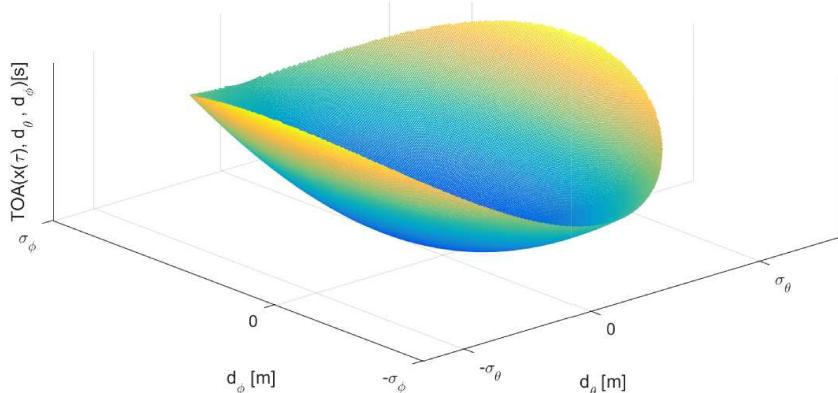


Figure 5.4: Time Of Arrival (TOA) for one ellipsoid  $E_D(\vec{x}(\tau), \vec{v})$

The intersection probability  $P_D(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi)$  for one time sample  $\tau$  is given by eq. 5.30, which has similar notation to eq. 5.22, sums are used instead of integrals and discrete probability density function  $P_D(\vec{x}(\tau), d_\theta, d_\varphi, \vec{p})$  for points from ellipse and cell intersection are used as iterator base set  $\vec{p} \in \{E_D(\vec{x}(\tau), \vec{v}) \cap c_{i,j,k}\}$ .

$$P_D(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi) = \sum_{\vec{p} \in \{E_D(\vec{x}(\tau), \vec{v}) \cap c_{i,j,k}\}} P_D(\vec{x}(\tau), d_\theta, d_\varphi, \vec{p}) \quad (5.30)$$

The *base intersection probability*  $P_{TD}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  5.31 is given as mean intersection probability of partial intersections  $P_D(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi)$  where step set  $T = \{i_e(c_{i,j,k}), \dots, i_l(c_{i,j,k})\}$  contains all viable intersection times with ellipsoids  $E(\vec{x}(\tau \in T), \vec{v})$ . The denominator is basically count of samples in sample time set  $T$ .

$$P_{TD}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k}) = \frac{\sum_{\tau=i_e(c_{i,j,k})}^{i_l(c_{i,j,k})} \sum_{\vec{p} \in E_D(\vec{x}(\tau), \vec{v})} P_D(\vec{x}(\tau), c_{i,j,k}, \theta, \varphi, \vec{p})}{\sum_{\tau=i_l(c_{i,j,k})}^1 1} \quad (5.31)$$

*Intersection of intruder cone and cell  $c_{i,j,k}$*  cell is defined by eq. 5.32. its a set of point  $\vec{p} \in \mathcal{R}^3$  where condition of intersection between ellipsoids  $E_D(\vec{x}(\tau), \vec{v})$  for times  $\tau \in \mathcal{R}^+$  and cell space  $c_{i,j,k}$  is met.

$$\mathcal{P}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k}) = \bigcup_{\forall \tau \in \mathcal{R}^+} \{\vec{p} \in \mathcal{R}^3 : \vec{p} \in c_{i,j,k} \cap E_D(\vec{x}(\tau), \vec{v})\} \quad (5.32)$$

*Intruder time of entry  $i_e(i_k, c_{i,j,k})$*  (5.33), for intruder  $i, k$  and cell  $c_{i,j,k}$  is approximated for discrete point set  $\mathcal{P}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  (5.32) as minimal time of arrival  $t_{TOA}(\vec{p})$  of member points  $\vec{p}$ .

$$i_e(i_k, c_{i,j,k}) \approx \min \{t_{TOA}(\vec{p}) : \vec{p} \in \mathcal{P}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})\} \quad (5.33)$$

*Intruder time of leave  $i_l(i_k, c_{i,j,k})$*  (5.34), for intruder  $i, k$  and cell  $c_{i,j,k}$  is approximated for discrete point set  $\mathcal{P}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  (5.32) as maximal time of arrival  $t_{TOA}(\vec{p})$  of member points  $\vec{p}$ .

$$i_l(i_k, c_{i,j,k}) \approx \max \{t_{TOA}(\vec{p}) : \vec{p} \in \mathcal{P}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})\} \quad (5.34)$$

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

1. *Starting position  $\vec{x}_s$*  - 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  $\vec{v}$*  - oriented velocity of intruder  $i_r$  at time of avoidance  $t_i$  in avoidance grid frame  $\mathcal{A}(t_i)$ .
3. *Horizontal uncertainty spread  $\theta$*  - 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*  $\varphi$  - 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.

*Flag vector*  $l, b, s, \tau \in \{0, 1\}$  is parametrization of probabilistic calculation:  $l$  stands for *lined intersection*,  $b$  stands for *body intersection*,  $s$  stands for *spread intersection*,  $\tau$  stands for *timed intersection*.

*Base intersection probability for line*  $P_L(i_k, c_{i,j,k})$  is defined as  $P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k})$ , where  $i_k$  is intruder with properties of initial position  $\vec{x}$ , velocity vector  $\vec{v}$  and  $c_{i,j,k}$  is target cell. (5.9). *Base intersection probability for body*  $P_B(i_k, c_{i,j,k})$  is defined as  $P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k})$  (5.13), where intruder  $i_r$  has additional property of the intruder body volume  $r$ . *Base intersection probability for spread*  $P_S(i_k, c_{i,j,k})$  is defined as  $P_{TD}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  (5.31), where intruder properties  $\theta, \varphi$  stands for intruder horizontal and vertical uncertainty spread.

*Time based intersection probability*  $P_{\tau,x}(i_k, c_{i,j,k}) \in [0, 1]$  is defined in eq. 5.5. This probability has two calculation modes, first is for 1D intersection (line), second is for volume intersection (body volume, spread elliptic cone). Cell entry  $t_e$  and leave  $t_l$  time for vehicle in avoidance grid  $\mathcal{A}(t_i)$  are given by eq. 5.3 and 5.4. 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. 5.33 and 5.34. All partial probabilities with respective definition references are summarized in eq. 5.35

$$P_L(i_k, c_{i,j,k}) = P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k}) \quad (5.9)$$

$$P_B(i_k, c_{i,j,k}) = P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k}) \quad (5.13)$$

$$P_S(i_k, c_{i,j,k}) = P_{TD}(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k}) \quad (5.31) \quad (5.35)$$

$$P_{\tau,x}(i_k, c_{i,j,k}) = \frac{\|[i_e(c_{i,j,k}), i_l(c_{i,j,k})] \cap [t_e, t_l]\|}{\|[t_e, t_l]\|} \quad (5.5)$$

With definition of all base and timed probabilities (5.35) and given flag vector  $l, b, s, \tau \in \{0, 1\}$  one can formulate final intersection probability  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.36) for intruder  $i_k$  and cell  $c_{i,j,k}$ . The principle is following: *maximum of selected probabilities product based on flag vector is final probability of intruder  $i_k$  in cell*. The time-based flag  $\tau$  is adding timed probability  $P_{\tau,x}(i_k, c_{i,j,k})$ , where time intersection probability is defined by  $x = \{L, B, S\}$  for line, body volume, spread ellipse time intersections ( $P_{\tau,L}(i_k, c_{i,j,k}) \neq P_{\tau,B}(i_k, c_{i,j,k}) \neq P_{\tau,S}(i_k, c_{i,j,k})$  for one intruder  $i_k$ ).

$$P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau) = \begin{cases} \tau = 0 & : \max \left\{ \begin{array}{l} P_L(i_k, c_{i,j,k}).l \\ P_B(i_k, c_{i,j,k}).b \\ P_S(i_k, c_{i,j,k}).s \end{array} \right\} \\ \tau = 1 & : \max \left\{ \begin{array}{l} P_{\tau,L}(i_k, c_{i,j,k}).P_L(i_k, c_{i,j,k}).l \\ P_{\tau,B}(i_k, c_{i,j,k}).P_B(i_k, c_{i,j,k}).b \\ P_{\tau,S}(i_k, c_{i,j,k}).P_S(i_k, c_{i,j,k}).s \end{array} \right\} \end{cases} \quad (5.36)$$

**Intruder obstacle probability**  $P_{O_I}(c_{i,j,k})$  is defined for set of intruders  $\mathcal{I}(\mathcal{A}(t_i))$  containing all detected intruders  $i_r \in \mathcal{I}$  in avoidance grid  $\mathcal{A}(t_i)$  for time of avoidance  $t_i$ . The equation 5.37 is simple inverted cumulative product of partial intruder in cell probabilities  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.36). The principle is simple, if there is one intruder in cell, cumulative probability remains the same  $1 - (1 - P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)) = P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$ . If there is multiple intruders  $i_1, i_2, \dots, i_k, k \in \mathbb{N}^+$ , the probability of collision is geometrically increasing, even with small probability for some intruders  $i_k$ .

$$P_{O_I}(c_{i,j,k}) = 1 - \left( \prod_{i_k \in \mathcal{I}(\mathcal{A}(t_i))} 1 - P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau) \right) \quad (5.37)$$

## 5.2 Detected obstacle probability

Probability of detected obstacle  $P_{O_D}$  describes how probable is to encounter detected obstacle in cell  $c_{i,j,k}$ . Probability of detected obstacle is merged information. Let say that obstacle can be detected by various sensors  $s_1, \dots, s_i, i \in \mathbb{N}^+$ , with partial obstacle certainty  $P_{O_D}(s_k, c_{i,j,k})$ , then cumulative probability of obstacle is given by:

$$P_{O_D}(c_{i,j,k}) = 1 - \left\{ \prod_{s_k \in s_1, \dots, s_i}^{i \in \mathbb{N}^+} (1 - P_{O_D}(s_k, c_{i,j,k})) \right\} \quad (5.38)$$

Final detected obstacle probability  $P_{O_D}(c_{i,j,k})$  is given as inverted cumulative probability  $P_{O_D}(s_k, c_{i,j,k})$  of partial probabilities for sensor array  $s_k \in s_1, \dots, s_i, i \in \mathbb{N}^+$ .

**Probability of obstacle occurrence in case of LiDAR sensor** , for this case  $s_k$  is fixed and its set as homogeneous two axis rotary LiDAR. For one cell  $c_{i,j,k}$  there exists set of passing LiDAR beams:

$$\mathcal{L}(c_{i,j,k}) = \left\{ [\theta \in \Theta, \varphi \in \Phi] \in \mathbb{R}^2 : \begin{array}{l} c_{i,j,k} \cdot \theta_s \leq \theta \leq c_{i,j,k} \cdot \theta_e \\ c_{i,j,k} \cdot \varphi_s \leq \varphi \leq c_{i,j,k} \cdot \varphi_e \end{array} \right\} \quad (5.39)$$

The pair  $[\theta, \varphi]$  is in homogeneous offset system given by product of discrete set of horizontal angles offsets  $\Theta$  and discrete set of vertical angles offsets  $\Phi$ . The set  $\mathcal{L}(c_{i,j,k})$  is finite countable and nonempty for any  $c_{i,j,k}$ , otherwise definition of avoidance grid  $\mathcal{A}(t_i)$  needs to be changed.

The hit function  $h : \mathcal{L}(c_{i,j,k}) \times \mathbb{R}^n \rightarrow [0, \infty]$  returns a distance of single beam return for beam passing through  $[\theta, \varphi] \in \mathcal{L}(c_{i,j,k})$  angle offsets with vehicle state space  $\vec{x} \in \mathbb{R}^n$ . Then set of LiDAR hits  $\mathcal{H}$  in cell  $c_{i,j,k}$  at system state  $\vec{x}$  is given as follow:

$$\mathcal{H}(\vec{x}, c_{i,j,k}) = \left\{ [d, \theta, \varphi] \in \mathbb{R}^3 : \begin{array}{l} c_{i,j,k} \cdot d_s \leq d \leq c_{i,j,k} \cdot d_e \\ \forall [\theta, \varphi] \in \mathcal{L}(c_{i,j,k}) \\ d = h(\mathcal{L}(c_{i,j,k}), [\theta, \varphi]) \end{array} \right\} \quad (5.40)$$

Probability of obstacle in case of LiDAR detection  $P_{O_L}$  is given as ratio between landed hits and possible hits:

$$P_{O_L}(\vec{x}, c_{i,j,k}) = \frac{|\mathcal{H}(\vec{x}, c_{i,j,k})|}{|\mathcal{L}(c_{i,j,k})|} \quad (5.41)$$

Probability of vision hindrance  $P_{V_H}$  is given as supplement to probability of obstacle:

$$P_{V_H}(\vec{x}, c_{i,j,k}) = 1 - \frac{|\mathcal{H}(\vec{x}, c_{i,j,k})|}{|\mathcal{L}(c_{i,j,k})|} \quad (5.42)$$

**Cell density function.** Let's start with differential form of cell surface calculation eq. 3.54. The target object have several hits in given cell Avoidance grid  $\mathcal{A}$  cell  $c_{i,j,k}$ . Cell  $c_i, j, k$  have following properties which are used in surface calculation:

1. *Horizontal span* ( $\theta_s, \theta_e$ ) - defines range of horizontal scanner partition.
2. *Vertical span* ( $\varphi_s, \varphi_e$ ) - defines range of vertical scanner partition.

By rewriting eq. 3.54. and using  $\theta$  as horizontal range parameter and  $\varphi$  as inverted vertical range parameter following surface integral is proposed:

$$\int_{\theta_s}^{\theta_e} \int_{\varphi_s}^{\varphi_e} dA \quad d\varphi d\theta = \int_{\theta_s}^{\theta_e} \int_{\varphi_s}^{\varphi_e} r^2 \cos(\varphi) \quad d\varphi d\theta \quad (5.43)$$

Numerical stable integration exist for boundaries  $\theta$  in  $[-\pi, \pi]$   $\varphi \in [0, \frac{pi}{2}]$  and is given by following equation:

$$A(r, \theta_s, \theta_e, \varphi_s, \varphi_e) = \begin{cases} \varphi_s < 0, \varphi_e \leq 0 : r^2(\sin |\varphi_s| - \sin |\varphi_e|)(\theta_e - \theta_s) \\ \varphi_s < 0, \varphi_e > 0 : r^2(\sin |\varphi_s| + \sin |\varphi_e|)(\theta_e - \theta_s) \\ \varphi_s \geq 0, \varphi_e < 0 : r^2(\sin \varphi_e - \sin \varphi_s)(\theta_e - \theta_s) \end{cases} \quad (5.44)$$

Intersection surface for cell  $A_C$  is then given by eq. 5.44. Most of the LiDAR scanners have *Points per rotation parameter [ppr]*.  $P_S[p/2\pi]$  Let's assume that scanning density is homogeneous. The covered area at distance  $r$  for one LiDAR swipe  $A_S$  is given by eq. 5.44. where  $\theta_s = \pi$ ,  $\theta_e = \pi$ , for full swipe and  $\varphi_s, \varphi_e$  set to maximal horizontal range. Then for object with frontal area surface  $A_O$ , the minimal hit count  $P_O$  is given as:

$$P_{O_D} = \min \left\{ P_S \frac{A_O}{A_S}, P_S \frac{A_C}{A_S} \right\} \quad (5.45)$$

*LiDAR hit formula* (5.45) covers hit count for cell  $c_{i,j,k}$  and object  $O$  of any size  $A$  and any detection distance  $r$ .

### 5.3 Visibility probability

For each cell  $c_{i,j,k}$  and each sensor  $s_k$  there exist hindrance of vision probability  $P_{V_H}$ , which defines how much vision is clouded in single cell. Example of hindrance calculation for LiDAR has been given by eq. 5.42. Let us consider cell row  $\mathcal{C}(j_{fix}, k_{fix})$  with fixed horizontal index  $j_{fix}$  and vertical index  $k_{fix}$  is given as series of:

$$\mathcal{C}(j_{fix}, k_{fix}) = \{c_{i,j,k} \in \mathcal{A}(t_i) : i \in \{1, \dots, a\}, j = j_{fix}, k = k_{fix}\} \quad (5.46)$$

For each cell  $c_{i,j,k}$  there exists a function which calculates final visibility hindrance probability  $P_{V_F}(c_{i,j,k}, \mathcal{S})$ , where  $\mathcal{S}$  is set of sensors providing hindrance rate. Then for ordered cell row  $\mathcal{C}(j_{fix}, k_{fix}) = \{c_{1,j_{fix},k_{fix}}, c_{2,j_{fix},k_{fix}}, \dots, c_{a,j_{fix},k_{fix}}\}$ ,  $a \in \mathbb{N}^+$  and for one selected

cell  $c_{i,j,k}$  the probability of visibility is given as supplement to hindrance from previous cells. The equation for this statement holds as follows:

$$P_V(c_{i_c,j_c,k_c}, \mathcal{S}) = 1 - \sum_{a \in \mathbb{N}^+}^{a < i_c} P_{V_F}(c_{a,j_c,k_c}, \mathcal{S}), \quad c_{a,j_c,k_c} \in \mathcal{C}(j_c, k_c) \quad (5.47)$$

For example cell  $c_{4,j_{fix},k_{fix}}$  is selected for visibility calculation  $P_V(c_{4,j_{fix},k_{fix}})$ , then cells  $c_{1,j_{fix},k_{fix}}, c_{2,j_{fix},k_{fix}},$  and  $c_{3,j_{fix},k_{fix}}$ , are used as a base of calculation for visibility hindrance probability  $P_{V_F}$ .

The maximum hindrance for any cell row  $|C(j_{fix}, k_{fix})| = k \in \mathbb{N}^+$  with boundary:

$$0 \leq \sum_{c \in C(j_{fix}, k_{fix})} P_{V_F}(c) \leq 1 \quad (5.48)$$

For one cell row  $\mathcal{C}(j_{fix}, k_{fix})$ , where count of layers is equal to 10, and layers have equal spacing. There is defined sensor field  $\mathcal{S} = \{s_1\}$ , where  $s_1$  is LiDAR sensor.

During consequent LiDAR scans  $s(t_0), s(t_1), s(t_2)$ , and  $s(t_3)$  the obstacle sets  $\mathcal{O}_1(t_1) = \{o_1\}$ ,  $\mathcal{O}_2(t_2) = \{o_1, o_2\}$ , and  $\mathcal{O}_3(t_3) = \{o_1, o_2, o_3\}$  are discovered. Assigned hindrance probabilities are like follow:

- Time  $t_0$  (fig. 5.5) - there is no obstacle nor hindrance, all cells are free
- Time  $t_1$  (fig. 5.6) -  $\mathcal{O}_1(t_1) = \{o_1\}$  was detected, the hindrance probability  $P_{V_F}(c_{3,j_{fix},k_{fix}})$  is equal to 0.25. The visibility in cells  $c_{4-10,j_{fix},k_{fix}}$  is 75 percent now.
- Time  $t_2$  (fig. 5.7) -  $\mathcal{O}_2(t_2) = \{o_1, o_2\}$  was detected, the additional hindrance probability  $P_{V_F}(c_{5,j_{fix},k_{fix}})$  is 0.15. The visibility in cells  $c_{6-10,j_{fix},k_{fix}}$  is lowered by additional 15 percent and its set to 60 percent now.
- Time  $t_3$  (fig. 5.8) -  $\mathcal{O}_3(t_3) = \{o_1, o_2, o_3\}$  was detected the additional hindrance probability  $P_{V_F}(c_{7,j_{fix},k_{fix}})$  is 0.20. The visibility in cells  $c_{8-10,j_{fix},k_{fix}}$  is lowered by additional 20 percent and its set to 40 percent now.

The vehicle is detecting in avoidance grid  $\mathcal{A}(t_0)$  and do not detected any obstacles (fig. 5.5). The cell row  $\mathcal{C}(j_{fix}, k_{fix})$  is completely visible.

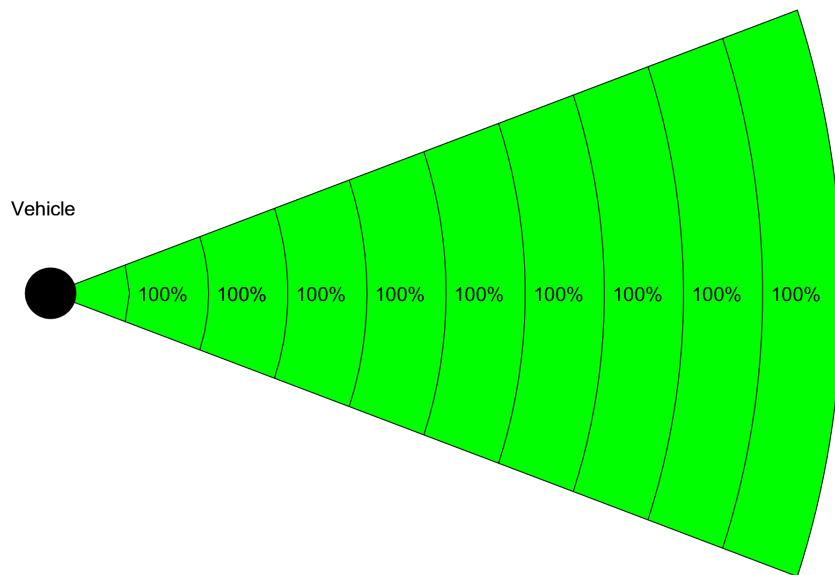


Figure 5.5: Visibility probability for cell row - free space

The vehicle is detecting in avoidance grid  $\mathcal{A}(t_1)$  and detects one obstacle  $O_1$  (fig. 5.6). The obstacle is placed in third cell from vehicle. Note that visibility of third cell is 100 percent, because no prior hindrance in line of sight exist.

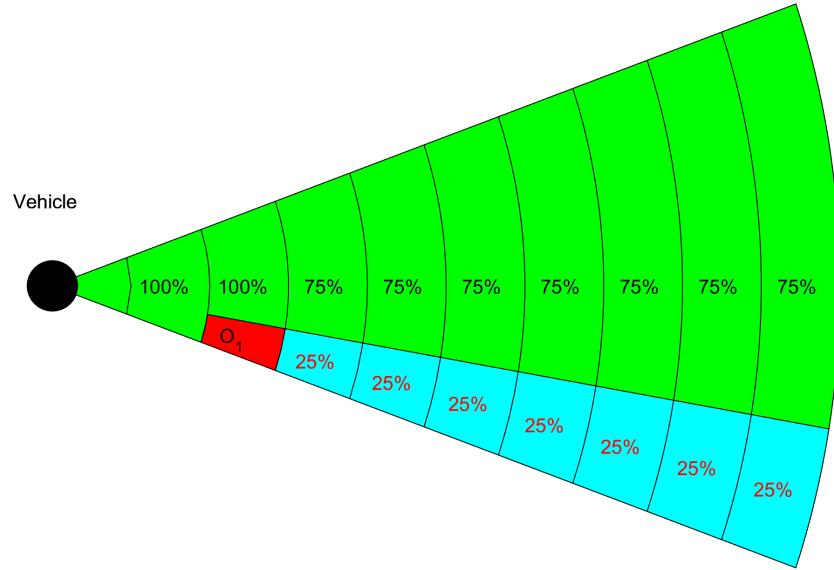


Figure 5.6: Visibility probability for cell row - first obstacle

The vehicle is detecting in avoidance grid  $\mathcal{A}(t_2)$  detects obstacles  $O_1, O_2$  (fig. 5.7). The cell row  $\mathcal{C}(j_{fix}, k_{fix})$  is now hindered with obstacles  $O_1$  with hindrance 0.25 and obstacle  $O_2$  with hindrance 0.15.

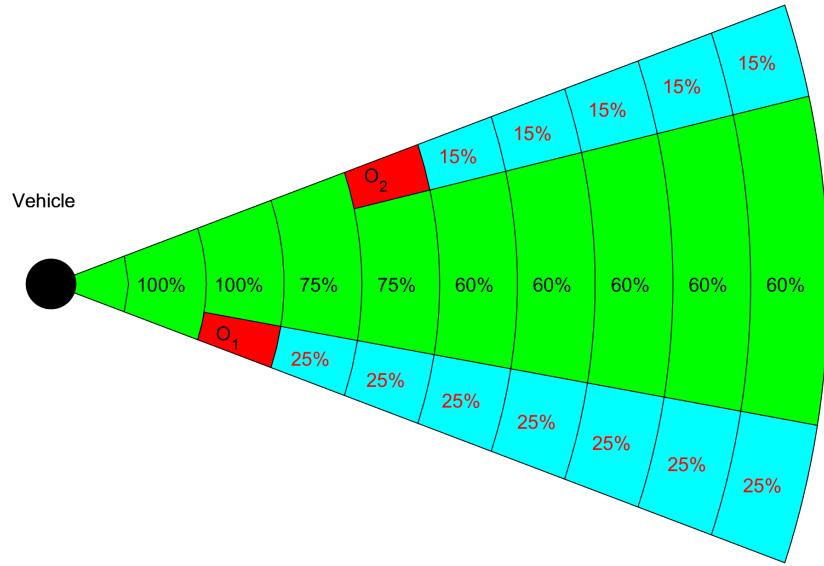


Figure 5.7: Visibility probability for cell row - second obstacle

The vehicle is detecting in avoidance grid  $\mathcal{A}(t_3)$  detects obstacles  $O_1, O_2, O_3$  (fig. 5.8). The cell row  $\mathcal{C}(j_{fix}, k_{fix})$  is now hindered with obstacles  $O_1$  with hindrance 0.25, obstacle  $O_2$  with hindrance 0.15, and obstacle  $O_3$  with hindrance 0.20.

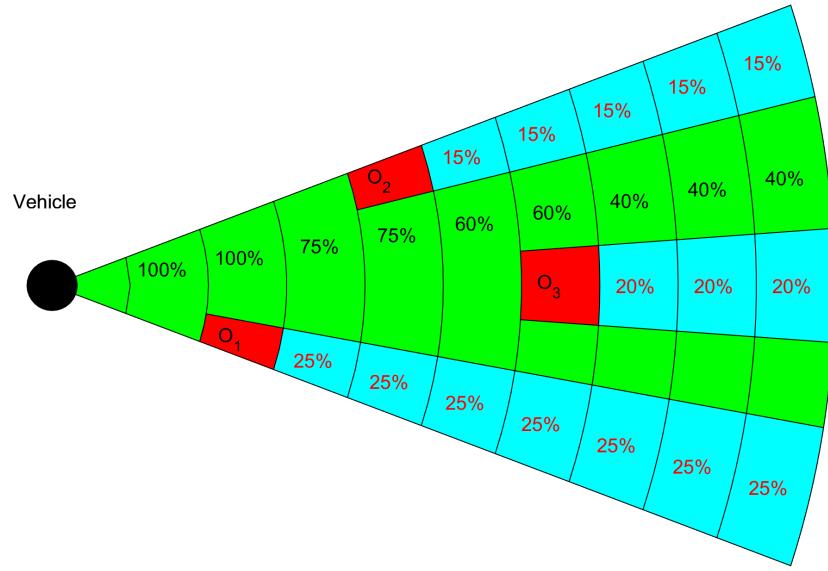


Figure 5.8: Visibility probability for cell row - third obstacle

## 5.4 Map obstacle probability

*Probability of Map Obstacle  $P_{O_M}$*  occurrence follows very simple logic. There are three possible cases:

*Map obstacle  $O_M$*  is charted on map (fig. 5.9), but is undetected by any sensor in sensor set  $\mathcal{S}$ . Therefore the probability of map obstacle occurrence is equal to 0.

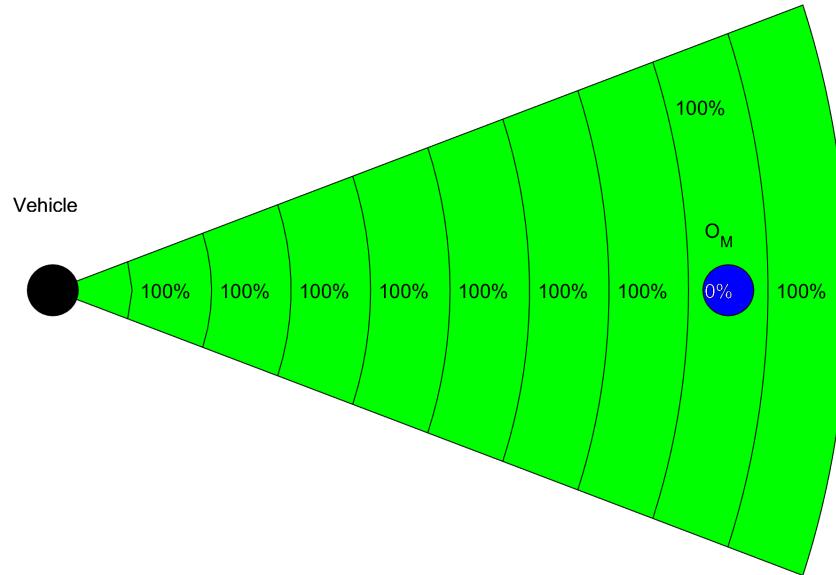


Figure 5.9: Map obstacle - undetected by sensory system

*Map obstacle  $O_M$*  is charted on map and detected by any sensor in sensor system  $\mathcal{S}$  (fig. 5.10). The probability of map obstacle  $P_{O_M}$  is equal to probability of detected obstacle  $P_{O_D}$ , therefore its equal to 1.

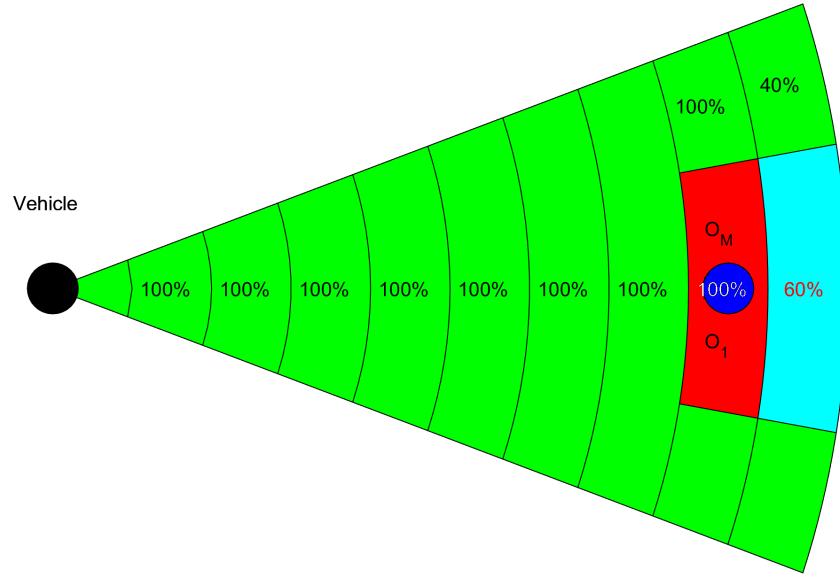


Figure 5.10: Map obstacle - detected by sensory system

*Map obstacle  $O_M$*  is hindered behind other detected obstacle  $O_1$  (fig. 5.11). The detected obstacle  $O_1 \in \mathcal{O}_D$  is in cell  $c_{i,j,k}$  and is reducing visibility in follow up cells  $c_{i_f > i,j,k}$  by 60 percent.

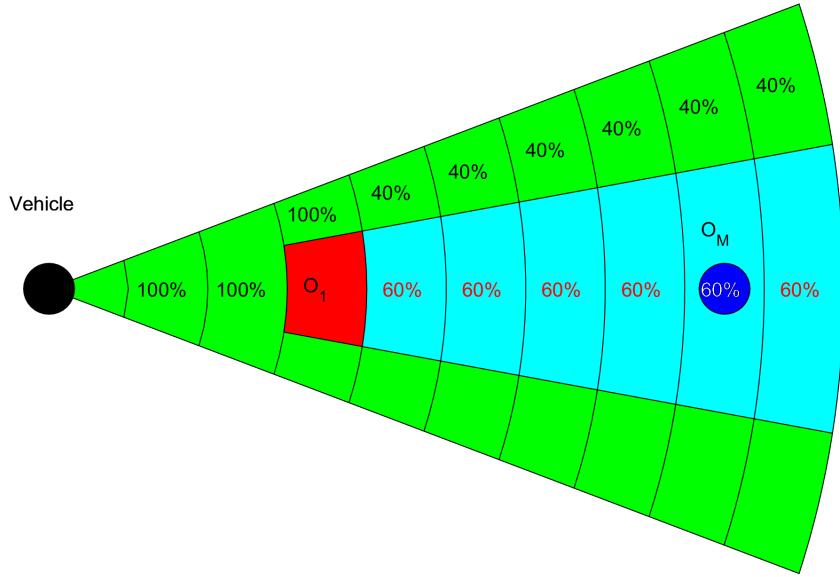


Figure 5.11: Map obstacle - hidden behind detected obstacle

The formulation of final map obstacle probability  $P_{O_M}(c_{i,j,k})$  is given by previous examples which are giving the *desired behaviour*. First we start with obstacle map definition.

The obstacle map  $\mathcal{O}_m$  (5.49) defines an map obstacle set  $o = [\vec{p}, \vec{o}, r, \vec{a}]$  where  $\vec{p}$  is position in global coordinate frame (center of mass),  $\vec{o}$  is orientation bounded to global coordinate reference frame,  $r$  is offset radius and  $a$  is set pf additional parameters.

$$\mathcal{O}_M = \{[\vec{p}, \vec{o}, r, \vec{a}] : \vec{p} \in \mathbb{R}^3, \vec{o} \in \mathbb{R}^3, r \in (0, \infty), \vec{a} \in \mathbb{R}^k, k \in \mathbb{N}\} \quad (5.49)$$

The space covered by any obstacle  $o \rightarrow \mathbb{R}^3$  is non-empty by definition. There are following types of charted obstacle considered in this example.

1. *Ball obstacle*  $\vec{a} = \emptyset$  - simple ball with center at  $\vec{p}$ , with offset radius  $r$ .
2. *Line obstacle*  $\vec{a} = [l]$  - simple line defined by length  $l \in (0, \infty)$  with center at  $\vec{p}$  and given orientation  $\vec{o}$  with respect to main axis in global coordinate frame, with offset radius  $r$ .
3. *Plane obstacle*  $\vec{a} = [l, w]$  - bounded rectangle plane partition defined by length  $l \in (0, \infty)$ , and width  $w \in (0, \infty)$  with center at  $\vec{p}$  and given orientation  $\vec{o}$  with respect to main axis in global coordinate frame, with offset radius  $r$ .
4. *Cuboid obstacle*  $\vec{a} = [l, w, d]$  - bounded cuboid space partition defined by length  $l \in (0, \infty)$ , width  $w \in (0, \infty)$ , and depth  $d \in (0, \infty)$  with center at  $\vec{p}$  and given orientation  $\vec{o}$  with respect to main axis in global coordinate frame, with offset radius  $r$ .

The obstacle set for some grid  $\mathcal{A}$  at decision time  $t_i$  intersection with obstacle map  $\mathcal{O}_m$  is given by  $\mathcal{O}_M(\mathcal{A}(t_i))$  (5.50). Where map obstacle space  $o$  is projected into avoidance grid space  $\mathcal{A}(t_i)$ . Projection output  $\vec{p}_l \subset \mathbb{R}^3$  is partial subset of local coordinate frame space given by  $A(t_i)$ . If there exist non empty intersection between avoidance grid  $\mathcal{A}(t_i)$  and local projection of obstacle space  $\vec{p}_l$  then the given obstacle belongs to  $\mathcal{O}_M(\mathcal{A}(t_i))$ .

$$\mathcal{O}_M(\mathcal{A}(t_i)) = \{o \in \mathcal{O}_M : \vec{p}_l : o \rightarrow \mathcal{A}(t_i), \vec{p}_l \cap \mathcal{A}(t_i) \neq \emptyset\} \quad (5.50)$$

$\mathcal{O}_M(c_{i,j,k}, \mathcal{A}(t_i))$  (5.51) is analogy to  $\mathcal{O}_M(\mathcal{A}(t_i))$  (5.50). The cell space  $c_{i,j,k} \subset \mathcal{A}(t_i)$  is considered as criterion instead of avoidance grid  $\mathcal{A}(t_i)$ .  $\mathcal{O}_M(c_{i,j,k}, \mathcal{A}(t_i))$  represents set of map obstacles in cell  $c_{i,j,k}$  in context of avoidance grid  $\mathcal{A}(t_i)$ .

$$\mathcal{O}_M(c_{i,j,k}, \mathcal{A}(t_i)) = \{o \in \mathcal{O}_M : \vec{p}_l : o \rightarrow \mathcal{A}(t_i), \vec{p}_l \cap c_{i,j,k} \neq \emptyset\} \quad (5.51)$$

*Base probability of map obstacle*  $P_{O_{M,B}}$  for single map obstacle  $o \in \mathcal{O}_M$  (5.52) is given as ration between intersection volume  $\iiint(o \rightarrow \mathcal{A}(t_i)) \cap c_{i,j,k} \text{ dxdydz}$  and cell volume  $\iiint c_{i,j,k} \text{ dxdydz}$

$$P_{O_{M,B}}(o, c_{i,j,k}) = \frac{\iiint(o \rightarrow \mathcal{A}(t_i)) \cap c_{i,j,k} \text{ dxdydz}}{\iiint c_{i,j,k} \text{ dxdydz}} \quad (5.52)$$

*Base probability of map obstacle*  $P_{O_{M,B}}$  for map obstacle set  $\mathcal{O}_M$  (5.53) is calculated in similar manner as base probability of map obstacle  $P_{O_{M,B}}$  for single map obstacle  $o \in \mathcal{O}_M$  (5.52). The obstacle member of denominator has changed to space union of all obstacle space in local coordinate frame of avoidance grid  $\mathcal{A}(t_i)$  as follow  $\bigcup_{o \in \mathcal{O}_M} o \rightarrow \mathcal{A}(t_i)$ .

$$P_{O_{M,B}}(\mathcal{O}_M, c_{i,j,k}) = \frac{\iiint (\bigcup_{o \in \mathcal{O}_M} o \rightarrow \mathcal{A}(t_i)) \cap c_{i,j,k} \text{ dxdydz}}{\iiint c_{i,j,k} \text{ dxdydz}} \quad (5.53)$$

*Final map obstacle probability*  $P_{O_M}(c_{i,j,k})$  (5.54) is given as product of *base probability of map obstacle*  $P_{O_{M,B}}$  for map obstacle set  $\mathcal{O}_M$  (5.53), and inverted *final visibility probability*  $P_V(c_{i,j,k})$  (5.47).

$$P_{O_M}(c_{i,j,k}) = P_{O_{M,B}}(\mathcal{O}_M, c_{i,j,k}) \quad (5.53) \times (1 - P_V(c_{i,j,k}, \mathcal{S})) \quad (5.47) \quad (5.54)$$

# Chapter 6

## Simulation Results

### 6.1 Mission control

This section briefly describes responsibilities and tasks of mission control module, implementation of mission control module is similar to *LSTS-Neptus* [? ? ].

**Navigation** main purpose is to execute mission between ordered waypoint set  $\mathcal{WP}$  (6.1).

$$\mathcal{WP} = \{W_1, W_2, \dots, W_n : W_i \in \mathbb{R}^3, i \in \{1, \dots, n\}, n \geq 2\} \quad (6.1)$$

Where each waypoint lies in 3D space  $\mathbb{R}^3$  and there is at least two waypoints. There is enforcement of implementation condition that distance of any two consecutive waypoints  $\forall i, j = i + 1, \|W_i - W_j\| \geq r_r$  is greater than reach radius  $r_r$ . *Goal waypoint*  $W_G$  is selected in following manner:

1. *First waypoint*  $W_1 \in \mathcal{WP}$  is selected as goal waypoint  $W_G$ .
2. *Until last waypoint*  $W_n \in \mathcal{WP}$  is reached under reach condition  $\|\vec{x}(t) \rightarrow \mathbb{R}^3 - W_G\| \leq r_r$ , check reach condition and select next waypoint in ordered set
3. *When last waypoint*  $W_n \in \mathcal{WP}$  is reached execute landing sequence.

**Movement execution** of the movement automaton  $\mathcal{MA}$  (def. 12). The main functionality of mission control is to link various movement buffers from avoidance grids  $\mathcal{A}(t_i)$  into one single movement chain. The overall architecture has been presented in my earlier works [? ].

The key concept is as follows. For dynamically generated set of decision times  $t_1, t_2, \dots, t_i$  there exists corresponding a set of avoidance grids  $\mathcal{A}(t_1), \mathcal{A}(t_2), \dots, \mathcal{A}(t_i)$ , where reach sets are given as set of trajectories. Each avoidance grid has corresponding avoidance trajectory which has been selected  $\mathcal{T}(\vec{x}, B)$ . During a flight there exist a corresponding buffer for each decision time  $t_1 \rightarrow B_1, t_2 \rightarrow B_2, \dots, t_i \rightarrow B_i$ .

Single movement buffer  $B_i$  from trajectory  $\mathcal{T}(\vec{x}(t_i), B_i)$  is given by following equation:

$$B_i = \{m_{i,1}, m_{i,2}, \dots, m_{i,k}, m_{i,k+1}, \dots, m_{i,j-1}, m_{i,j}\}, \quad 1 \leq k \leq i, i \in \mathbb{N}^+ \quad (6.2)$$

Where  $m_{i,1}$  is first *to be executed/executed* and  $m_{i,k}$  is last *to be executed/executed* movement,  $m_{i,k+1}, \dots, m_{i,j}$  is flushed part (not used part of buffer),  $k$  is count of *to be executed/executed movements*, and  $i$  is length of movement buffer  $B_i$ . This can be extended

to any buffer belonging to any decision time. Then by assumptions given in [? ], there exists positive integer  $k_i$  for any avoidance grid  $\mathcal{A}(t_i)$  such movements can be chained.

*Final movement buffer*  $B$  (6.3) is ordered chain of *to be executed/executed* movements  $\{m_{a,1}, m_{a,2}, \dots, m_{a,k}\}$  from partial trajectories at times of decision  $t_1, \dots, t_i$  with belonging buffers  $\{B_1, \dots, B_i\}$ .

$$B = \bigcup_{B_a=\{B_1, \dots, B_i\}} \{m_{a,1}, m_{a,2}, \dots, m_{a,k}\} \quad (6.3)$$

The example of *final movement buffer*  $B$  is given in fig. 6.1. The blue line represents *to be executed/executed* movements, red line represents *flushed movements*, purple circles represents *decision points* at time  $t_1, \dots, t_5$ . Last portion at  $t_5$  is denominated red, because it was completely throw away due the *waypoint reach condition* was met.

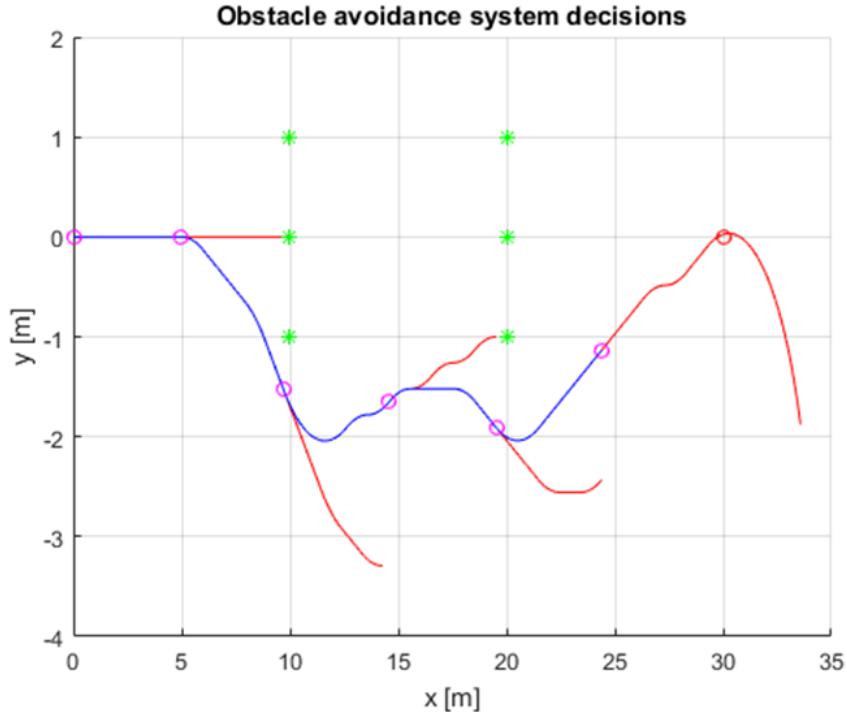


Figure 6.1: Example of partial movement buffers chaining.

**Obstacle processing and Control switch** has been reused from [? ].

## 6.2 Reduced reach set approximation computation

The worst case scenario for LiDAR scanning have been given in [? ]. That means the minimal conical range of LiDAR in middle class is  $\tilde{10} \text{ m}$ . The main idea is to test spread reduction vs. coverage ratio  $C_R$  on grid with many layers. The deterministic approach of reach set estimation in [? ] was sustainable to calculate full reach set for 5 movements and 10 layers, that implies count of nodes ( $n = \sum_{i=1}^1 05^i$ ). The proposed grid will have 15 layers and 9 movements (estimation of such reach set in full form is impossible). *The pruning of unfeasible pathways* is executed in iterative manner, which enables to minimize memory consumption and effort during *reach set estimation calculation*. The scalability of presented approaches is one of the main features and even the calculation time is scaling well. The approximation algorithm implementation has *borderline polynomial* complexity.

Three different methods have been developed and tested on following avoidance grid  $\mathcal{A}(t_i)$ :

1. Range = 15m, 15 layers
2. Horizontal range:  $\theta \in [-\frac{\pi}{4}; \frac{\pi}{4}]$ , 7 horizontal cells
3. Vertical range:  $\varphi \in [-\frac{\pi}{6}; \frac{\pi}{6}]$ , 5 vertical cells
4. 525 cells, 35 for each layer

*Complexity* of unbounded tree search with pruning procedure after computation is given as:

$$\mathcal{O}(n) \leq m \times r \times h \times v \times l + e^{m \times r} \quad (6.4)$$

Where  $m$  - is count of movements,  $r$  is spread rating,  $h$  is horizontal cell count of  $\mathcal{A}(t_i)$ ,  $v$  is vertical cell count of  $\mathcal{A}(t_i)$ ,  $l$  is layer count of  $\mathcal{A}(t_i)$ . The problematic term is  $e^{m \times r}$  which is exponential unexplored nodes residuum. The complexity  $\mathcal{O}(n)$  is non-polynomial now.

To solve this issue the post-expansion pruning algorithm needs to be introduced, this allows to transform  $e^{m \times r}$  to a linear/polynomial member  $m \times h \times v \times l$ . Therefore complexity  $\mathcal{O}(n)$  is now bounded by:

$$\mathcal{O}(n) \leq m \times (r + 1) \times h \times v \times l \quad (6.5)$$

The spread ratio  $r$  and movement count  $h$  is given, the scalable parameters are grid member counts  $h, v, l$ , it is obvious that  $h \times v \times l$  is cell count  $|\mathcal{A}(t_i)|$ . For our grid following formula for boundary can be used:

$$\mathcal{O}(n) \leq 9 \times k \times 525 \quad (6.6)$$

The performance parameters for various algorithms are summarized in following table:

Method/Parameter	Nodes	Trajectories	Coverage ratio $C_R$	Note
Chaotic method	2483	725	0.90	Forced spread: 9
Harmonic method	1405	180	0.30	
Combined method	2162	305	0.87	CH: 8, H: 1

Table 6.1: Reduced reach set computation methods performance

Reduced reach set approximation  $\mathcal{R}_R$  using combined method can cover almost same cell footprints with very small node count:

$$\frac{\text{node count } \mathcal{R}_R}{\text{node count } \mathcal{R}_F} = \frac{2483}{\sum_{k=1}^{15} 9^k} \sim 1.2088 \times 10^{-12} \quad (6.7)$$

Depending on selected spread algorithm and pruning algorithm, the reduced reach set keep properties:

1. *Smoothness* - trajectories contained in reduced reach set approximation are smooth and with low cost.
2. *Coverage*  $C_R$  - trajectories are covering high amount of footprints from full reach set approximation footprint set  $\mathcal{Q}_F$ .

## 6.3 Moving obstacles spread models

This section discuss implementation and performance of various intruder intersection models. The conditions of intersection was stated as continuous space problem, for real implementation numeric approximation methods keeping probabilistic properties of continuous space were used. Following basic models have been identified in section 5.1.:

1. *Direct (Line) Intersection* - for static/timed environment.
2. *Future (Line) Movement* - for timed environment.
3. *Vehicle body volume* - for timed environment.
4. *Uncertainty spread* - for static/timed environment.

Given is intruder  $I$  with following properties:

1. *Position in Local Coordinate Frame* -  $[6, 8, 0]^T$
2. *Velocity* - scalar  $1ms^{-1}$ , vector  $[0, -1, 0]^T$   $[ms^{-1}, ms^{-1}, ms^{-1}]$
3. *Horizontal spread  $\theta$*  -  $[-\pi/12, \pi/12]$   $[rad, rad]$
4. *Vehicle body radius* -  $2m$
5. *Vertical spread  $\varphi$*  -  $[-\pi/16, \pi/16]$   $[rad, rad]$

All intersection methods have been tested on following avoidance grid  $\mathcal{A}(t_i)$ :

1. Range = 10 m, 10 layers
2. Horizontal range:  $\theta \in [-\frac{\pi}{4}; \frac{\pi}{4}]$ , 7 horizontal cells
3. Vertical range:  $\varphi \in [-\frac{\pi}{6}; \frac{\pi}{6}]$ , 5 vertical cells
4. 350 cells, 35 for each layer

### 6.3.1 Line intersection - Static

For line intersection standard *line back-stepping* numeric algorithm was used. Base intersection probability for line  $P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k})$  (5.9) was used. The implementation is trivial and intersection probability  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.36) The flags are set  $l = 1$ ,  $b = 0$ ,  $s = 0$ , and  $\tau = 0$ . That means only line intersection is accounted.

Figure 6.2: Obstacle space for static line intersection

The figure 6.2 is showing trajectory space and middle horizontal layer obstacle space  $\mathcal{O}(t)$ . The blue line is representing intruder intersection trajectory starting at top and finishing at bottom of avoidance grid  $\mathcal{A}(t_i)$ . The red portion of trajectories on left sub-figure is reflecting the impacted portions of UAV trajectories. The red cells on right sub-figure are reflecting impacted cells with probability of intruder  $P_I = 1$ .

Figure 6.3: Reach set approximation for static line intersection - middle layer

The figure 6.3 is showing reachable set approximation trajectories  $\mathcal{R}(\hat{x}(t_i), t_i, t_{i+1})$  and middle layer of avoidance grid  $\mathcal{A}(t_i)$  with assessed reachability. The blue line is denominating the intruder trajectory (side approach), it starts at top and ends at bottom. The left sub-figure shows set of trajectories where green parts are executable and red parts are crash trajectories. The right sub-figure shows the *reachable space*, the red cells are cells which can not be reached and the green cells are cells which can be reached. Notice that the green cells in front are reachable, because there exists at least one trajectory leading there through unoccupied space.

### 6.3.2 Line intersection - Timed

For line intersection standard *line back-stepping* numeric algorithm was used. Base intersection probability for line  $P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k})$  (5.9) was used. The implementation is trivial and intersection probability  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.36) The flags are set  $l = 1$ ,  $b = 0$ ,  $s = 0$ , and  $\tau = 1$ . That means The base intersection probability and time interval intersection probability components are accounted. The expected danger zones are valid if and only if the intruder and vehicle velocity remains constant. If this condition can not be full-filled the  $\tau = 0$  in probabilistic model. The obstacle probability is denoted in white/red scale, the reachability probability is denoted in green/red scale.

Figure 6.4: Obstacle space for timed line intersection

The figure 6.4 is showing trajectory space and middle horizontal layer obstacle space  $\mathcal{O}(t)$ . The blue line is representing intruder intersection trajectory starting at top and finishing at bottom of avoidance grid  $\mathcal{A}(t_i)$ . The red portion of trajectories on left sub-figure is reflecting the impacted portions of UAV trajectories. The red cells on right sub-figure are reflecting impacted cells with probability of intruder  $P_I \in [0, 1]$ . Notice that probability of intersection is not always equal to 1, because of time component it can be slightly decreased (pink cells).

Figure 6.5: Reach set approximation for timed line intersection - middle layer

The figure 6.5 shows reachability property intersection with intruder linear trajectory (light blue). The left sub-figure shows the approximation of reach set  $\mathcal{R}(\hat{x}(t_i), t_i, t_{i+1})$ . The set of trajectories is colored in green-red color, meaning that reachability of the trajectory is absolute when green or not reachable at all if red. The left sub-figure shows reachability status of cells in middle layer. Some of cells are unreachable with reachability probability  $P_R(c_{i,j,k}) \in [0, 1]$ . Please note that last layer is fully reachable and intruder can be avoided via the other layers.

### 6.3.3 Line intersection - Future movements

For line intersection standard *line back-stepping* numeric algorithm was used. Base intersection probability for line  $P_T(i_k(\vec{x}, \vec{v}), c_{i,j,k})$  (5.9) was used. The implementation is trivial and intersection probability  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.36) The flags are set  $l = 1$ ,  $b = 0$ ,  $s = 0$ , and  $\tau = 1$ . That means The base intersection probability and time interval intersection probability components are accounted. The expected danger zones are valid

if and only if the intruder and vehicle velocity remains constant. If this condition can not be full-filled the  $\tau = 0$  in probabilistic model. The obstacle probability is denoted in white/red scale, the reachability probability is denoted in green/red scale. There has been implemented a special case for future movement and it improves time intersection probability:

$$P_{O_I}(i_k, c_{i,j,k}) = \begin{cases} t_e \geq i_l : & P_{O_I}(i_k, c_{i,j,k}, l = 1, b = 0, s = 0, \tau = 1) \\ t_e < i_l : & P_{O_I}(i_k, c_{i,j,k}, l = 1, b = 0, s = 0, \tau = 0)/2 \end{cases} \quad (6.8)$$

Where  $t_e$  is vehicle entry time to cell  $c_{i,j,k}$ ,  $i_l$  is intruder leave time of cell  $c_{i,j,k}$  other notation strictly follows (5.36). The main idea is to force avoidance from behind of intruder heading, which has been achieved, but is not required to encode higher level decision logic into probabilistic distributions.

Figure 6.6: Obstacle space for timed line intersection with future movement

The figure 6.6 is showing trajectory space and middle horizontal layer obstacle space  $\mathcal{O}(t)$ . The blue line is representing intruder intersection trajectory starting at top and finishing at bottom of avoidance grid  $\mathcal{A}(t_i)$ . The red portion of trajectories on left sub-figure is reflecting the impacted portions of UAV trajectories. The red cells on right sub-figure are reflecting impacted cells with probability of intruder  $P_I \in [0, 1]$ . Notice that probability of intersection is not always equal to 1, because of time component it can be slightly decreased (pink cells). The future movement cells are located as new addition at bottom of trajectory, for reference compare fig. 6.4.

Figure 6.7: Reach set approximation for timed line intersection with future movement - middle layer

The figure 6.7 shows reachability property intersection with intruder linear trajectory (light blue). The left sub-figure shows the approximation of reach set  $\mathcal{R}(\hat{x}(t_i), t_i, t_{i+1})$ . The set of trajectories is colored in green-red color, meaning that reachability of the trajectory is absolute when green or not reachable at all if red. The left sub-figure shows reachability status of cells in middle layer. Some of cells are unreachable with reachability probability  $P_R(c_{i,j,k}) \in [0, 1]$ . Please note how reach set have been changed due the future movements, it is almost certain that vehicle will avoid intruder from behind (fig. 6.5).

### 6.3.4 Vehicle body volume - timed

*Base intersection probability*  $P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k})$  (5.13) is defined for volume ball  $\mathcal{B}(\vec{x}(t), r)$  (5.11). The volume ball  $\mathcal{B}(\vec{x}(t), r)$  is given only as a boundary can be represented as a convex set  $S_1$ . The grid cell  $c_{i,j,k}$  can be represented as polyhedron. The numeric approximation of these two structures intersection was proposed in [?], *strong conical hull intersection property* guarantees to avoid false positive intersection cases. The algorithm in [?] also gives the volume of the output, which can be used to enhance *base intersection probability* like follow:

$$P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k}, t) = \frac{V(c_{i,j,k} \cap \mathcal{B}(\vec{x}(t), r))}{V(c_{i,j,k})} \in [0, 1] \quad (6.9)$$

Where  $V(c_{i,j,k} \cap \mathcal{B}(\vec{x}(t), r))$  is volume of cell  $c_{i,j,k}$  belonging to some avoidance grid  $\mathcal{A}(t_i)$  at some avoidance time  $t_i$ , intersected with a volume of ball  $\mathcal{B}$  with center given by  $\vec{x}(t)$  (5.1) and radius  $r$ . The denominator is simple volume of cell  $V(c_{i,j,k})$ . The volume of convex polytopes can be calculated with linear complexity  $\mathcal{O}(n)$ , where  $n$  is count of vertexes as proposed in [? ].

This is great for fixed time  $t$ , but the time is not fixed in our case. For fixed time interval  $[t_e, t_l]$  where  $t_e$  entry time is given as first time when  $P_T(i_k(\vec{x}, \vec{v}, r, t), c_{i,j,k}) \neq 0$ , and  $t_l$  leave time is given as the last time when  $P_T(i_k(\vec{x}, \vec{v}, r, t), c_{i,j,k}) \neq 0$ . Then base probability  $P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k})$  for varying time  $t \in [t_e, t_l]$  can be obtained as a maximum of all probable intersections  $P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k}, t)$ , summarized in following equation:

$$P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k}) = \max_{t \in [t_e, t_l]} P_T(i_k(\vec{x}, \vec{v}, r), c_{i,j,k}, t) \quad (6.10)$$

*Tubular intersection* is possible low-cost replacement for embedded, systems, this type of intersection is similar to *conic intersection* (5.32), but the vehicle body radius parameter  $r_b$  is independent on time. The tubular slice is given as follow:

$$\mathcal{T}(\vec{x}(t), \vec{v}, r_s, r_b) = \left\{ \vec{p} \in \mathbb{R}^3 : \forall_{i \neq j} p_i \exists p_j, \|p_i - p_j\| \leq r_s, (\vec{p} - \vec{x}) \perp \vec{v}, \|\vec{p} - \vec{x}\| \leq r_b \right\} \quad (6.11)$$

*Tubular slice* fox fixed time  $t$  and given vehicle position  $\vec{x}(t) \in \mathbb{R}^3$  and orientation  $\vec{v} \in \mathbb{R}^3$  with defined step radius  $r_s$  and defined body radius  $r_r$  is given as set of points where each point holds following up to following conditions:

1.  $\forall_{i \neq j} p_i \exists p_j, \|p_i - p_j\| \leq r_s$  - for each point  $p_i$  there exists a point  $p_j (i \neq j)$  within the step radius  $r_s$ . This guarantees coverage and homogeneity, the  $r_s$  is chosen as 1/10 of average cell  $c_{i,j,i,k} \in \mathcal{A}(t_i)$  radius.
2.  $(\vec{p} - \vec{x}) \perp \vec{v}$  - all points  $\vec{p}$  are orthogonal to velocity vector  $\vec{v}$ .
3.  $\|\vec{p} - \vec{x}\| \leq r_b$  - all points are within bounded area of circle given by body radius  $r_b$ .

The *tube approximation* is given by following equation:

$$\mathcal{T}(\vec{x}, \vec{v}, r_s, r_b) = \left\{ \bigcup_{\vec{x}(t)=f(\vec{x}, \vec{v})}^{t \in [t_s, t_e]} \mathcal{T}(\vec{x}(t), \vec{v}, r_s, r_b) \right\} \quad (6.12)$$

The *tube approximation* is simply union of all tubular slices over defined time period  $[t_e, t_l]$  where  $t_e$  is entry time and  $t_l$  is leave time, mentioned earlier. The interval can be discredited by using steepest function or simple marginalization rule  $\|\vec{x}(t_i) - \vec{x}(t_{i+1})\| \leq r_r$ .

If *tube approximation* is used the cell entry  $t_e(c_{i,j,k})$  and leave time  $t_l(c_{i,j,k})$  are given based on *tubular slices* intersection  $\mathcal{T}(\vec{x}(t), \vec{v}, r_s, r_b)$  time  $t$ .

Figure 6.8: Obstacle space for timed intruder body volume

*Obstacle space for intruder body volume* with time dependency is depicted in fig. 6.8 with following meaning:

1. *Obstacle probability* - shows how three consequent timed balls  $\mathcal{B}(\vec{x}(t), r = 1)$  are projected intro reach set approximation  $\mathcal{R}(\hat{x}(t_i), t_i, t_{i+1})$ , red denotes unfeasible part of trajectories crossing dangerous area where intruder body can be encountered

2. *Layer 1 (top)* - shows that there is any expectation of intruder body.
3. *Layer 2* - contains top part of ball intersections, shades of red are denoting intersection probability [0, 1], [white, red].
4. *Layer 3 (middle)* - contains middle part of ball intersections, shades of red are denoting intersection probability [0, 1], [white, red]. The blue line denotes intruder linear trajectory.
5. *Layer 4* - contains bottom part of ball intersections, shades of red are denoting intersection probability [0, 1], [white, red].
6. *Layer 5 (bottom)* - shows that there is any expectation of intruder body.

Figure 6.9: Reach set approximation for timed vehicle body volume - all layers

*Reach set approximation for timed vehicle body volume* is given by fig. 6.9, the explanation of figure is following:

1. *Reachability probability* (red-green scale) shows reachability of each point of trajectories contained in reach set approximation  $\mathcal{R}(\hat{x}(t_i), t_i, t_{i+1})$ , the red shades denotes unreachable parts of trajectories and the green part denotes reachable parts of trajectories.
2. *Layer 1 (top)* - the top layer is completely reachable.
3. *Layer 2* - the portion of layer is unreachable, note that some frontal cells are reachable, because there exists trajectories from first layer.
4. *Layer 3 (middle)* - the middle layer is almost completely unreachable due the intruder occupation of neighbouring layers, intruder trajectory is denoted by blue line.
5. *Layer 4* - the portion of layer is unreachable due the vehicle body, note that some frontal cells are reachable due the existence of trajectories leading through fifth layer.
6. *Layer 5 (bottom)* - the bottom layer is completely reachable

### 6.3.5 Uncertainty spread - Static

*Intruder linear model*  $\vec{x}(t) = f(\vec{x}, \vec{v})$  have been introduced by 5.14, its simple linear trajectory model, base intersection structure is *ellipsoid*  $E(\vec{x}(t), \vec{v})$  (5.17). The *base intersection probability* for single intruder  $i_k$  defined as  $P_T(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  is given by 5.31. Intruders  $i_k$  intersection times for single cell  $c_{i,j,k} \in \mathcal{A}(t_i)$  are defined as follow:

1. *Intruder time of entry*  $i_e(i_k, c_{i,j,k})$  (5.33).
2. *Intruder time of leave*  $i_l(i_k, c_{i,j,k})$  (5.34).

*Single Intruder intersection model*  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.36) set up with following flags: spread flag  $s = 1$ . Because the conical intersection covers a huge amount of space in avoidance grid  $\mathcal{A}(t_i)$  it is recommended not to overshoot intruder *horizontal spread*  $\theta$  and intruder *vertical spread*  $\varphi$  estimates.

*Intruder obstacle probability*  $P_{O_I}(c_{i,j,k})$  (5.37) was not considered in this simulation, because single intruder was used. Overall implementation is straightforward transcription of mentioned definitions.

Figure 6.10: Obstacle space for static uncertainty spread

Uncertainty cone intersection is creating *virtual obstacles*, because real obstacle is moving intruder body. The cone is denoted as *blue* line borders, with notion of vehicle linear trajectory (center axis of cone). Obstacle probability is more concentrated at the detection point (upper part of diagrams) and becomes more spread and thin with increasing horizontal  $\theta$  and horizontal  $\varphi$  spread. The intruder clash (obstacle) probability distribution is in fig. 6.10 with following sub-figures:

1. *Obstacle probability* - displays impact of intruder on trajectory parts where the intersection with intruder is probable, green denotes low probability, red denotes high probability. The probability is fairly (slightly brownish) low, because of wide spread of intruder horizontal  $\theta$ , and vertical  $\varphi$  distributions.
2. *Layer 1 (top)* - there is only residual clash probability at the bottom of the diagram, due high vertical spread  $\varphi$  value.
3. *Layer 2* - there is more residual clash probability at the bottom of the diagram, due high vertical spread  $\varphi$  value.
4. *Layer 3 (middle)* - clash probability is concentrated at the top and slowly vanishes over distance from point of detection, due the high horizontal spread  $\theta$  value.
5. *Layer 4* - there is more residual clash probability at the bottom of the diagram, due high vertical spread  $\varphi$  value.
6. *Layer 5 (bottom)* - there is only residual clash probability at the bottom of the diagram, due high vertical spread  $\varphi$  value.

Figure 6.11: Reach set approximation for static uncertainty spread - all layers

Reach set  $\mathcal{R}(\hat{x}, t_i, t_{i+1})$ (fig. 6.11) is given as follow:

1. *Reachability probability* - The reachability probability of trajectories reflects obstacle set  $\mathcal{O}(t_i)$
2. *Layer 1 (top)* - the residual obstacle probability caused from extreme values of *vertical spread*  $\varphi$  are causing lowered reachability in lower portions of cells. The upper part is relatively safe.

3. *Layer 2* - the residual obstacle probability caused from extreme values of *vertical spread*  $\varphi$  are causing significantly lowered reachability in lower portions of cells. The upper part is still safe.
4. *Layer 3 (middle)* - reflects the middle layer in fig. 6.10.
5. *Layer 4* - the residual obstacle probability caused from extreme values of *vertical spread*  $\varphi$  are causing significantly lowered reachability in lower portions of cells. The upper part is still safe.
6. *Layer 5 (bottom)* - the residual obstacle probability caused from extreme values of *vertical spread*  $\varphi$  are causing lowered reachability in lower portions of cells. The upper part is relatively safe.

### 6.3.6 Uncertainty spread - Timed

*Intruder linear model*  $\vec{x}(t) = f(\vec{x}, \vec{v})$  have been introduced by 5.14, its simple linear trajectory model, base intersection structure is *ellipsoid*  $E(\vec{x}(t), \vec{v})$  (5.17). The *base intersection probability* for single intruder  $i_k$  defined as  $P_T(i_k(\vec{x}_s, \vec{v}, \theta, \varphi), c_{i,j,k})$  is given by 5.31. Intruders  $i_k$  intersection times for single cell  $c_{i,j,k} \in \mathcal{A}(t_i)$  are defined as follow:

1. *Intruder time of entry*  $i_e(i_k, c_{i,j,k})$  (5.33).
2. *Intruder time of leave*  $i_l(i_k, c_{i,j,k})$  (5.34).

*Single Intruder intersection model*  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.36) set up with following flags: spread flag  $s = 1$ , timed  $\tau = 1$ . This settings will intersection cells where is zero chance to meet an intruder with given vehicle dynamics. This option is significantly reducing obstacle space  $\mathcal{O}(t_i)$ , but if and only if the assumed intruder and vehicle velocities holds static values during collision frame.

*Intruder obstacle probability*  $P_{O_I}(c_{i,j,k})$  (5.37) was not considered in this simulation, because single intruder was used.

Figure 6.12: Obstacle space for static uncertainty spread

Similar to fig. 6.10 only time aspect is now accounted.

Figure 6.13: Reach set approximation for static uncertainty spread - all layers

Similar to fig. 6.11 only time aspect is now accounted.

## 6.4 Static obstacles avoidance

*Static obstacle avoidance* testing scenario has been introduced in [? ]. This chapter will introduce only a brief introduction to the test. The vehicle equipped with LiDAR sensor and ADS-B transceiver/receiver is flying in uncharted territory and executing a mission consisting from waypoint set  $\mathcal{WP}$ . The simulation is executed in mission local coordinate frame, where the vehicle initial position is center of frame and vehicle heading is aligned with main axis.

The ordered waypoint set  $\mathcal{WP}$  is defined as follow:

1. *Starting waypoint*  $W_1$  -  $[0, 0, 0]$  - not necessary, just to emphasize the local coordinate frame center
2. *First obstacle waypoint*  $W_2$  -  $[20, 0, 0]$ .
3. *Second obstacle waypoint*  $W_3$  -  $[20, 20, 0]$ .
4. *Third obstacle waypoint*  $W_4$  -  $[0, 20, 0]$ .
5. *Final waypoint*  $W_5$  -  $[0, 0, 10]$ .

The obstacle set  $\mathcal{O}$  is defined as set of ball obstacles  $\mathcal{O}_{\mathcal{B}}(\vec{p}, r)$ , where  $\vec{p}$  indicates obstacle center position and  $r$  stands for obstacle radius. The testing obstacle set was defined as follow set of map/detected obstacles  $\mathcal{O}_M = \mathcal{O}_D$ :

1. *First obstacle*  $\mathcal{B}_{\mathcal{O}}(1)$  -  $\mathcal{O}_{\mathcal{B}}(\vec{p} = [10, 0, 0], r = 2)$
2. *Second obstacle*  $\mathcal{B}_{\mathcal{O}}(2)$  -  $\mathcal{O}_{\mathcal{B}}(\vec{p} = [20, 10, 0], r = 2)$
3. *Third obstacle*  $\mathcal{B}_{\mathcal{O}}(3)$  -  $\mathcal{O}_{\mathcal{B}}(\vec{p} = [10, 20, 0], r = 2)$

Other simulation parameters are stated like follows:

1. *Vehicle model*  $\dot{x} = f(x, u)$  is given by eq. 4.4.
2. *Vehicle control*  $\mathcal{MA}$  is defined in section 4.3.
3. *Vehicle body radius*  $r_b$  is given as 30 cm.
4. *Vehicle turning radius*  $r_t$  is given as 2 m, this parameter impacts the decision time  $t_i$  selection based on the in-veritable crash distance [? ].
5. *Movement automaton prediction error*  $E_p = (\mathcal{MA})$  is equal to 30 cm.
6. *Safety margin*  $s_m$  is set to 60 cm.
7. *Avoidance grid*  $A(t_i)$  properties were set as follow:
  - a. start distance  $d_s$  0 m,
  - b. end distance  $d_e$  10 m,
  - c. step distance  $s_d$  1 m (10 layers),
  - d. horizontal span from  $\theta_s$  as  $-\pi/4$  to  $\theta_e$  to  $\pi/4$ ,
  - e. horizontal cell count  $c_h$  as 7,
  - f. vertical span from  $\varphi_s$  as  $-\pi/6$  to  $\varphi_e$  to  $\pi/6$ ,
  - g. vertical cell count  $c_v$  as 5 (layer count).

*Note:* The obstacles are proposed as convex obstacles, the real world obstacles usually breaches the convex surface assumption, the approach was tested on non convex obstacles and open traps. The simplicity of testing example is given to show main properties of approach. Comparison to deterministic approach performance can be made by comparing section 6 of [? ].

Legend of simulation results is following:

1. *Executed trajectory* (blue line) - executed trajectory by vehicle at moment of snapshot
2. *Planned trajectory* (red line) - planned trajectory within avoidance grid  $\mathcal{A}(t_i)$ , trajectory is re-planned every decision time  $t_i$ . Note that in case of this simulation the trajectory was re-planned every movement.
3. *Waypoints* (green square markers) - mission plan for multiple waypoints.
4. *Obstacles* (red point clouds) - obstacle sets break into point clouds.
5. *Field of vision* (black dashed line) - denotes current avoidance grid  $\mathcal{A}(t_i)$  boundary starting at vehicle position at time of snapshot.

Following decisive moment have been taken into consideration when making snapshots:

1. *First obstacle set  $\mathcal{B}_0(1)$  avoidance* (fig. 6.14) - vehicle is heading straight to the obstacle center, the moment captured is when vehicle is avoiding the obstacle. The obstacle is fully covered by field of the vision and vehicle is avoiding it from right side without breaching the safety margin  $s_m$ .
2. *Second obstacle set  $\mathcal{B}_0(2)$  avoidance* (fig. 6.15) - The vehicle is avoiding the second obstacle after the turn, the obstacle is partially in vehicles FOV. The vehicle will crash if there was not a movement restriction given by restriction of reachability property  $P_R(c_{i,j,k}), \forall c_{i,j,k} \in \mathcal{A}(t_i)$
3. *Third obstacle set  $\mathcal{B}_0(3)$  avoidance* (fig. 6.16). - Same as second obstacle. Please note that the cost function  $J^*(\mathcal{T}(\vec{x}_0, B))$  is designed to prioritize planar movements. The horizontal/vertical separation was not used in this framework and cost can impact the decision making.
4. *Vehicle trajectory after mission competition* (fig. 6.17) - The overview of trajectory and decisions, it is assumed that all paths between  $W_4$  and  $W_5$  are not obscured, this is to show, that avoidance grid  $\mathcal{A}(t_i)$  can generate smooth trajectories in free space.

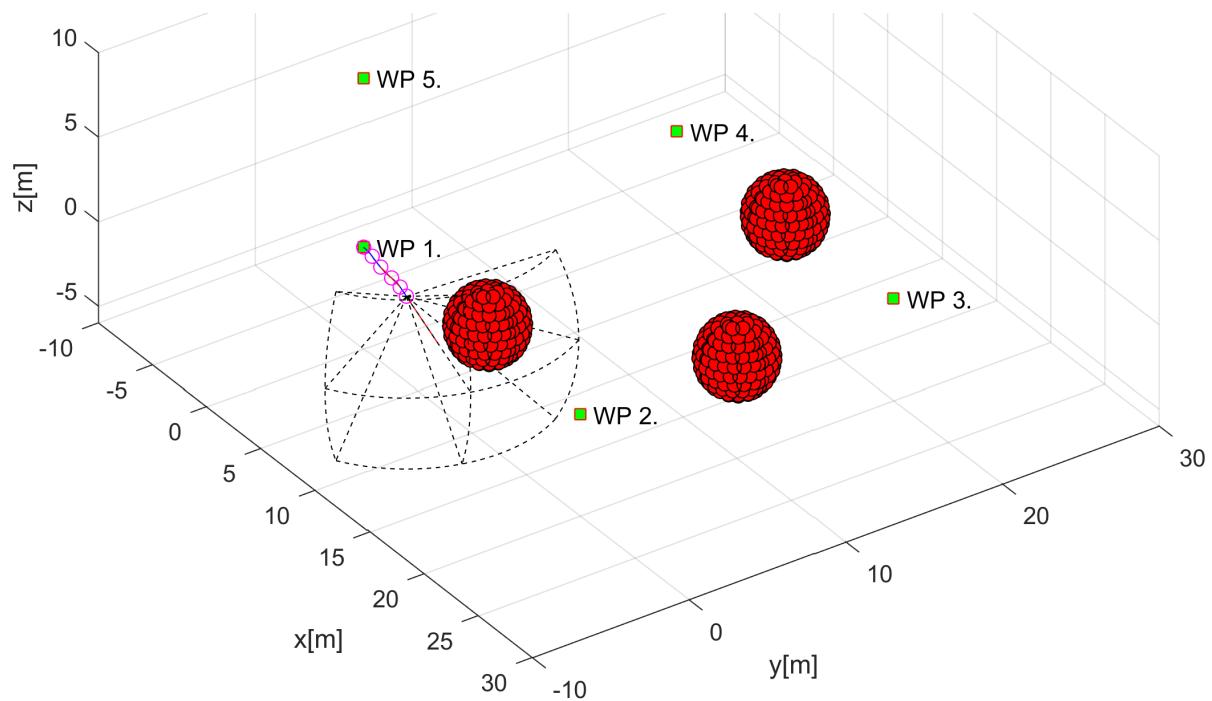


Figure 6.14: First obstacle set  $\mathcal{B}_0(1)$  avoidance

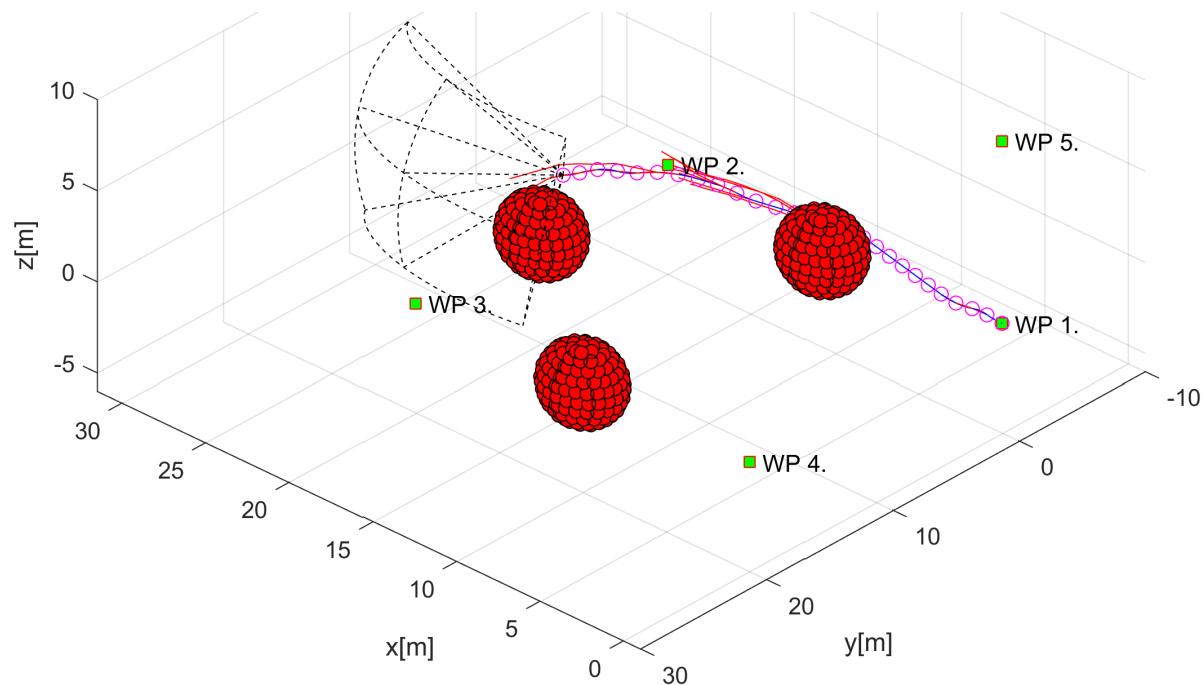


Figure 6.15: Second obstacle set  $\mathcal{B}_0(2)$  avoidance

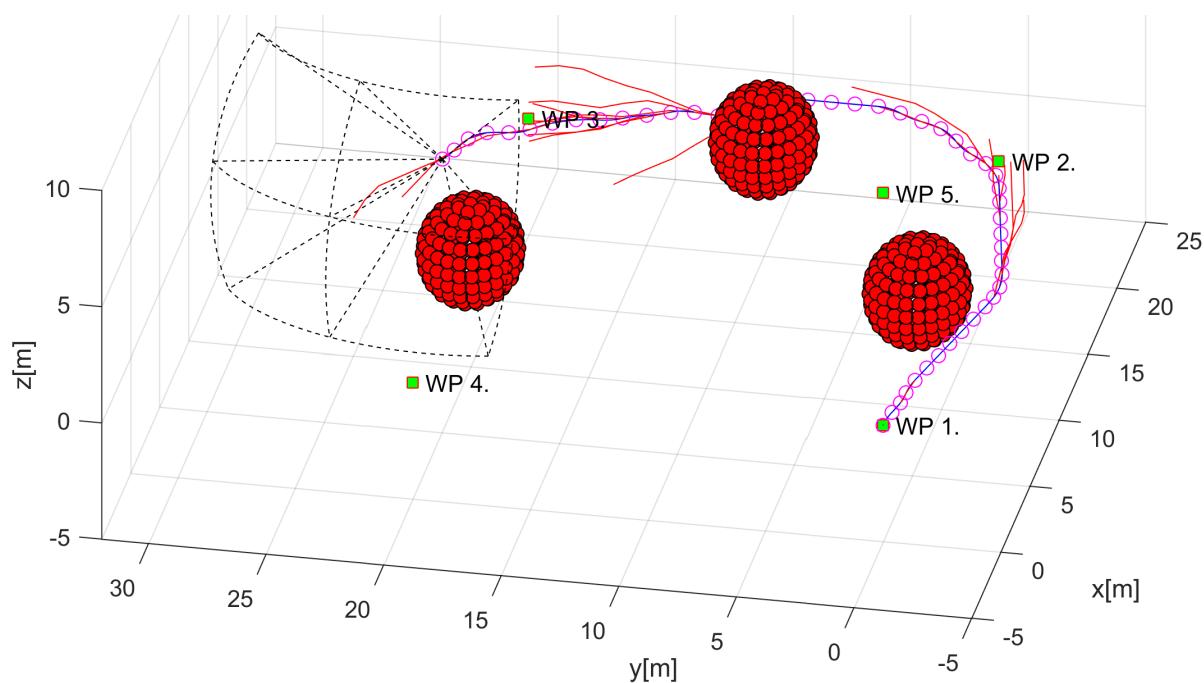


Figure 6.16: Third obstacle set  $\mathcal{B}_0(3)$  avoidance

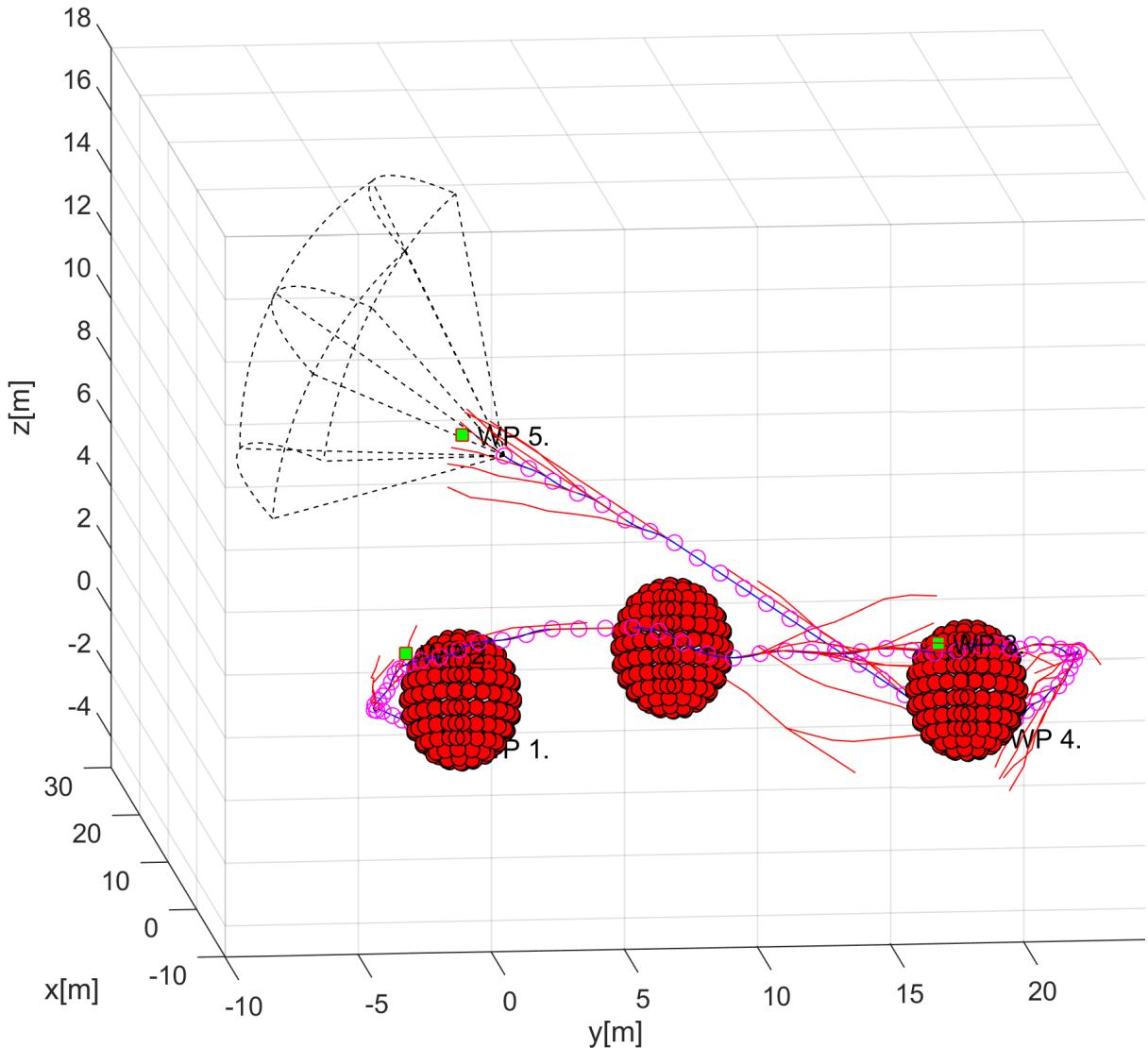


Figure 6.17: Vehicle trajectory after mission competition

#### 6.4.1 Probabilistic model evaluation

The *general algorithm* (4.19,4.20), defines multiple probabilities for cells  $c_{i,j,k} \in \mathcal{A}$  and trajectories  $\mathcal{T} \in \mathcal{R}$ , in case of static obstacle avoidance the key focus probabilities are following:

1. *Reachability probability*  $P_R(c_{i,j,k})$ ,  $P_R(\mathcal{T}(x_0, B))$  - the reachability was impacted by obstacle space  $\mathcal{O}$  and visibility, for sake of simplicity the reachability was reaching only two values  $P_R \in \{0, 1\}$ .
2. *Obstacle probability*  $P_O(c_{i,j,k})$  - obstacle probability in cell was assessed using general algorithm (4.19,4.20) and detected obstacle probability  $P_{O_D}$  (5.38).
3. *Visibility probability*  $P_V(c_{i,j,k})$  - calculated according to standard visibility formula  $P_V(c_{i,j,k})$  (5.47).

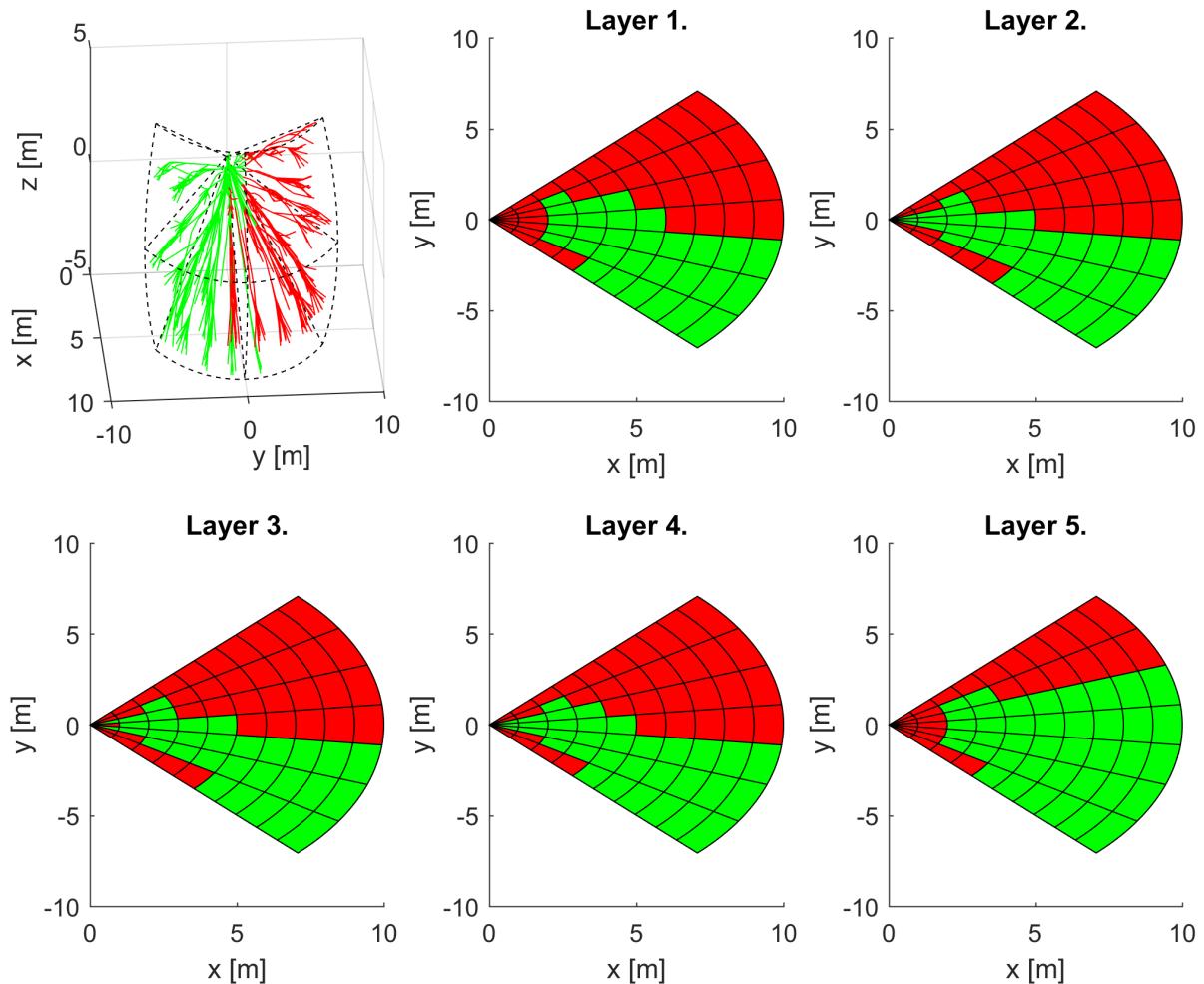


Figure 6.18: Reachability probability assessment for  $\mathcal{B}_0(1)$  avoidance

For intersection please refer to *first obstacle set  $\mathcal{B}_0(1)$  avoidance* (fig. 6.14). The obstacle space is slightly in right-upper quadrant (from vehicles viewpoint). The red trajectories are leading to obstacle or trough obstacle space, the green trajectories are leading trough or to safe space. *Layer 1.-5.* right part is not reachable (red cells), there is possible escape paths on left side (green cells).

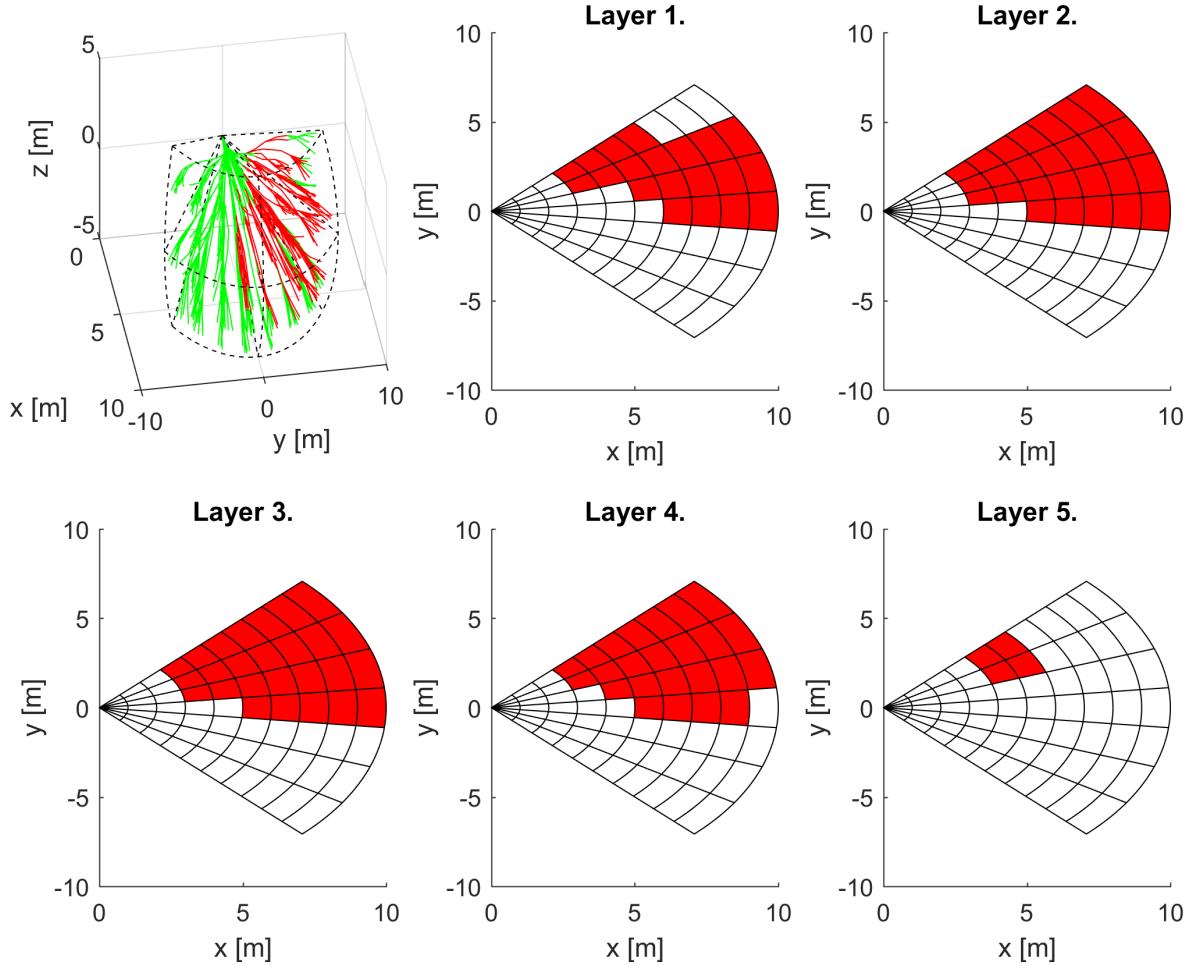


Figure 6.19: Obstacle probability assessment for  $\mathcal{B}_0(1)$  avoidance

For intersection please refer to *first obstacle set  $\mathcal{B}_0(1)$  avoidance* (fig. 6.14) The obstacle space is slightly in right-upper quadrant (from vehicles viewpoint). The red trajectories are leading to obstacle or trough obstacle space, the green trajectories are leading trough or to safe space. The obstacle space accounts only visible part of the obstacle, therefore in *layer 1. and 4.* is space which is considered without obstacle (white cells). The obstacle space (red cells) in bottom layer (*layer 5.*) is very small due the vehicle is tilted down a little in time of obstacle assessment.

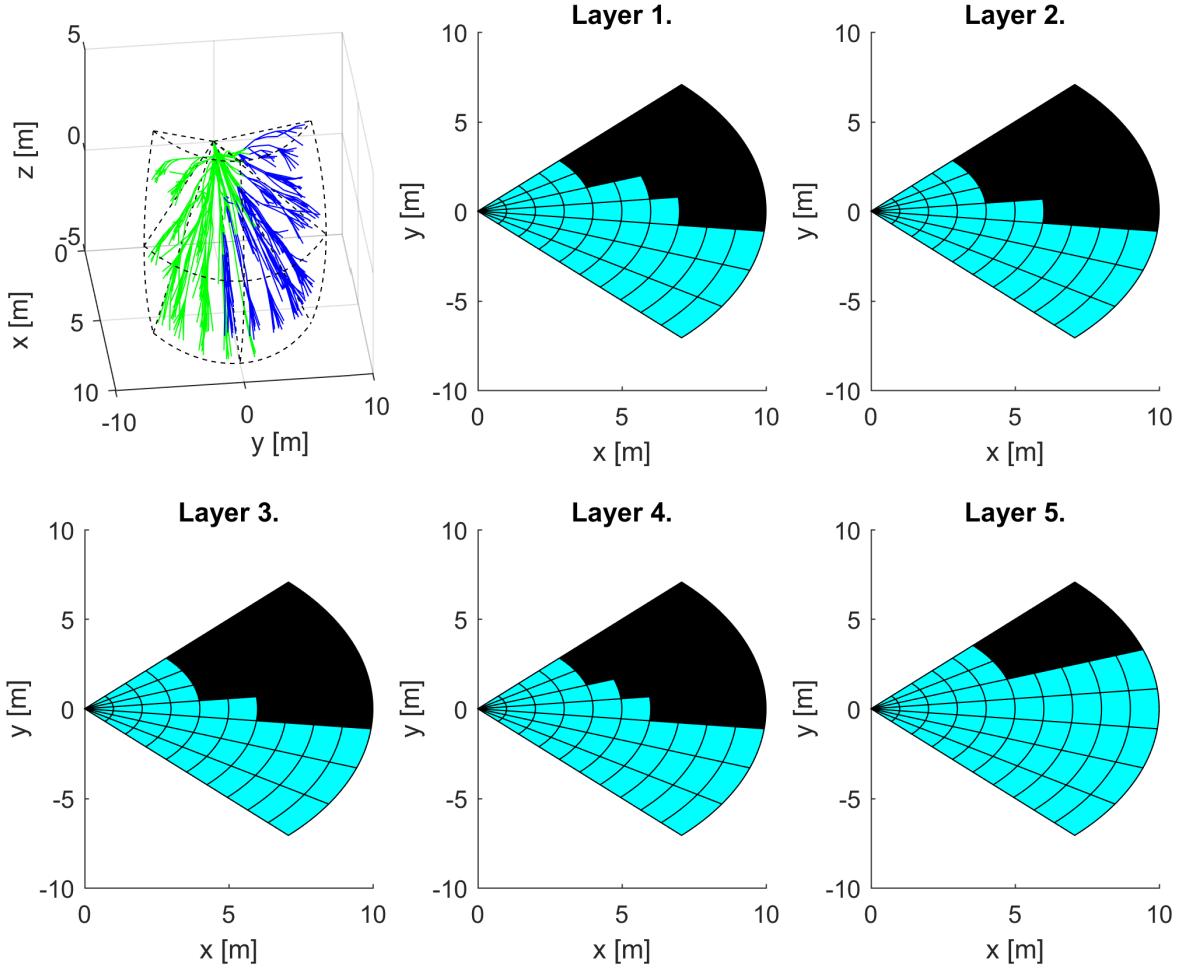


Figure 6.20: Visibility probability assessment for  $\mathcal{B}_0(1)$  avoidance

For intersection please refer to *first obstacle set  $\mathcal{B}_0(1)$  avoidance* (fig. 6.14). The *visibility space is obscured*

### 6.4.2 Avoidance performance

*Obstacle avoidance performance* for autonomous systems is evaluated by multiple criteria. A necessary condition is *crash distance  $d_c$  does not cross safety margin  $s_m$* . In the other world that vehicle does not crash into static obstacle.

*The sufficient conditions* are following:

1. *Energy consumption* - For each system  $\dot{x} = f(x, u)$  cost functional  $J(x, u, t)$  can be proposed to reflect system performance or energy consumption. This functional is therefore used in optimization maximization/minimization problem.
2. *Trajectory tracking* - Trajectory of vehicle is given by ordered set of waypoints  $WP = \{W_1, W_2, \dots, W_i\}$ . The given trajectory may not be feasible for vehicle system  $\dot{x} = f(x, u)$ . To differentiate the performance of tracking Mean Square error is introduced like following:

$$e = \sqrt{\left( (\hat{x}(t) \rightarrow \mathbb{R}^k) - (x(t) \rightarrow \mathbb{R}^k) \right)^2} \quad (6.13)$$

Where  $(\hat{x}(t) \rightarrow \mathbb{R}^k)$  is expected parameters projected to space  $\mathbb{R}^k$  and  $(x(t) \rightarrow \mathbb{R}^k)$  is real system performance projected to same space. Usually performance parameter is defined as deviation from projected trajectory  $\tilde{x}(t) \in \mathbb{R}^3$ .

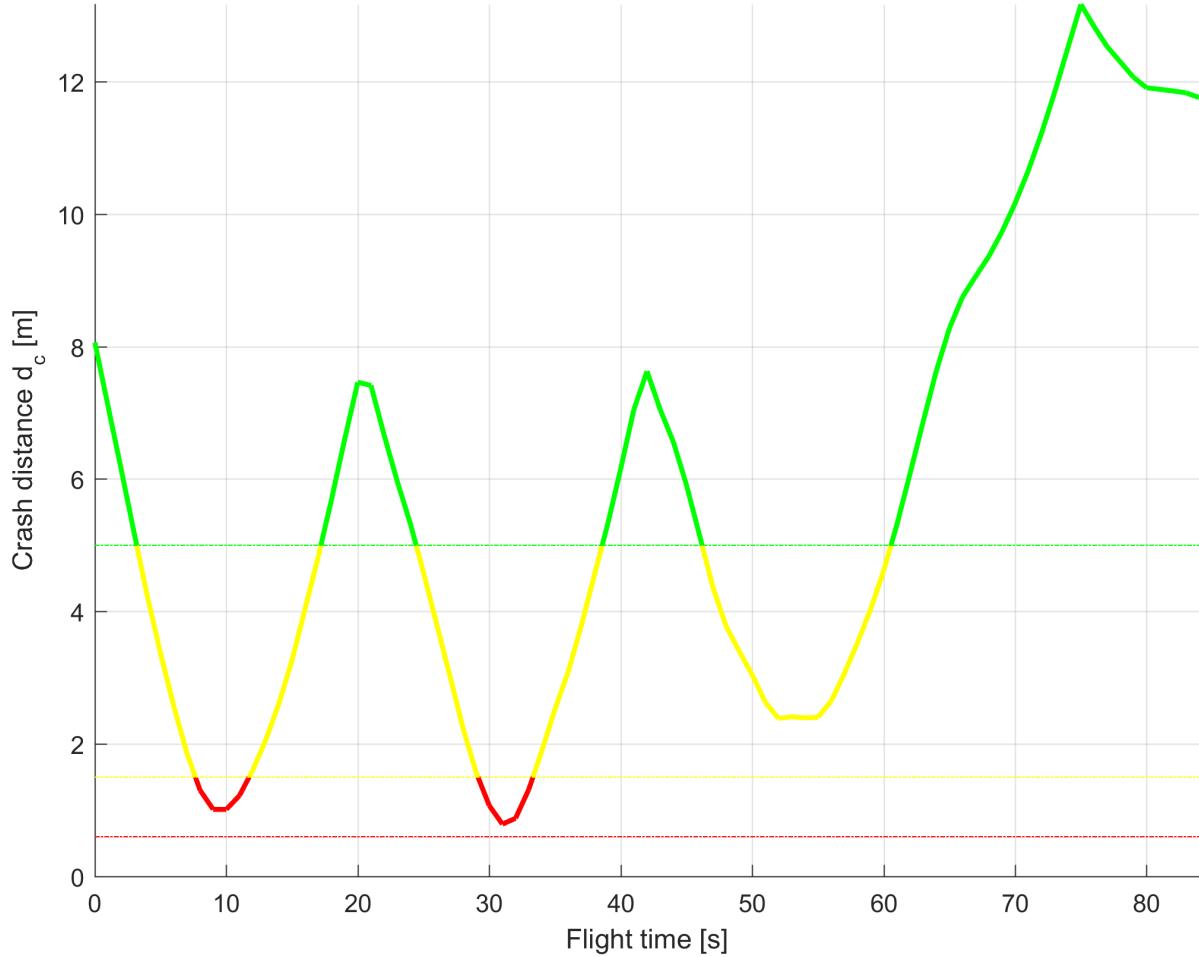


Figure 6.21: Crash distance  $c_r$  evolution during mission

*Crash distance to nearest obstacle evolution* (fig 6.21) displays how far vehicle was to nearest obstacle during the mission execution. There are three lines representing:

1. *Conservative methods margin  $c_m$*  (green dashed line) - conservative methods are based on simple preposition *turn back while you can*, therefore conservatives methods margin  $c_m$  is calculated like follow:

$$c_m = \frac{2 \times r_t + 3 \times r_v}{4.6m} \quad (6.14)$$

Where  $r_t$  is turning radius of vehicle and  $r_v$  is radius of vehicle. The conservative methods margin  $c_m$  is 4.6 meters for given simulation parameter.

2. *Adaptive methods margin  $a_m$*  (yellow dashed line) - are more energy consumption focused. The focus is on not deviating from optimal trajectory as long as possible. Typical representative of adaptive algorithms is potential field avoidance algorithm.

The *potential field avoidance* principle is following: Each obstacle (static or moving) has a potential like a magnet, our vehicle has inverse potential. The magnitude of potentials can change over time depending on various factors (like the danger level etc.). The margin of adaptive methods was calculated by following formula:

$$a_m = 6 \times r_v = 6 \times 0.3 = 1.8m \quad (6.15)$$

Where  $r_v$  stands for vehicle radius. The adaptive methods margin  $a_m$ , where the *potential field method* is key comparison element is 1.8 m

3. *Safety margin*  $s_m$  (red dashed line) has been discussed in section 6.4. Safety margin of our method  $s_m$  is set as 60cm

*The crash distance*  $c_r t$  never breaches  $d_c(t) \geq s_m$  therefore *necessary condition* is satisfied. The *proposed approach* outperforms *conservative methods*  $\exists t, d_c(t) \leq c_m$  (yellow part of  $d_c$  evolution) also outperforms *adaptive methods*, because  $\exists t, d_c(t) \leq a_m$ .

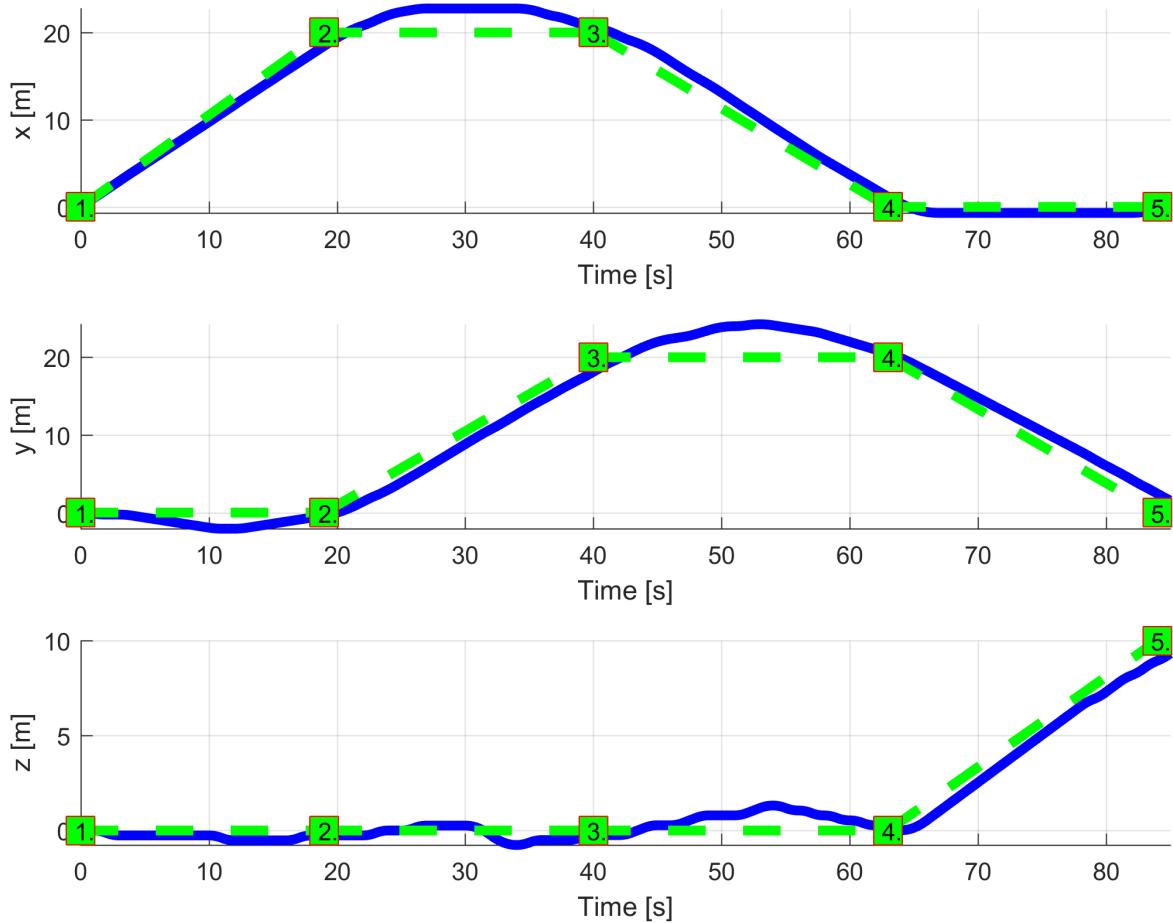


Figure 6.22: Trajectory tracking performance

*Trajectory tracking performance* (fig. 6.22) displays trajectory tracking problem divided into state tracking problem of  $x(t)$ ,  $y(t)$ , and  $z(t)$ . The ordered waypoint set  $WP$  is displayed as numbered green squares. The *desired trajectory*  $\hat{x}(t)$ ,  $\hat{y}(t)$ , and  $\hat{z}(t)$  is displayed as green dashed line between waypoints. The *vehicle performance*  $x(t)$ ,  $y(t)$ , and  $z(t)$  is displayed as blue line.

*Overall performance* is optimal in avoidance grid  $\mathcal{A}(t_i)$  and semi-optimal in joined trajectory  $\mathcal{T}(x_0, B_1, \dots, B_i)$  [? ]. This property transits from deterministic approach and its kept based on given threshold.

## 6.5 Intruders avoidance

*Intruder avoidance* have been studied on set of avoidable intruders which maneuverability and velocity was smaller or equal to vehicle velocity and maneuverability. The ordered waypoint set  $\mathcal{WP}$  is defined as follow:

1. *Starting waypoint*  $W_1 - [0, 0, 0]$  - not necessary, just to emphasize the local coordinate frame center
2. *Final waypoint*  $W_2 - [55, 0, 0]$ .

Other simulation parameters are stated like follows:

1. *Vehicle model*  $\dot{x} = f(x, u)$  is given by eq. 4.4.
2. *Vehicle control*  $\mathcal{MA}$  is defined in section 4.3.
3. *Vehicle body radius*  $r_b$  is given as 30 cm.
4. *Intruder horizontal spread* is estimated as  $\pi/16$  [rad] for each intruder  $i_k \in \mathcal{I}$ .
5. *Intruder vertical spread* is estimated as  $\pi/24$  [rad] for each intruder  $i_k \in \mathcal{I}$ .
6. *Vehicle turning radius*  $r_t$  is given as 2 m, this parameter impacts the decision time  $t_i$  selection based on the in-veritable crash distance [? ].
7. *Movement automaton prediction error*  $E_p = (\mathcal{MA})$  is equal to 30 cm.
8. *Safety margin*  $s_m$  is set to 60 cm.
9. *Avoidance grid*  $A(t_i)$  properties were set as follow:
  - a. start distance  $d_s$  0 m,
  - b. end distance  $d_e$  10 m,
  - c. step distance  $s_d$  1 m (10 layers),
  - d. horizontal span from  $\theta_s$  as  $-\pi/4$  to  $\theta_e$  to  $\pi/4$ ,
  - e. horizontal cell count  $c_h$  as 7,
  - f. vertical span from  $\varphi_s$  as  $-\pi/6$  to  $\varphi_e$  to  $\pi/6$ ,
  - g. vertical cell count  $c_v$  as 5 (layer count).
10. *Intruders set*  $\mathcal{I}$  is expanding over mission time  $t \in [t_s, t_e]$  and each intruder  $i \in \mathcal{I}$  is detected at least at border of FOV.

*Intruder set  $\mathcal{I}$*  (tab 6.2) contains set of eight intruders  $i_1, \dots, i_8$  which are discovered at mission times  $t = 5, \dots, 40$  seconds. The intruder initial position  $\vec{x}_l(t_D)$  is in 3D local coordinate frame given by vehicle  $\vec{x}(t)$  (4.4) position and orientation at time of detection  $t_D$ . The same goes for intruder velocity vector  $\vec{v}_l(t_D)$ . The intruder model (5.14) is simple linear line model.

The intruder intersection model  $P_{O_I}(i_k, c_{i,j,k}, l, b, s, \tau)$  (5.35,5.36) has following intersection settings:

1. *Linear intersection mode l* set as *true*.
2. *Body volume intersection mode b* set as *false*.
3. *Ellipsoid cone intersection mode s* set as *true*.
4. *Timed intersection mode  $\tau$*  *false*.

*Note:* All intruders are attacking vehicle from *left flank* and are discovered at the margin of the FOV. All intruders have speed equal to vehicle speed  $v_v$  and its  $1\text{ ms}^{-1}$ .

Intruder	$t_D$	$\vec{x}_l(t_D)$	$\vec{v}_l(t_D)$
$i_1$	5 s	$[6, 8, -0.5]^T$	$[0, -1, 0]^T$
$i_2$	10 s	$[6, 8, -0.5]^T$	$[0, -1, 0]^T$
$i_3$	15 s	$[6, 8, -0.5]^T$	$[0, -1, 0]^T$
$i_4$	20 s	$[6, 8, -0.5]^T$	$[0, -1, 0]^T$
$i_5$	25 s	$[6, 8, 0.5]^T$	$[0, -1, 0]^T$
$i_6$	30 s	$[6, 8, 0.5]^T$	$[0, -1, 0]^T$
$i_7$	35 s	$[6, 8, 0.5]^T$	$[0, -1, 0]^T$
$i_8$	40 s	$[6, 8, 0.5]^T$	$[0, -1, 0]^T$

Table 6.2: Intruder set  $\mathcal{I}$ .

The avoidance of intruders  $i_1, i_2, \dots, i_8 \in \mathcal{I}$  (tab. 6.2). is shown in figures 6.23 and 6.24. where:

1. *Executed trajectory* (blue line) is trajectory where our vehicle flew in final, this trajectory is keeping safety property ( $s_m \geq 1.2m$ ).
2. *Planned trajectory* (red line) - at some decision time  $t_i$  in avoidance grid  $\mathcal{A}(t_i)$  The most feasible trajectory to safely escape is chosen. Because of changing situation, popping up of new intruders in this case, the escape trajectory is often changed (red lines fig. 6.26).
3. *Avoidance grid* (black dashed line boundary) - avoidance grid is boundary is given by vehicle position and orientation  $x(t) \rightarrow \mathbb{R}^6$ . Its depending on expected vehicle state  $\hat{x}(t_i)$  at time of avoidance  $t_i$ .The boundary of avoidance grid  $\mathcal{A}(=)$  is in fig. 6.22. Plotting avoidance grid  $\mathcal{A}$  for each time of decision  $t_i$  is pointless, because it will overshadow more important information like, intruder position and planned trajectory.
4. *Decision point* (empty magenta circle) - shows when vehicle generated new avoidance grid  $\mathcal{A}$  for expected state of vehicle  $\hat{x}(t)$ .

5. *Waypoints* (numbered green square) - direct line between two waypoint is the shortest distance to fly, the waypoints must be reached in the order.
6. *Intruder position* (full red circle) - intruder  $i_k$  position at decision time  $t_i$ , this is intruder  $i_k$  expected position.
7. *Intruder trajectory* (linked full magenta circles) - intruder positions joined via line in previous decision times  $t_{i-1}, \dots, t_{i-d}$ , where  $t_{i-d}$  is detection time.

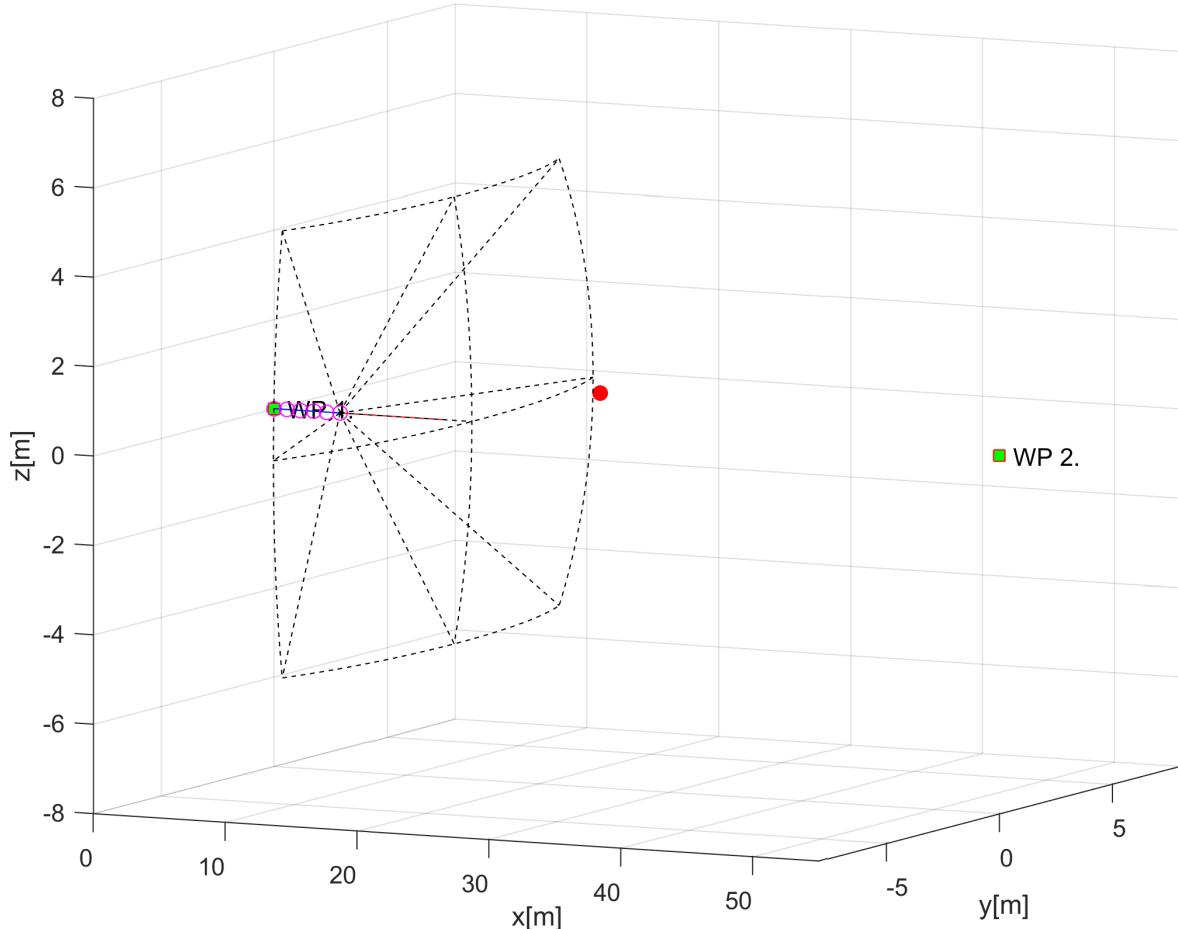


Figure 6.23: First intruder detection.

*First intruder detection* (fig. 6.23) shows intruder  $i_1 \in \mathcal{I}$  approaching vehicle (defender) from its left flank. It is guaranteed that the intruder  $i_1$  and vehicle will clash each other at time  $t_i + 4s$ , if and only if vehicle does not change course. The *vehicle* is also looking for an optimal avoidance trajectory to lower the energy consumption (reminder: simple flew trajectory was used as cost function).

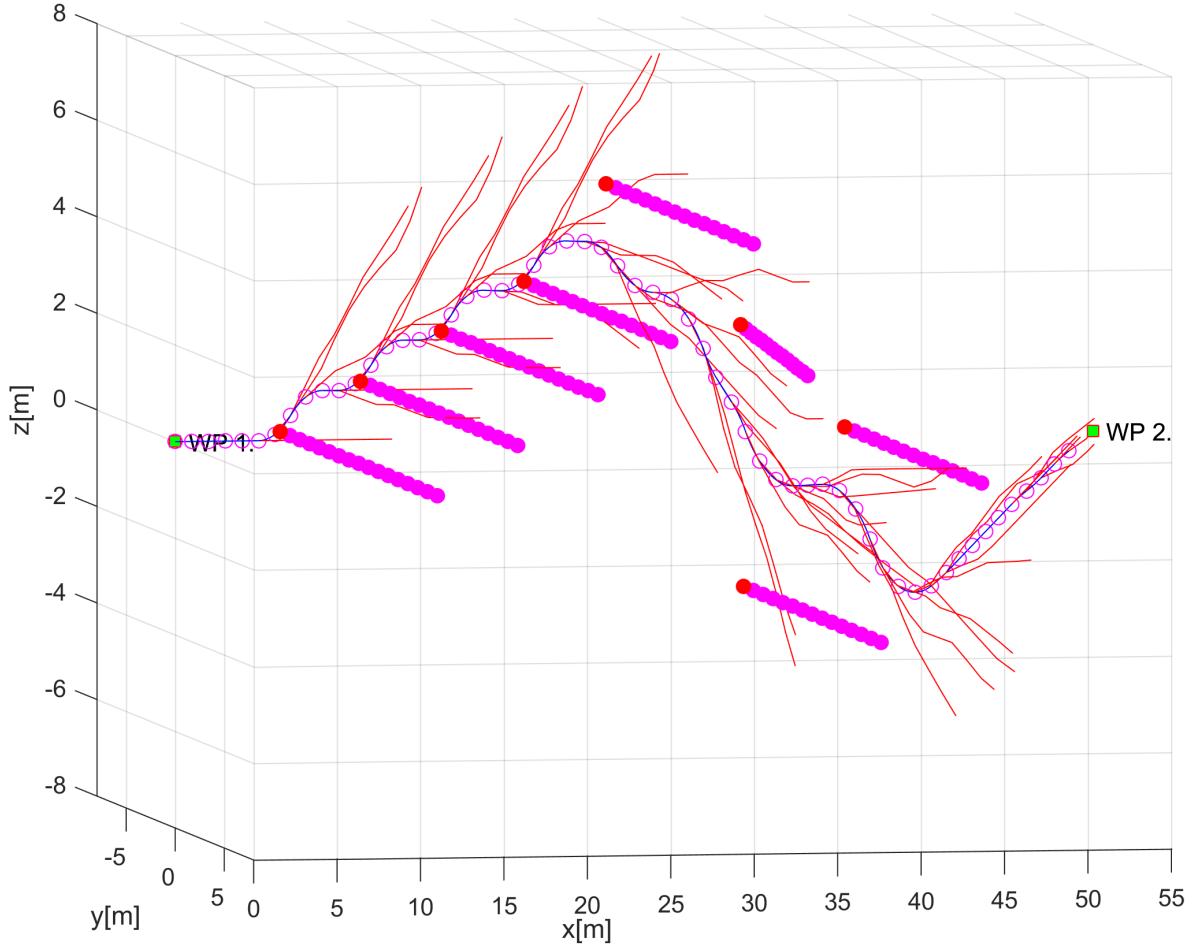


Figure 6.24: After all intruders avoidance

The final trajectory *after all intruder avoidance*  $i_1, \dots, i_8$  is given by fig.6.24. Please notice multiple trajectory change given by discovered intruders (tab. 6.2).

### 6.5.1 Intruder avoidance evaluation

Due to the best knowledge of authors, all previously discussed conservative/adaptive approaches are mixing together the static and moving obstacles. There is not an single approach which separates concept of static and concept of moving obstacles (intruders) like proposed approach.

The only important and necessary condition is safety margin, which in this case is given as:

$$s_m(i_k) = s_m + 2 \times r_i(i_k) \quad (6.16)$$

Where  $s_m$  is safety margin of *our vehicle*,  $s_m(i_k)$  is safety margin for selected intruder. The intruder safety margin is depending on intruder body volume, in this case body volume is given by radius  $r_i(i_k)$ . All intruders  $\forall i_k \in \mathcal{I}$  have body radius  $r_i$  equal to 60 cm, therefore the safety margin for any intruder avoidance  $s_m(\forall i_k) = 1.2m$ .

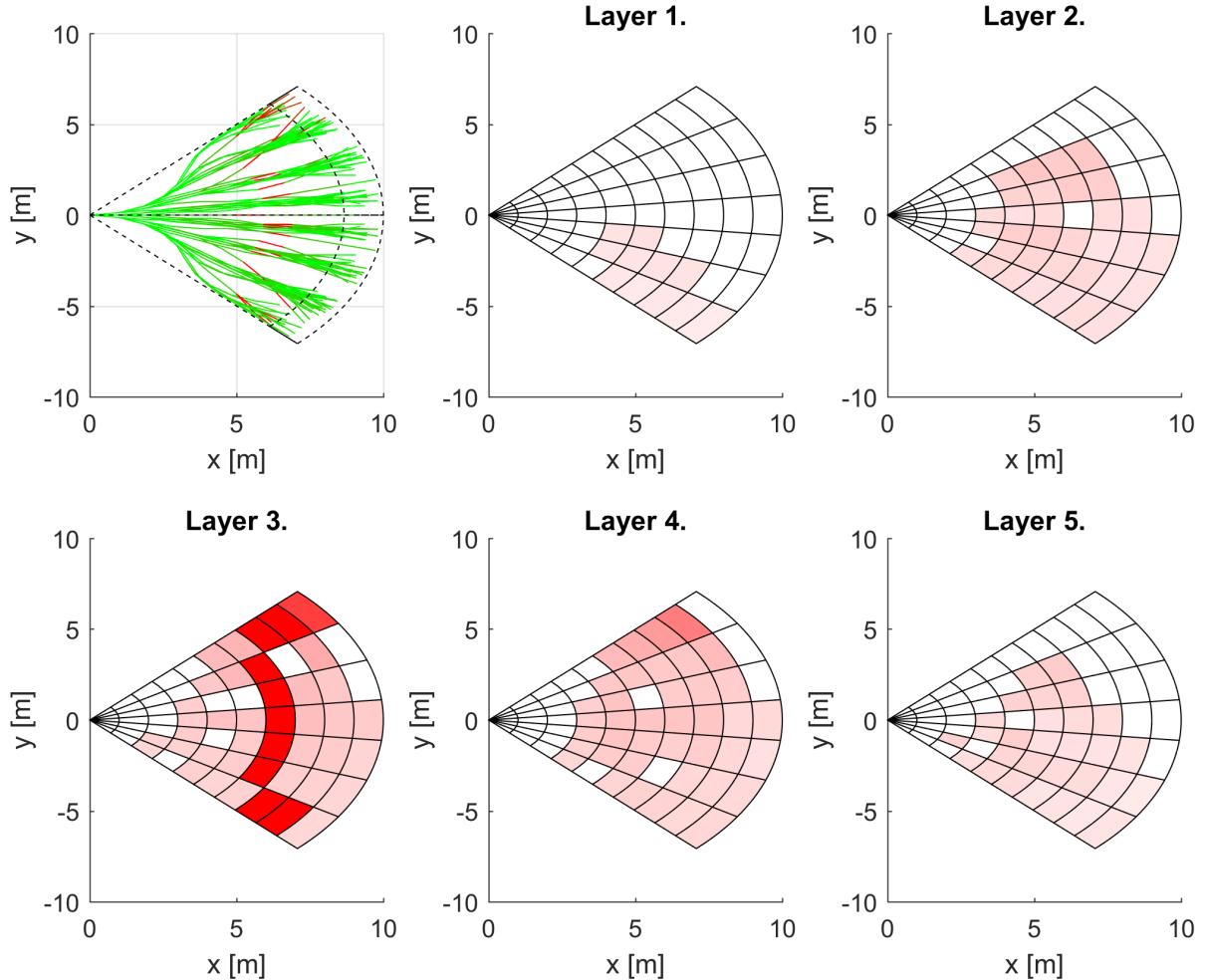


Figure 6.25: Obstacle probability after first intruder detection

*Obstacle probability*  $P_{O_I}$  (fig. 6.25) is combination of *linear intersection model* (5.9) and *conic intersection model* (5.31).

The trajectory set  $\{\mathcal{T}\} \in \mathcal{R}$  is given by first sub-figure, only direct intersection cell belonging trajectories are impacted (red). Other trajectories in cones are less impacted (shades of green).

*Layer 1.* is impacted on right side and its the least impacted layer, due the lowered intruder position  $z = -0.5$  ( $i_1$  in tab. 6.2). *Layer 2.* contains only residuals from conic intersection and only one row on left side is free for avoidance. *Layer 3.* contains direct line intersection (bright red cells) and residual conic intersection (shades of red) its one of the most impacted layers. *Layer 4.* contains conic intersection residuals (shades of red), almost all layer is covered, but the probabilities of clash are rather low (2.5%). *Layer 5.* contains conic intersection residuals, it is almost symmetrical with layer 2.

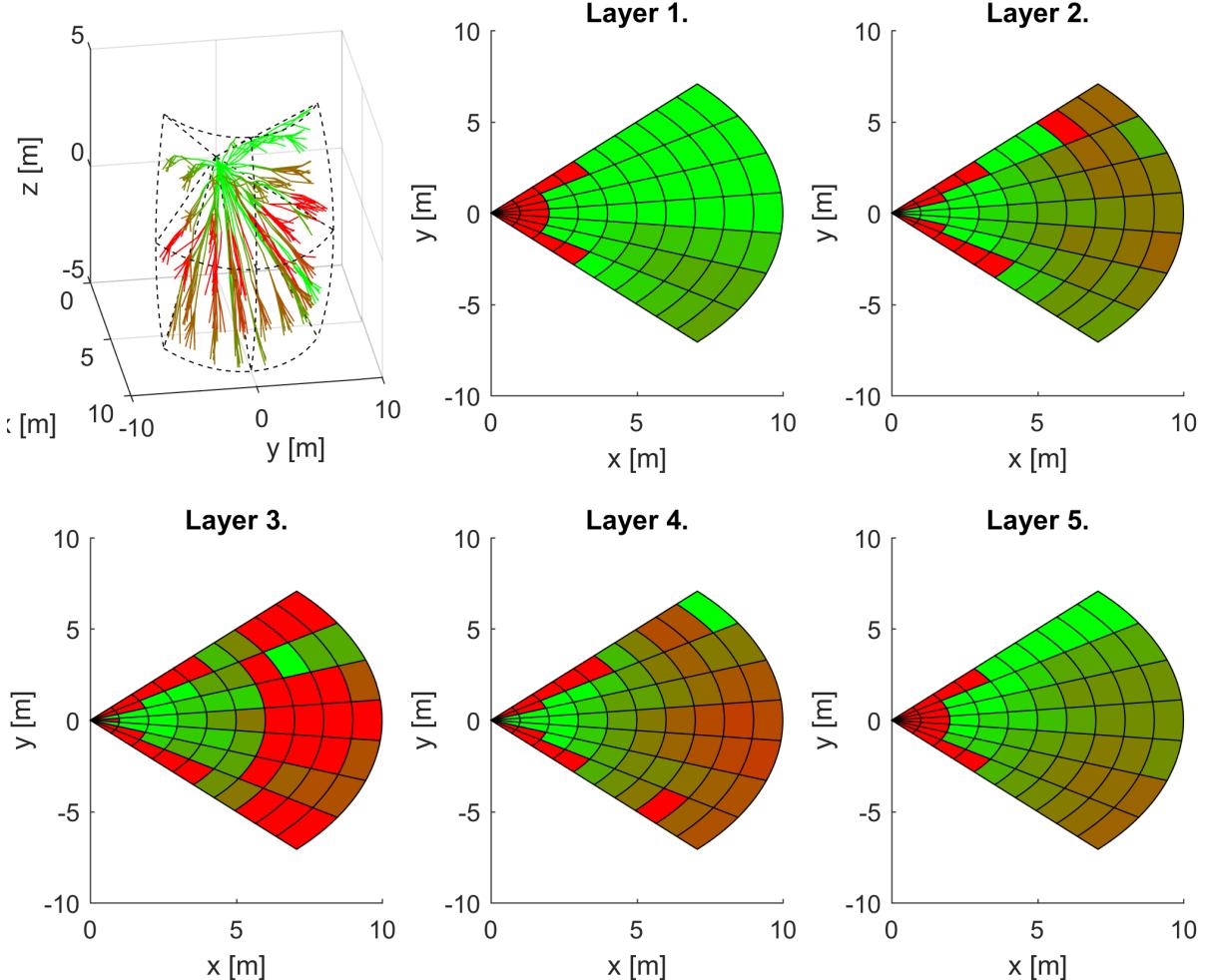


Figure 6.26: Reachability probability after first intruder detection

The reach set  $\mathcal{R}(\hat{x}(t_i), t_i, t_{i+1})$  for situation (fig. 6.23) is given at top-left sub-figure (6.26), the trajectories are colored green if reachable, brownish is semi-reachable, and red if unreachable. Please note that majority of brown trajectories are leading trough layers except intersection one (layer 3).

Reachability probability  $P_R(c_{i,j,k})$  for cells  $c_{i,j,k} \in \mathcal{A}(t_i)$  is varying due the different distribution of obstacle space (fig. 6.25). The green cells are fully reachable, the brownish cells are partially reachable (more red then worse is reachability), the red cells are unreachable. *Layer 1.* is reachable in upper half. *Layer 2.* is partially reachable on the edge, due the many routes leading trough safe *layer 1.*. *Layer 3.* is completely blocked due the linear part of obstacle intersection ( $P_{O_I} = 1$ ). *Layer 4.* is similar to *layer 2.* but there are no patches of reachable space, because *layer 5.* has only one reachable cell row  $\mathcal{C}(1, 5)$  (5.46).

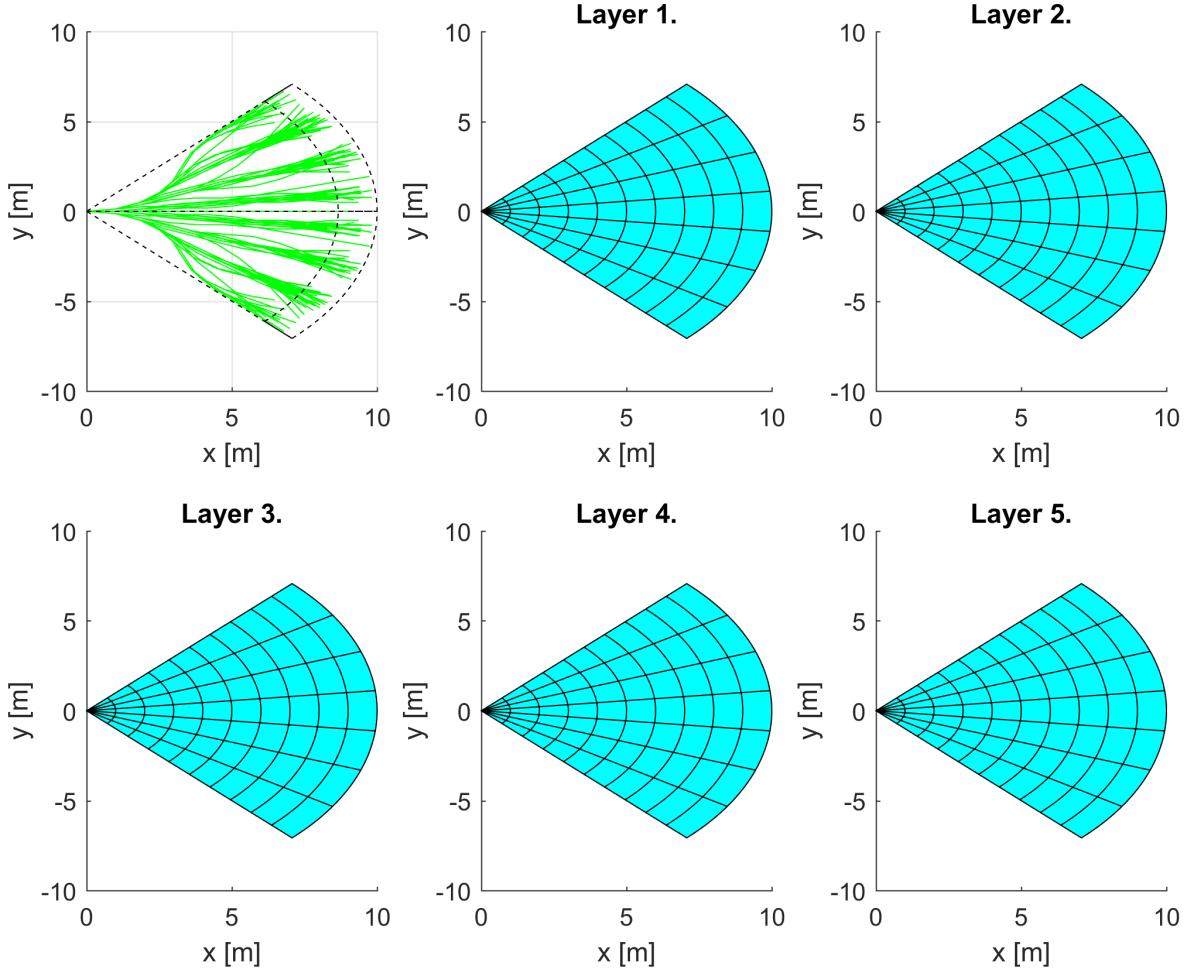


Figure 6.27: Visibility probability after first intruder detection

The visibility probability  $P_V(C_{i,j,k})$  is equal to one for all cells, because there is no real hindrance in visibility. The intruder obstacle probability  $P_{O_I}$  can be accounted as a sort of virtual obstacle to be avoided.

Concept of *virtual obstacles* is very strong and it can be used to implement virtual barriers around object of significant importance.

*Intruder collision distance*  $c_D(t)$  is a minimal distance between any intruder and vehicle trajectory at given time snapshot  $\tau$ . For fixed time  $\tau$ , there exist a set of intruder positions like follow:

$$\mathcal{I}_P(\tau) = \{\vec{p} \in \mathbb{R}^3 : p = \vec{x}(\tau) \rightarrow \mathbb{R}^3, \vec{x}(\tau) \in i_k, i_k \in \mathcal{I}(\tau)\} \quad (6.17)$$

Where  $\vec{p}$  is point in global coordinates, created by projection of intruder state  $\vec{x}(\tau)$  to *global coordinate frame*  $\mathbb{R}^3$  for each intruder  $i_k$  in detected intruders set  $\mathcal{I}(\tau)$ .

*The closest intruder*  $i_c$  in detected intruder set  $\mathcal{I}(\tau)$  for fixed time  $\tau$  is given as:

$$i_c(\tau, \vec{x}_v(\tau)) = i_k \in \mathcal{I} : \min_{\forall \vec{p}_i \in \mathcal{I}_P(\tau), \vec{x}_v(\tau)} \{\|\vec{p}_i - (\vec{x}_v(\tau) \rightarrow \mathbb{R}^3)\|\} \quad (6.18)$$

Where the  $\vec{x}_v(\tau) \rightarrow \mathbb{R}^3$  is projection of our vehicle state  $\vec{x}_v$  at fixed time  $\tau$  projected to global coordinate frame  $\mathbb{R}^3$ , the closest intruder  $i_c$  is selected based on vehicle position and set of intruders positions  $\mathcal{I}_P(\tau)$  (6.17). The norm  $\|\cdot\|$  is standard euclidean.

*Collision distance* function  $c_D(t)$  is calculated for steepest time  $\tau$  in mission time frame  $[t_s, t_e]$ , where  $t_s$  is mission start time and  $t_e$  mission end time. For fixed time  $\tau$

the minimal distance between vehicle position  $\vec{x}_v(\tau) \rightarrow \mathbb{R}^3$  in global coordinate frame  $\mathbb{R}^3$ , and intruder position  $\vec{p}_i$  from intruder set  $\mathcal{I}_P(\tau)$  used as crash distance  $c_d$  for particular  $\tau$ . The collision distance is formulated in following equation:

$$c_D(t) = \left\{ c_d \in \mathbb{R} : c_d = \min_{\forall \vec{p}_i \in \mathcal{I}_P(\tau), \vec{x}_v(\tau)} \{ \| \vec{p}_i - (\vec{x}_v(\tau) \rightarrow \mathbb{R}^3) \| \}, \forall \tau \in [t_s, t_e] \right\} \quad (6.19)$$

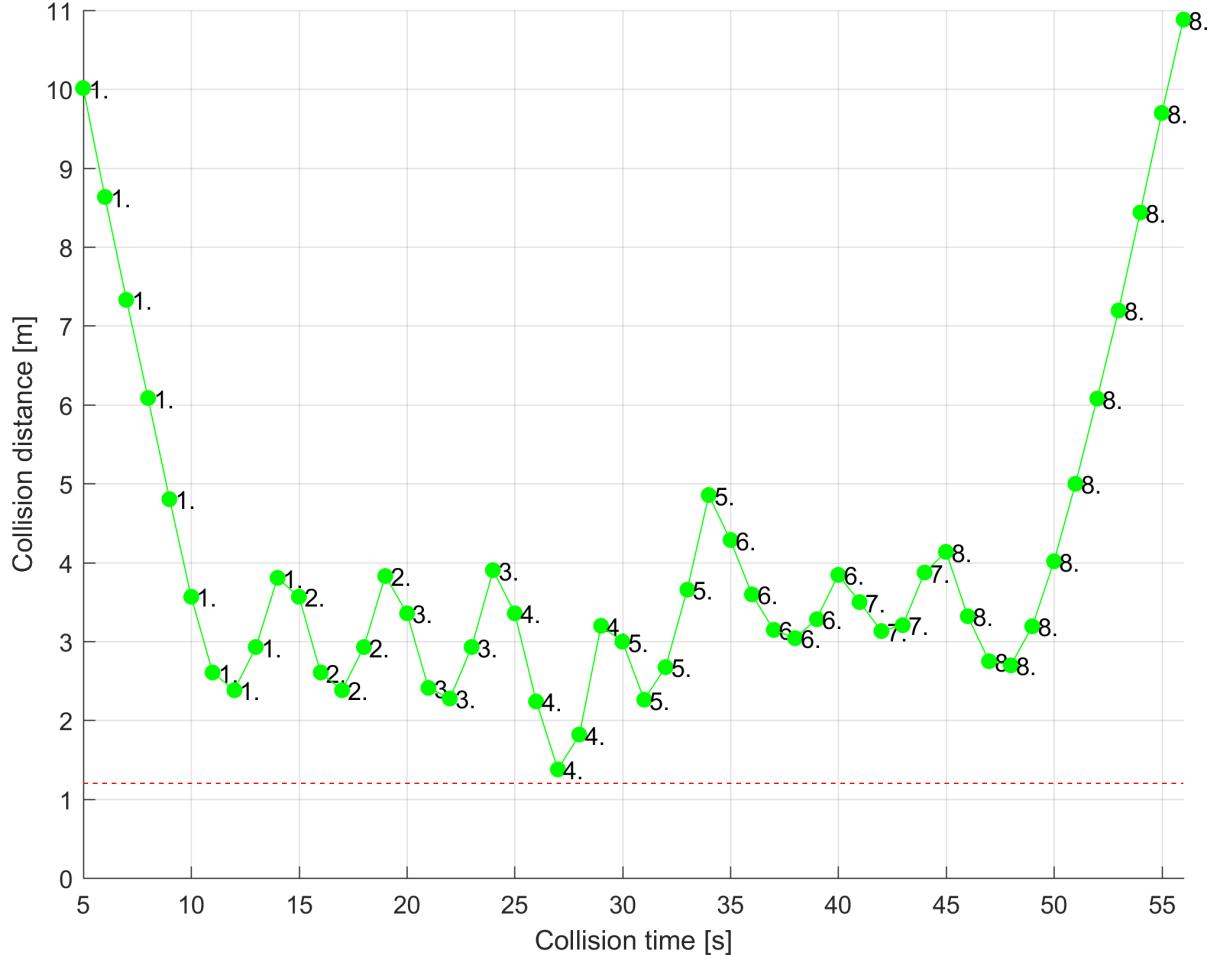


Figure 6.28: Intruder collision distance  $c_d(t)$  evolution.

*Intruder collision distance  $c_d(t)$  evolution* (6.28) shows that the necessary condition  $c_D(t) \geq s_m(1.2m)$  holds. *Green line* shows evolution over time. *Green full circles* are marking the closest intruder  $i_k \in \mathcal{I}(t)$ , defined in table 6.2. The time starts at 5s when first intruder is detected. The evolution of  $c_D(t)$  reflects how intruders were popping out and tried to collide with our vehicle. The closest intruder was  $i_4$  whom achieves distance of 136cm.

# Chapter 7

## Conclusion and Future work

**Conceptual problem formulation** is introduced in chapter 1. Where related work and goals are main focus, the goals have been fulfilled to some extent and some aspects needs additional work.

**The data fusion problem** has been formulated in 2, the three questions of *visibility, reachability, and obstacle probability* have been confronted with deterministic approach shortcomings (sec. 2.1), where problem of *LiDAR density* (fig. 2.1), *map and detected obstacles fusion* (fig. 2.2), *Intruder behaviour* (fig. 2.3), and *safety of passing trajectories* (fig. 2.4) have been opened.

**The state of art** have been discussed in chapter 3. This chapter is taken from most parts [? ]. It introduces the notion of UAV model in continuous time (3.2) and in discrete time (3.3). The key concepts of reach sets are given in section 3.2. The reach set used further in  $\mathcal{R}(\hat{x}, t_i, t_{i_1})$  (3.4), is point based reach set with restricted control. Movement automaton  $\mathcal{MA}$  concept is introduced in section 3.4, by definition 12. *Movement automaton prediction stability* is given by definition 13. *Movement automaton Lyapunov stability* is proven in section 3.4.2. The other important properties, like *controllability, robustness, observability* have been proven in [? ].

**The control framework based on reach sets** have been introduced in [? ]. The complex topology scheme and communication is omitted in conclusion, only key modules for this work will be discussed. *Obstacle avoidance* in terms of data fusion with dynamic obstacle sets  $\mathcal{O}_D$ ,  $\mathcal{O}_M$ , originating from various data sources  $\mathcal{S} = \{s_1, s_2, \dots, s_k\}, k \in \mathbb{N}$ . Movement constraints are mainly given by static uncharted obstacles and non-cooperative intruders. The main focus is on *data fusion*. Usually multiple layer control concept is used. Control concept used for this work is in fig. 7.1.

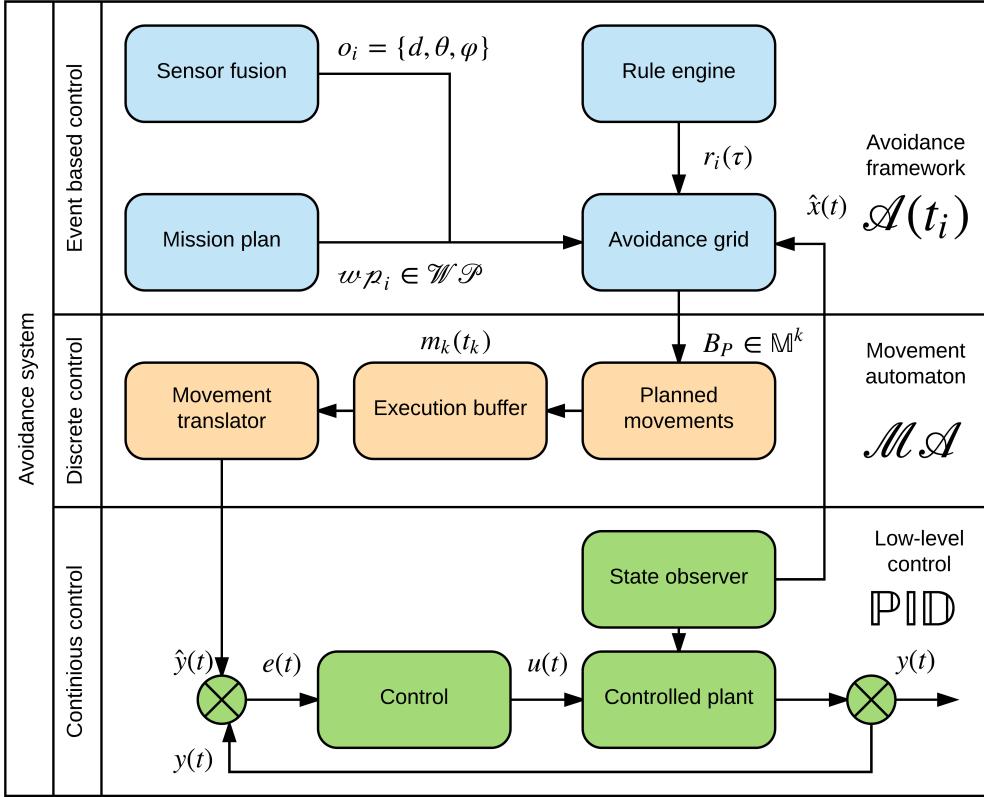


Figure 7.1: Decision frame  $\mathcal{D}(0)$ .

*Event based control* module is responsible for high level decisions and mission execution. This module generates high level input commands which are optimal to given criterion. *Event based control* consist from following artifacts:

1. *Sensor fusion* - fuses data from sensor system with observed system state  $\hat{x}$ , outputs detected obstacles  $o_i \in \mathcal{O}$ .
2. *Mission plan* - contains mission plan  $WP$ , determines navigation goal waypoint  $WP_g$ .
3. *Rule engine* - determines additional constraints for avoidance grid  $\mathcal{A}(t_i)$ , determines escape goal in avoidance grid  $c_{i,j,k}$ .
4. *Avoidance grid* - responsible for optimal path planning in partially known space, consumes observed system state  $\hat{x}(t)$ , rule engine decisions  $r_i(\tau)$ , obstacle set  $o_i \in \mathcal{O}$  and goal waypoint  $WP_i$ , produces avoidance command chain  $B_p \in \mathbb{M}^k$ .

*Discrete control* module is responsible for processing high level movement command chain  $B_p \in \mathbb{M}^k$ . It is implemented as special type of open hybrid automaton, movement automaton  $\mathcal{MA}$ .

*Continuous control* module is interfaces from *avoidance system* via the movement automaton. The same is achieved with *sensor fusion* module which is interfacing the sensor network  $\mathcal{S}$  and avoidance grid  $\mathcal{A}(t_i)$ .

**Problem of data fusion** is formulated in chapter 4, outlining important *general data fusion algorithm* proposal, section 4.4. The identified probabilities and input artifacts are summarized in eq. 4.19. The *main fusion algorithm* is proposed in eq. 4.20. The new definition of *reachability probability*  $P_R$  bounded to trajectory  $\mathcal{T}(x_0, B) \in \mathcal{R}$  have been established by definition 20, furthermore projected into cell  $c_{i,j,k} \in \mathcal{A}$  reachability probability established by definition 21.

**Reduced reach set** calculation have been introduced due the calculation complexity of probabilistic distribution over full reach set. Concept of *coverage ratio*  $C_R$  (4.27) which considers *trajectory footprint* (def. 22, 23). The *constrained expansion procedure* (sec 4.5.1) have been developed to improve estimated reach set calculation. Three new procedures to reach set  $\mathcal{R}(\hat{x}, t_i, t_{i+1})$  have been developed:

1. Chaotic reach set approximation (sec. 4.5.2).
2. Harmonic reach set approximation (sec. 4.5.3).
3. Combined reach set approximation (considered in testing) (sec .4.5.4).

**Important results** are linked to research proposal goals:

*Goal 1.: Universal interface for vehicle sensors* have been introduced in section 4.4.

*Goal 2.: Probabilistic model for avoidance grid* have been summarized in topics:

1. *Intruder intersection model* (sec. 5.1) introduces the concept of intruder intersection with avoidance grid. The notable are boundaries for cell entry (5.3) and cell leave time (5.4) which enables to take time into account when calculating probability (5.5). Concept of base intersection probability have been defined for following intersection modes:
  - a. linear intersection model (5.9),
  - b. vehicle body volume intersection model (5.13),
  - c. conic spread intersection model (5.23).

Then *general producible intersection model* (5.36) of one intruder  $i \in \mathcal{I}$  has been defined combining the previously mentioned intersection modes of base probabilities and time constraints (5.35). The intruder obstacle probability for single cell  $c_{i,j,k} \in \mathcal{A}(t_i)$  and multiple intruders  $i \in \mathcal{I}(\tau)$  has been given by (5.37).

2. *Detected obstacle probability* generic model for multiple sensors  $s_k \in \mathcal{S}$  has been given by eq. 5.38. *LiDAR hit formula* (5.45) has been created for simple homogeneous rotary LiDAR sensor.
3. *Visibility probability* for sensor network  $\mathcal{S}$  has been defined in eq. 5.47.
4. *Map obstacle probability* is defined as combination of visibility and raw map obstacle probability (5.54).

*Goal 3.: Data fusion procedure* is proposed in section 4.4.

*Goal 4.: Comparison of probabilistic and deterministic approach* have been simulated in section 6.4. Detailed comparison with results from deterministic approach [? ] are given in section 6.4.1. *Avoidance performance* is compared in section 6.4.2.

**Future work** on probabilistic approach is mainly in partition of sensor fusion. Currently the sensor fusion for LiDAR and ADS-B have been proposed, but there is more ranging sensors which are not coherence with avoidance grid like LiDAR.

*Sensor fusion priority problem:* Vehicle can be equipped with various cooperative and non-cooperative sensors  $s_1, s_2, \dots, s_4 \in \mathcal{S}$ . Which have some overlapped areas and some precision. It would be necessary to develop *prioritized sensor fusion*, complex outline of this problem in deterministic context have been given by [? ].

**Reusable components for deterministic approach** : Probabilistic approach have introduced concepts of *coverage ratio*  $C_R$  and *reduced reach set approximation*. These results can be used to reduce reach set complexity and improve scalability of deterministic approach [? ].

*The intruder intersection model* can be reused in deterministic approach, where the step function to asset the reachability of cell.