

Chapter 7

Simulations

The chapter presents the set of simulations developed according to a test plan (sec. 7.1). Test configuration (sec. 7.2) targets at exercising and evaluating proposed framework. The test cases are grouped in the following sections:

1. *Non-cooperative test cases* (sec. 7.3).
2. *Cooperative test cases* (sec. 7.4).
3. *Test cases conclusion* (sec. 7.5).
4. *Reach set approximation performance tests* (sec. 7.6).

7.1 Test Plan

The *Avoidance requirements* are given in (sec. ??), namely:

1. *Safety Margin Enforcement* (sec. ??) - keep UAS safe depending on situation.
2. *Path Tracking* (sec. ??) - track mission is given by a set of *waypoints* in the manner of *Energy Efficiency* (sec. ??).

These are given as nominal behavior (sec. ??), further enhanced by rule-based behavior (sec. ??).

The *Navigation requirements*, out of this scope, are given in (sec. ??). These are satisfied by *Mission Control Run* (sec. ??).

7.1.1 Testing approach

The purpose of this section is to show complex scenarios, not unit testing of framework functionality. The focus is on *borderline* cases for typical situations in an *expected environment*. The *mode switch* between *Navigation* and *Emergency Avoidance*.

The *Tests* are designed to focus on particular functionality in specific *operational environment* with main *obstacle/weather/intruder feature* with environment induced *constraints*. There is also *UTM* factor and *Navigation penalty*.

Operational Environment is classified according to:

1. *Operation space* - important for *Low Altitude Operations*, the difficulty of *Avoidance Maneuvers* is proportionally increasing with *Obstacle density*. There are following main categories
 - a. *Rural environment* - the relief and man-made structures are sparsely spread around the *operation space*; the UAS is operating on *very low altitude* (≤ 50 feet).
 - b. *Urban environment* - the concentration of the man-made structures are much higher, and they are more incorporated into land relief pattern, the UAS is operating on *very low altitude*.
 - c. *Open air* - the concentration of ground structures is very low, the concentration of *cooperative* and *non-cooperative intruders* is increased, the UAS is operating in altitude ranging from *50 feet* to *space border*. This brings us to:
2. *Airspace category* - when *Operation Space* pattern is categorized as *Open air* and depending on *altitude above mean sea level*. The UTM is *designed authority* for controlled airspace in current *F/G class airspace*.
 - a. *Controlled* - Open air where authority is present. The cases when *Authority* is not enforced due to the UTM malfunction, *C2 link loss* or other cause are not considered.
 - b. *Non-Controlled* - Open air operation space where is no central arbiter to determine or enforce traffic attendants behavior.

Static obstacles: Static obstacles with various features detectable by main *LiDAR* sensor. The main purpose is to show avoidance capabilities combined with heavy restrictions imposed by *soft* and *hard* constraints. The original purpose of our approach was to provide robust framework for static obstacle avoidance. Three tests with increasing obstacle density and navigation complexity are delivered.

Operational Space Constraints depends mainly on the *operational environment*. The standard set of constraints were taken into account for our test cases:

1. *Rural, Urban environment (low altitude)* are geo-fencing zones, ground (hard constraints), non-controlled airspace altitudes (soft constraints).
2. *Non-controlled airspace constraints (open air)* are geo-fencing zones (hard constraints), restricted airspace (hard constraint), weather (soft/hard constraint), controlled airspace (hard constraint), very low altitude border (soft constraints).
3. *Controlled airspace constraints (open air)* are restricted airspace (hard constraint), weather (soft/hard constraint), non-controlled airspace boundary (hard constraints), UTM Directives (hard constraints).

Air Traffic Attendants:

1. *Non-cooperative UAS* (Intruder) - there are some intruders with some degree of authority, size and *severity*. There were three test cases for non-cooperative intruders. Non-cooperative Intruders can be categorized as following based on behavior:

- a. *Chaotic* intruders usually tend to behave unpredictable, for example, bird or *UAS in distress*, for this type of intruders *Maneuver Uncertainty Intersection Model* is used (app. ??).
- b. *Harmonic* intruder usually follows long straight paths, for example, UAS converging to waypoint, for this type of intruder *Body Volume Intersection Model* is used. (app. ??).

Cooperative UAS (Intruder) - there are cooperative intruders who are obeying authority (UTM) or follow *common consensus*. The work focus on *UTM* authority implementation in four test cases. These test cases are reflecting the traffic management situations essential for successful UTM collision management

Weather impose *soft* and *hard* space constraint, which can be moving or static. The *soft constraint avoidance* is covered by *hard constraint avoidance*. The *static constrained area* is covered by *static obstacle avoidance* capability due to the *data fusion procedure* [1]. The only case which is not covered is *Moving constrained area*; small constraints can be covered by intruder models. The ideal candidate is a *storm*, because it covers quite a large area, the clouds are constantly moving, and severity is changing with time.

UTM: The *UAS Traffic Management* service should be implemented in *controlled airspace* by 2035. It is necessary to study impact of UTM services on the *Detect and Avoid* systems like ours.

The most basic service is *Identity provider* which should be implemented by 2020.

Then there are *location services*, which are necessary for coordinated collision avoidance, these were implemented in our solution up to necessary level for *Rules Of the Air* implementation.

Mission tracking is service tracking deviations from *declared mission plan* and *actual execution*. These statistics were used in all tests to track deviations from the reference trajectory.

Directives for Traffic management and *Collision prevention* are implemented as the functional life cycle of *Position notification* (sec. ??), *Collision Case* (sec. ??) for UTM. The directive handling is implemented as *Rule engine* (sec. ??) on UAS side.

Navigation: Navigation algorithm is depending on *Navigation mode*. UAS is usually in *Navigation mode* most of the time, despite this fact, UAS was forced into *Emergency Avoidance Mode* most of the time in test cases. The navigation complexity has been divided into following categories:

1. *Open space* - UAS has visibility to goal waypoint most of the time; there are no traps.
2. *Hidden waypoint* - UAS does not have visibility to goal waypoint, most of the time; there are irregular traps sometimes.
3. *Maze solving* - UAS line of sight for goal waypoint is hindered by multiple obstacles, there are irregular traps often.
4. *Rule following* - UAS navigation capabilities are constrained by rule enforcement.

7.1.2 Test Cases Summary

Test cases are summarized in (tab. 7.1).

Test Case Name	Operational Environment	Air Traffic Attendants	Weather	UTM	Navigation	Scenario
Building Avoidance	Non-controlled (Rural) $4 \times buildings$	-	-	-	Open space	Fly mission around four buildings
Slalom	Non-controlled (Rural) $14 \times buildings$	-	-	-	Hidden waypoint	Navigate to hidden waypoint
Maze	Non-controlled (Urban) $30 \times buildings$	-	-	-	Maze structure	Solve maze with multiple curves
Storm	Non-controlled (Rural) $0 \times buildings$	-	Storm	-	Open Space	Avoid approaching storm
Emergency Converging	Non-controlled (Open air)	Non-cooperative UAS (1x)	-	-	Open Space	Converging situation resolution w. o. UTM
Emergency Head on	Non-controlled (Open air)	Non-cooperative UAS (1x)	-	-	Open Space	Head on situation resolution w. o. UTM
Emergency Multiple	Non-controlled (Open air)	Non-cooperative UAS (3x)	-	-	Open Space	Multi-collision case resolution w. o. UTM
Rule-based Converging	Controlled (Open air)	Cooperative UAS(1x)	-	Full	Follow Rules	Converging situation resolution with UTM
Rule-based Head on	Controlled (Open air)	Cooperative UAS(1x)	-	Full	Follow Rules	Head on situation resolution with UTM
Rule-based Multiple	Controlled (Open air)	Cooperative UAS(3x)	-	Full	Follow Rules	Multi-collision case resolution with UTM
Rule-based Overtake	Controlled (Open air)	Cooperative UAS (1x)	-	Full	Follow Rules	Overtake by UAS different speed ratio

Table 7.1: Test Cases Summary.

7.1.3 Performance Evaluation

Evaluation method: *Test cases* were evaluated according to performance requirements defined in (sec. ??). The method was tracking critical parameter for *Safety* (sec. ??) (primary) and *Trajectory Tracking* (sec. ??) (secondary) including *Energy Efficiency* (sec. ??).

Safety Margin Performance Evaluation: The *safety of UAS* is main concern of *DAA system*. The common concept of *safety margin* is evaluated.

The *threat* is multidimensional; there are often multiple *static obstacles*, *intruders* or *weather constraints*. To reduce the multidimensional threats to one-dimensional value *crash distance* concept is used:

$$\text{crashDistance}(t) = \text{distance}(\text{UAScenter}(t), \text{threat})$$

where *selection the criterion* is:

$$\min \left\{ \begin{array}{c} \left(\text{distance}(\text{UAScenter}(t), \text{threat}) - \dots \right) \\ \dots - \text{threat.SafetyMargin} \\ : \forall \text{threat} \in \text{KnownWorld}(t) \end{array} \right\} \quad (7.1)$$

The *crash distance* (eq. 7.1) for given time is evaluated as shortest distance between UAS center and threat. The threat origins from the known world (sec. ??). The *threat* has safety margin. The distance to safety margin is used as a prioritization criterion in our test cases (tab. 7.1).

The *safety margin* evolution over time (eq. 7.2) is calculated similarly to *crash distance*. The most dangerous threat is selected based on *distance to the safety margin* criterion. The value of *safety margin* property is then used.

$$\text{safetyMargin}(t) = \text{threat.SafetyMargin}$$

where *the selection criterion* is:

$$\min \left\{ \begin{array}{c} \left(\text{distance}(\text{UAScenter}(t), \text{threat}) - \dots \right) \\ \dots - \text{threat.SafetyMargin} \\ : \forall \text{threat} \in \text{KnownWorld}(t) \end{array} \right\} \quad (7.2)$$

The *distance to safety margin* (eq. 7.3) is calculated as a difference between the *crash distance* (eq. 7.1) and *safety margin* (eq. 7.2). The *acceptance criteria* for safety is the *distance to safety margin* ≥ 0 .

$$\text{distanceToSafetyMargin}(t) = \text{crashDistance}(t) - \text{safetyMargin}(t) \geq 0 \quad (7.3)$$

Note. On Signed Distance: The most works are using *unsigned distance*. This work considers the *signed distance* with the following intervals:

1. + (away from the safety margin).
2. 0 (touching margin with UAS edge).
3. - (inside margin - crash/collision/broken boundary).

Distance to Safety Margin peaks are measured:

1. *Minimal* distance to safety margin indicates if *acceptance criterion* (eq. 7.3 is met).
2. *Maximal* distance to safety margin indicates the future *minimal detection range*. All scenarios were considered as borderline cases.

Trajectory Tracking Evaluation is a secondary priority after safety, following parameters were checked:

1. *Waypoint reach* - the *Mission* (??) is considered as completed if and only if \forall waypoints are reached and in the given order (check the output of ??). Moreover, if there is multiple UAS, each must meet the condition.
2. *Acceptable deviation* - for *tracking problem* (eq. ??) is a trajectory which in addition to *basic obstacle problem* (sec. ??) keeps deviation from the *reference trajectory* under a certain threshold (eq. ??).

Trajectory tracking deviation threshold (eq. 7.4) is defined as double of maximal distance between *goal waypoint* and *previous waypoint*.

$$\text{trackingDeviationThreshold} = 2 \times \text{distance}(\text{goalWaypoint}, \text{previousWaypoint}) \quad (7.4)$$

Note. If *goal waypoint* is first in the *mission*, the *UAS initial condition* is considered as a *previous waypoint*.

Computation Load: There is a theoretical definition of *intersection models* for *static obstacles and constraints* (sec. ??), *moving obstacles and constraints* (sec. ??), *avoidance run* (sec. ??), *mission control* (sec. ??) computational complexity.

The practical application requires to measure *computation load* in constrained environment. Let say that *avoidance framework* is running on stand alone embedded computer with 1.2 GHz processor and 1GB of dedicated RAM. This is simulated by *virtual machine*.

The *simulations* were executed in *Matlab/Simulink* environment¹ using: *UTM*², *Navigation loop*³, *Avoidance grid*⁴ and *Reach set*⁵ implementations.

The *decision frame* length is set to 1s which gives *computation load* (eq. 7.5). The *computation load* represents the portion of the *previous decision frame* used to current decision frame calculation.

$$\text{computationLoad} = \frac{\text{computationTime(frame)}}{\text{decisionFrameDuration}} \times 100, \quad [\%; s, s] \quad (7.5)$$

Note. *Computation load* is depending on the actual situation; when the UAS is in *navigation mode*, it should be low, when the UAS is in a *clustered environment* it should be high.

Matlab implementation is quite ineffective; the Python/C++ implementation can give better results.

For *computational feasibility* there is *implicit acceptance criterion* (eq. 7.6): the computation of a feasible path for *this time-frame* must end in the *previous time-frame*.

$$\forall \text{time} \in \text{Mission} : \quad \text{computationLoad} < 100\% \quad (7.6)$$

¹Prototype framework implementation: <https://github.com/logomo/Feature-based-ACAS/>

²UTM class: .../UavTraficManagement/UTMControl.m

³Navigation Loop main class: .../MissionControl/MissionControl.m

⁴Avoidance Grid class: .../AvoidanceGrid/AvoidanceGrid.m

⁵Reach set tree class: .../AvoidanceGrid/PredictorNode.m

7.2 Testing Configuration

All *simulations* are run with the configuration described in this *section*. The UAS used for the purposes is given by *model and control* (sec. ??).

UAS parameters: An *UAS system* (tab. 7.3) is modeled after small scale toy model with maximal body radius 30 cm, maximal speed 4 m.s⁻¹, weight 450 g., maximal flight duration 20 min, maximal turning rate 15 deg.s⁻¹. The *body margin* is set to 0.3m; the *near-miss radius* is double of *body margin*; thus 0.6 m, the *well clear radius* is set to 5 m. Margins can be set to any value if they are complaint with condition (7.7).

$$0 < \text{bodyMargin} \leq \text{nearMissRadius} \leq \text{wellClearRadius} \leq \text{gridDistance} \quad (7.7)$$

Note. The *safety margin* is broad term used to describe the *minimal distance* between UAS and *adversarial object*. The *Safety margin* is:

1. *Near miss radius* in case of *non-controlled airspace* or *emergency avoidance mode*.
2. *Well clear radius* in case of *controlled airspace* and *navigation mode*.

Decision time: Decision time can be set by the user to any positive non-zero value (7.8). The *Decision time* is equal 1 s, and *Decision frames* are synchronized.

$$\text{maxAlgorithmCalculationTime} \leq \text{decisionTome} \leq \infty \quad (7.8)$$

Speed: For *all movements* constant speed 1 m.s⁻¹ is used. Speed can be changed to any value in the given boundary (7.9).

$$0 \leq \text{speed} \leq \min \left(\begin{array}{l} 0.5 \times (\text{navigationGrid.distance}/\text{decisionFrame}) \\ 0.5 \times (\text{avoidanceGrid.distance}/\text{decisionFrame}) \end{array} \right) \quad (7.9)$$

Movement automaton: The *movement set* is given in (tab. 7.2). The *movement set* contains horizontal, vertical, and, combined movements.

Grids: Used *Navigation grid parameters* are given in (tab. 7.4). Selected *Navigation Reach set* is *ACAS-like* with enabled horizontal/vertical separation. Used *Avoidance grid parameters* are given in (tab. 7.5). Selected *Avoidance Reach set* is *combined* because of high *coverage ratio*.

The user can define own grid parameters according to the *space discretization rules* (sec. ??) and chose own *reach set type* according to preference (sec. ??).

Movement	Roll	Pitch	Yaw
Straight	0°	0°	0°
Left	0°	15°	0°
Right	0°	-15°	0°
Up	0°	0°	-15°
Down	0°	0°	15°
UpLeft	0°	15°	-15°
UpRight	0°	-15°	-15°
DownLeft	0°	15°	15°
DownRight	0°	-15°	15°

Table 7.2: Movement orientations.

UAS parameters	
speed	1 $m s^{-1}$
horizontal turning r.	3.82 m
vertical turning r.	3.82 m
body radius	0.3 m
near miss r.	0.6 m
well clear r.	5 m

Table 7.3: UAS parameters.

Navigation Grid	
RSA type	ACAS-like
distance range	0 – 10 m
layer step	1 m
horizontal range	±45°
horizontal cells	7
vertical range	±30°
vertical cells	5

Table 7.4: Navigation Space parameters.

Avoidance Grid	
RSA type	combined
distance range	0 – 10 m
layer step	1 m
horizontal range	±45°
horizontal cells	7
vertical range	±30°
vertical cells	5

Table 7.5: Avoidance Space parameters.

Coloring		
Airc.	Executed	Planned
UAS 1	blue	red
UAS 2	cyan	magenta
UAS 3	green	yellow
UAS 4	black	green

Table 7.6: UAS coloring.

7.3 Non-cooperative test cases

The *main* goal of this section is to show operative capabilities for *non-cooperative* avoidance mode in *emergency* and *solo situations*.

Test avoidance capabilities against *static obstacles*, *non-cooperative intruders*, *moving hard constraints* are covered.

Coverage of the *soft constraints*, *map obstacles*, and *detected obstacles* are implicitly covered due to the properties of *safety* and *body margins* (tab. ??).

1. *Building Avoidance* (sec. 7.3.1) covers *static obstacles* explicitly and *map obstacles*, *hard constraints*, *ground avoidance* implicitly.
2. *Slalom* (sec. 7.3.2) covers *open space navigation capabilities*, showing the determinism of the *avoidance loop run*, in addition to *building avoidance*.
3. *Maze* (sec. 7.3.3) covers *closed space navigation capabilities*, showing the higher level navigation properties of primitive *right-side* 2D maze solver. The main point is to show the possibility to enrich the *Navigation loop algorithm* (fig. ??).
4. *Storm* (sec. 7.3.4) covers *hard moving constraints avoidance* explicitly and *hard static constraints*, *soft static constraints*, *soft moving constraints* implicitly.
5. *Emergency converging scenario* (sec. 7.3.5) covers *non-cooperative intruder with the right of way* avoidance capability.
6. *Emergency head-on scenario* covers (sec. 7.3.6) *non-cooperative intruder without right of way* avoidance capability
7. *Emergency mixed scenario* (sec. 7.3.7) covers *multiple intruders with/without right of the way* avoidance capability.

7.3.1 Building avoidance

Scenario: The *UAS* is flying the mission given by (tab. 7.7) in the *open space environment*. There exists a map of obstacles with defined *safety and body margins*. *Reference trajectory* (direct interconnection of waypoints) is going through partially known space with some charted obstacles.

Position		Waypoints			
$[x, y, z]$	$[\theta, \varpi, \psi]$	\mathcal{WP}_1	\mathcal{WP}_2	\mathcal{WP}_3	\mathcal{WP}_4
$[0, 0, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[100, 0, 0]^T$	$[100, 100, 0]^T$	$[0, 100, 0]^T$	$[0, 0, 0]^T$

Table 7.7: Mission setup for *Building avoidance* scenario.

Obstacle set: Obstacles are discovered during a flight by *UAS LiDAR sensor*, the set of obstacles is defined in (tab. 7.8).

Obstacle			Body Margin			Safety Margin
id	position	type	min.	max.	avg.	
1	$[50, 0, 0]^T$	polygonal	14	20	16	5
2	$[100, 50, 0]^T$	hospital	12	18	14	7
3	$[50, 100, 0]^T$	unusual	10	20	15	8
4	$[0, 50, 0]^T$	square	18	20	19	4

Table 7.8: *Obstacle set for Building avoidance scenario.*

Main Goal: Show *static obstacle avoidance capability* in an *open space environment*, using *LiDAR scanning* and *obstacle map* as the *information sources*.

Acceptance criteria:

1. Proper *algorithm mode switch*:
 - a *Avoidance mode* is active when the *UAS* is nearby to the obstacle (*distance* (*obstacleCenter*, *UASPosition*) $\leq 20m$).
 - b *Navigation mode* is active when the *UAS* is further away from any obstacle (*UAS* is actively converging to *goal waypoint*).
2. *Minimal safety margin distance* $\geq 0m$
3. *Reach each waypoint* (tab. 7.7) in the given order.

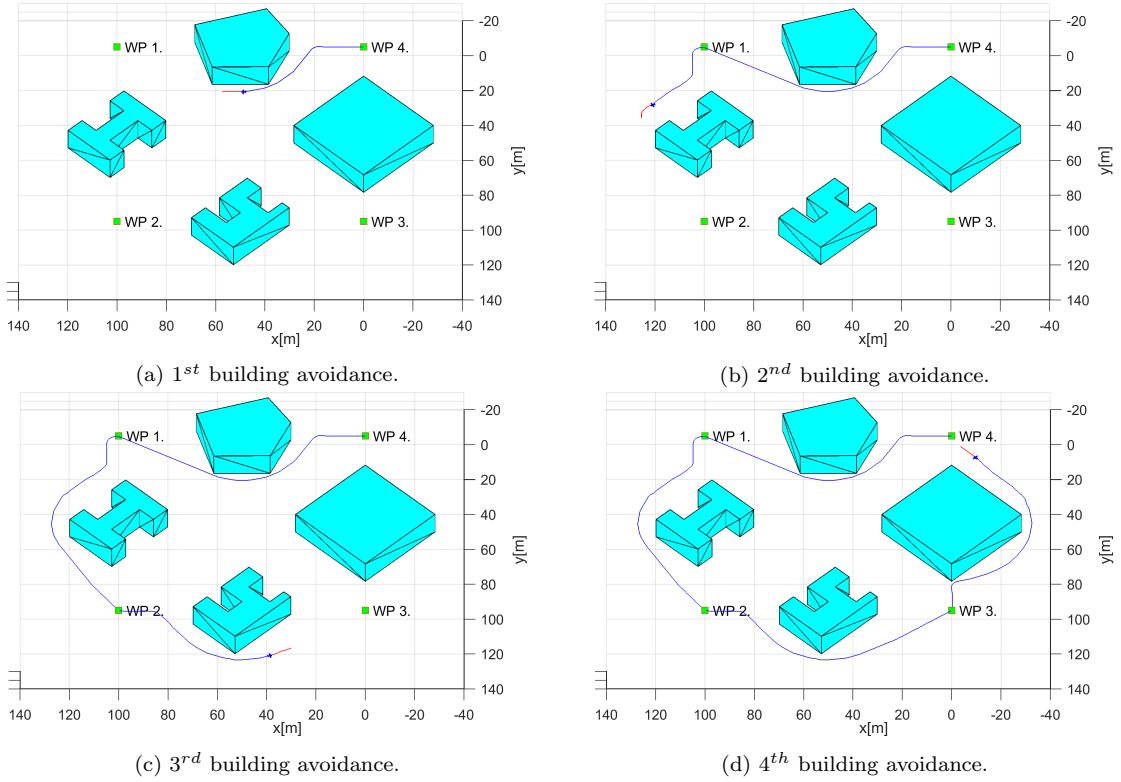
Testing Setup: The *standard test setup* defined in (tab. 7.2, 7.3, 7.4, 7.5, 7.6) is used with following parameter override:

1. *Avoidance grid - type - ACAS-like* with *horizontal enabled maneuvers*

Note. Enforced *safety margin* does not exceed the *avoidance grid range* (10 m). The concept of *Static obstacle avoidance* is in detail discussed in the *progress report* [1].

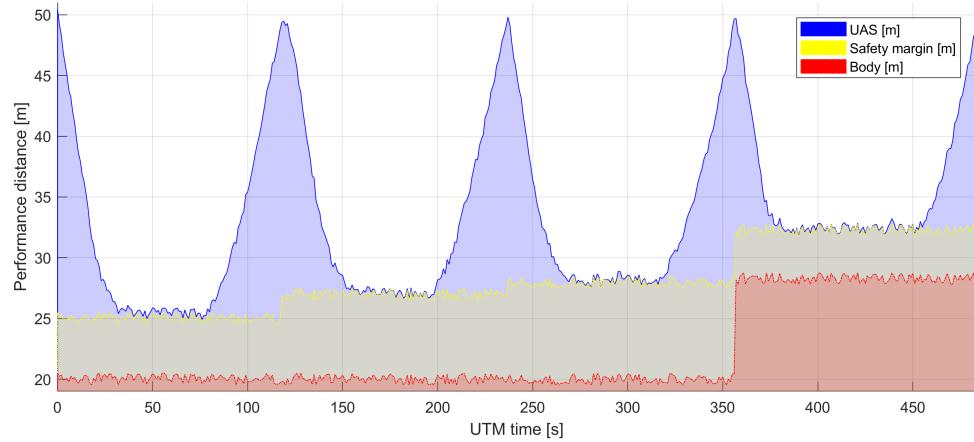
Simulation Run: Notable moments from the *simulation run* (fig. 7.1) are the following:

1. *1st building avoidance.* (fig. 7.1a) - UAS avoids the building from the left side because overall trajectory cost is cheaper. The first building is a convex obstacle.
2. *2nd building avoidance.* (fig. 7.1b) - UAS avoids the building from the right side while avoiding an active non-convex portion of the building.
3. *3rd building avoidance.* (fig. 7.1c) - UAS avoids the building from the right side, missing both traps from it.
4. *4th building avoidance.* (fig. 7.1d) - UAS avoids the building from the right side. This building is also a convex obstacle.

Figure 7.1: Test scenario for *Building avoidance* (static ground obstacles).

Distance to Body/Safety Margin Evolution: The distance of *UAS* center to the nearest obstacle (blue) does not break a *safety margin* (of the closest obstacle (yellow)) nor *body margin* of the closest obstacle (red) as it can be seen in (fig. 7.2). *Acceptance condition for algorithm mode switch* can be shown by *UAS active avoidance of obstacles*.

Note. The *body* and *safety margins* are changing depending on *UAS position and orientation*, is changing reflecting (tab. 7.8) margins.

Figure 7.2: Distance to body/safety margin evolution for *Building avoidance scenario*.

Distance to Body/Safety Margin Peaks: Minimal distance to *safety margin* is 0.69 m. The *minimal distance to obstacle body* is 4.69 m which is more than sufficient for tested UAS type. *Safety margin acceptance criteria* have been achieved because the minimal distance is greater than zero. The minimal *body margin distance* is 4.69 m for obstacle no. 4 (tab. 7.8).

Parameter	UAS 1	
Distance to Safety Margin	min	0.69
	max	24.98
Distance to Body Margin	min	4.69
	max	29.98

Table 7.9: Distance to Body/Safety Margin Peaks for *Building avoidance scenario*.

Path Tracking Performance: Reference path (green dashed line) is given as direct interconnection between waypoints (green numbered square). The real trajectory (solid blue line) is split into its XYZ components. *All mission waypoints* (fig. 7.3) have been reached in the given order. There are some deviations on $X - Y$ horizontal axes, while the UAS was in the *avoidance mode*.

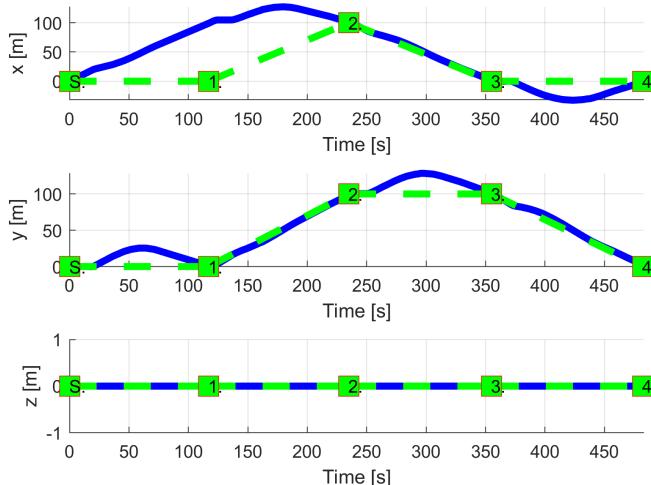


Figure 7.3: *Building avoidance path tracking*.

Path Tracking Deviations: Deviations (tab. 7.10) from the *reference trajectory* are in expected ranges considering the *mission plan* (tab. 7.7) and *obstacle properties* (tab. 7.8).

Param.	UAS 1			
	WP ₁	WP	WP ₃	WP ₄
max x	104	86	5.34	32.52
max y	25.39	6.59	28.2	4.55
max z	0	0	0	0
max dist.	107.05	86.2	28.7	32.84

Table 7.10: Path tracking for properties *Building avoidance*.

Computation Load: The *computation load* for scenario (fig.7.4) shows used time (y-axis) over decision frame (x-axis).

There is a slight increase in *computation time* when UAS is in *Emergency Avoidance Mode*.

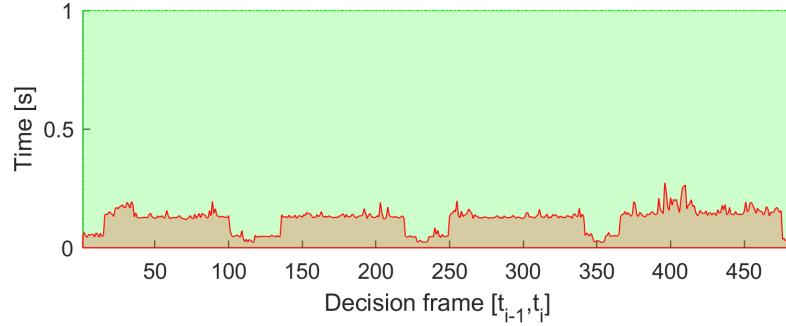


Figure 7.4: Computation time for *Building avoidance* scenario.

7.3.2 Slalom

Scenario: The *UAS* is flying the mission given by (tab. 7.11) in the *open-space environment*. An Operational space is more clustered than in the case of *Building Avoidance* (sec. 7.3.1). This map of notable *buildings* with defined *safety and body margins* imposing additional flight constraints. The *UAS* is flying through partially known space with some charted obstacles.

The *goal waypoint* is hidden behind the sensors line of sight. There are multiple cost equivalent trajectories to reach the goal.

Position		\mathcal{WP}_1
$[x, y, z]$	$[\theta, \varpi, \psi]$	
$[25, 5, 0]^T$	$[0^\circ, 0^\circ, 90^\circ]^T$	$[35, 75, 0]^T$

Table 7.11: Mission setup for *Slalom* scenario.

Obstacle set: Obstacles are discovered during a flight by *UAS LiDAR Sensor*. The set of obstacles is defined in (tab. 7.12) Some obstacles does not have *Line of Sight* during a flight, which causes additional constraints during the *avoidance trajectory selection* process.

Obstacle		Body Margin			Safety Margin
position	type	min.	max.	avg.	
multiple (4)	hospital	$[0.5, 1]$	$[2.2, 3.1]$	$[1.5, 3]$	$[1, 3]$
multiple (7)	unusual	$[0.3, 1]$	$[2.3, 3.5]$	$[2, 3]$	$[1, 4]$
multiple (3)	square	$[3, 4]$	$[4, 5]$	$[4, 5]$	$[1, 4]$

Table 7.12: *Obstacle set* for *Slalom* scenario.

Main goal: Show *static obstacle avoidance* in a *clustered environment* with *shorter decision frames* due to the obstacle density. Show *hidden waypoint navigation capability* and *Behind Line of Sight* impact on decision making.

Acceptance Criteria are given as follow:

1. *Hidden waypoint reach* - the UAS will safely reach *goal waypoint*.
2. *Minimal safety margin distance* ≥ 0 .
3. *Hindered space* is accounted into decision making (BLOS impact).

Testing setup: The *standard test setup* defined in (tab. 7.2. 7.3. 7.4. 7.5. 7.6) is used with following parameter override:

1. *Avoidance grid - type - ACAS-like with horizontal enabled maneuvers*

Note. The *vertical separation* was disabled because *UAS* will increase its altitude to reach *goal waypoint*.

Simulation run: Notable moments from this *simulation run* (fig. 7.5) are the following:

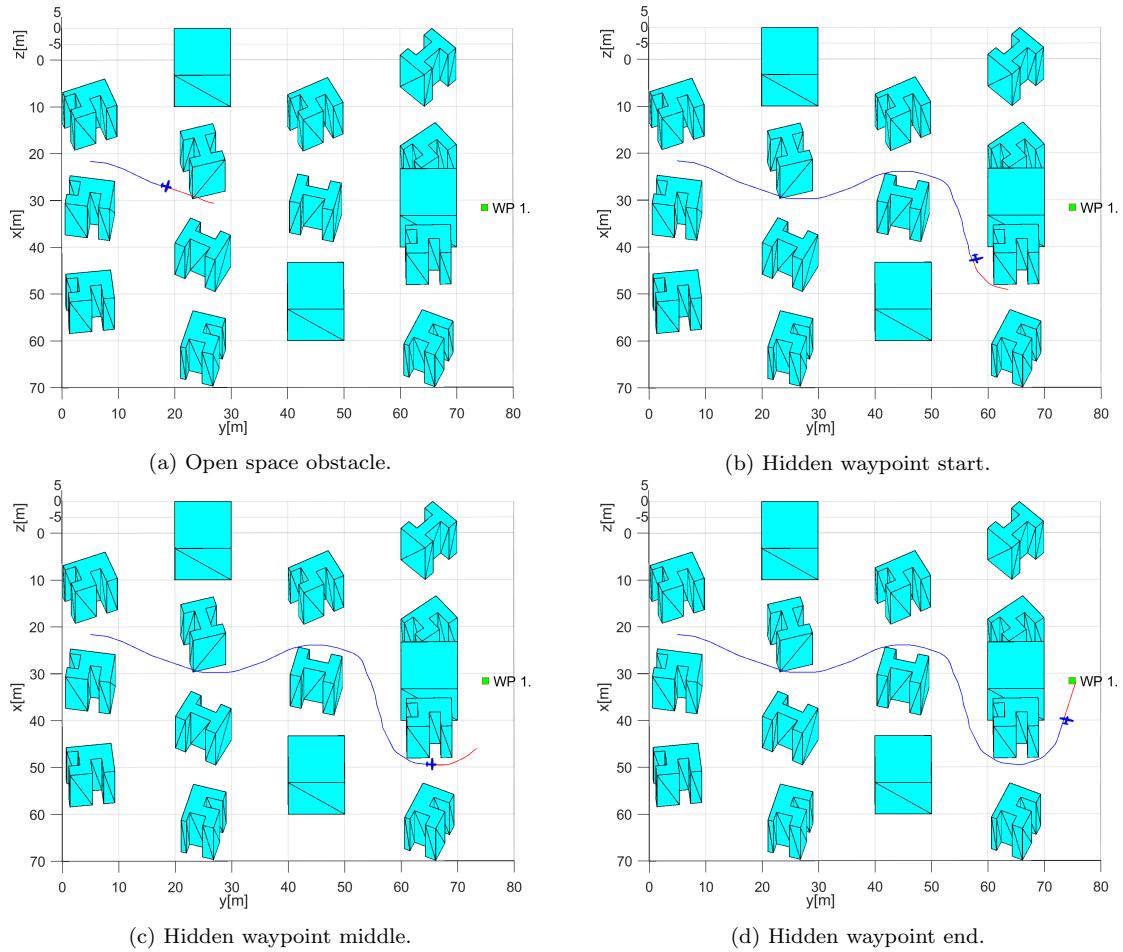


Figure 7.5: Test scenario for *Slalom* with a *hidden waypoint*.

1. *Open space obstacle* (fig. 7.5a) - avoidance of open space obstacle, while tracking *hidden waypoint*. This is standard navigation procedure, the middle building in front of *goal waypoint* is hidden by building in front of UAS.
2. *Hidden waypoint navigation* is shown in three stages start (fig. 7.5b), middle (fig. 7.5c), and end phase (fig. 7.5d). The *hidden goal waypoint* has been reached, and first acceptance

criteria were fulfilled. The *Decision points* of the navigation loop are placed in very high density around this area. The avoided building had following traps which were avoided:

- Trap (fig. 7.5b) on the left side of *UAS* was avoided because there was no turning point inside of space.
- Trap (fig. 7.5c) on the left side of *UAS* was avoided because it was not wide enough to be considered as trajectory space.

Distance to Body/Safety Margin Evolution: The *UAS* (blue fill) does not break a *safety margin* (yellow fill) nor *body margin* (red fill) as you can see in (fig. 7.6). Hindered space is accounted into decision making because the distance to closest obstacle will never breach *safety margin* (yellow fill). If it was not, the *UAS* would break *safety* or *body* margin.

Body and *Safety margin* is changing values depending on the *nearest obstacle* and *mutual position of obstacle and UAS*. The ranges of *body* and *safety margins* are reflected in (tab. 7.12).

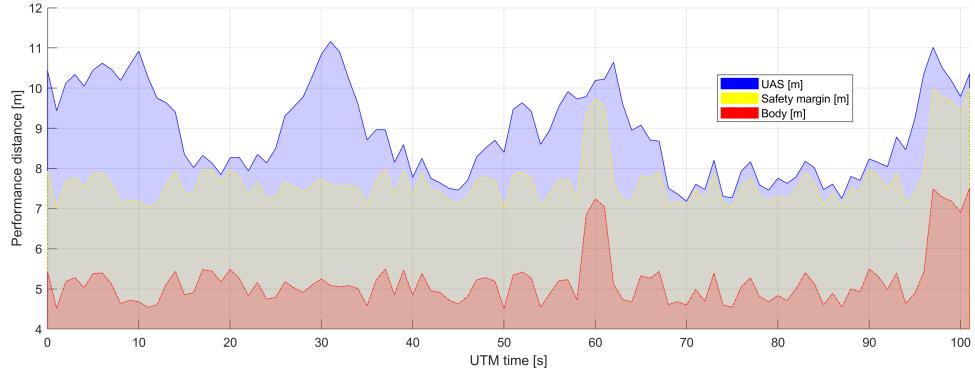


Figure 7.6: Distance to body/safety margin evolution for *Slalom scenario*.

Distance to Body/Safety Margin Peaks: The *UAS* distance to the boundary of *safety* and *body* margin is given in (tab. 7.13). The minimal distance of *UAS border*(blue line) to *safety margin boundary* (yellow line fig. 7.6) is 0.0856 m which can be considered as marginal 0. The minimal *body margin* distance is 2.5856 m; it takes into account *safety margin* of 2.5 m. The condition $safetyMarginDistance \geq 0$ holds.

The difference between minimal and maximal *safety margin distance* is ~ 3 m which indicate that the mission environment is tightly packed with obstacles.

Parameter	UAS 1	
Distance to Safety Margin	min	0.0856
	max	3.7391
Distance to Body Margin	min	2.5856
	max	6.2391

Table 7.13: Distance to body/safety margin peaks for *Slalom scenario*.

Path tracking performance: Path tracking is given in (fig. 7.7). The line between a Starting position (green square, marked S) and goal waypoint (green square marked 1) is reference trajectory (green dashed line). The flown trajectory (blue solid line) is showing evolution over mission time (Time [s]) in global coordinate frame split into three axes (x[m], y[m],

$z[m]$). The UAS was all time in *Emergency Avoidance Mode* due to the vicinity of dangerous obstacles.

The *UAS* reached final navigation waypoint, which fulfills acceptance criteria. The UAS has taken a significant detour ($x[m]$ evolution) due to the hidden *waypoint*.

The test has been run multiple times to check if *Right-Up* preference for avoidance is always selected. *Small noise* (0.5-1m) was added to obstacle positions. The algorithm always chose a similar deterministic path. The higher noise levels were not possible due to the obstacle original size (tab. 7.12).

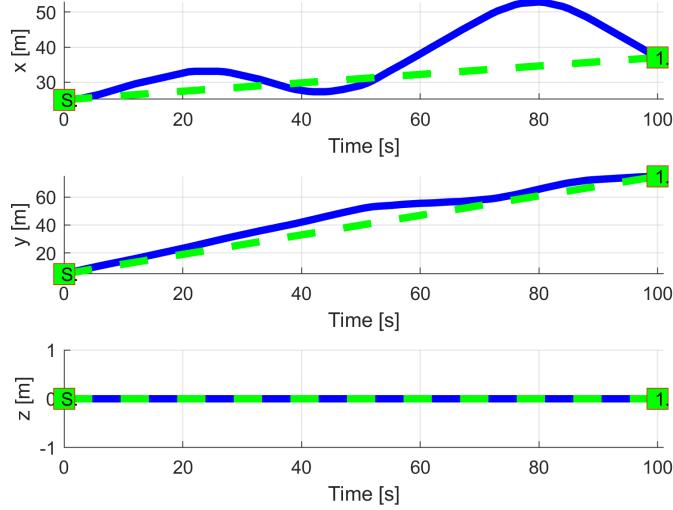


Figure 7.7: *Slalom* path tracking.

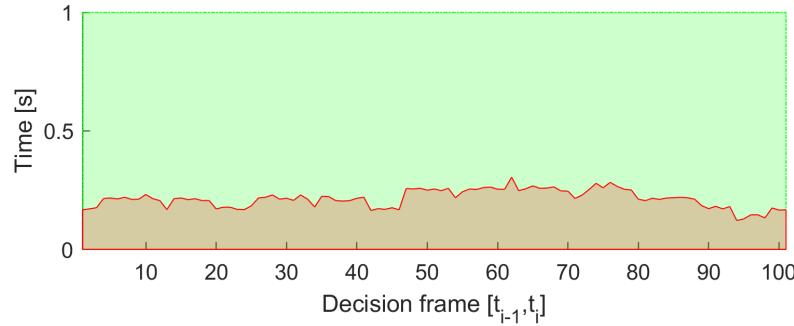
Path Tracking Deviations: Deviations given in (tab. 7.14) from *reference trajectory* (fig. 7.7) are in expected ranges considering the *mission plan* (tab. 7.11) and *obstacle properties* (7.12).

Param.	UAS 1
	\mathcal{WP}_1
$\max x $	17.90
$\max y $	12.41
$\max z $	0
$\max dist.$	20.06

Table 7.14: Path tracking properties for *Slalom* scenario.

Computation Load: The *computation load* for scenario (fig.7.8) shows used time (y-axis) over decision frame (x-axis).

The UAS is moving over *semi-clustered* environment the *computation load* is almost constant.

Figure 7.8: Computation time for *Slalom* scenario.

7.3.3 Maze

Scenario: The UAS is flying a mission given by (tab. 7.15) in *closed space* constrained by ground from the bottom, airspace constraint from top and building from sides. The maneuverable space is *maze-like* with *hidden goal waypoint*.

There exists an *Obstacle map* with defined *safety* and *body margins*. *Reference trajectories* (direct interconnection of the initial position and *goal waypoint*) is going through *partially known space* with some charted obstacles.

Position		\mathcal{WP}_1
$[x, y, z]$	$[\theta, \varpi, \psi]$	
$[15, 15, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[15, 75, 0]^T$

Table 7.15: Mission setup for *Maze* scenario.

Obstacle set: *Obstacles* are discovered during a flight by *UAS LiDAR* sensor. The *Obstacle set* is defined in (tab. 7.16). The obstacles are placed in a *virtual grid* with *cell size* $10 \times 10m$. There are following obstacles:

1. $5 \times$ *Hospital building* - H-shaped, with two open traps, with minimal body margin in the range $0.5 - 1m$, with maximal body margin in the range $2.2 - 3.1m$ and variable *safety margin* in the range $1 - 3m$.
2. $12 \times$ *Unusual trap building* - square-shaped building with two traps on the neighbouring side, with minimal body margin in the range $0.3 - 1m$, with maximal body margin in the range $2.3 - 3.5m$ and variable *safety margin* in the range $1 - 4m$.
3. $6 \times$ *Square building* - square-shaped building with minimal body margin in the range $3 - 4m$, with maximal body margin in the range $4 - 5m$ and variable *safety margin* in the range $1 - 4m$.
4. $7 \times$ *U-shaped Trap* - thin walled U shaped trap designed to catch incoming flying objects, with minimal body margin in the range $2 - 4m$, maximal body margin in the range $3 - 5m$ and various *safety margin* in the range $1 - 2m$.

The purpose of these *Obstacles* except *Square building* type is to create false positive path diversions. These diversions are designed to take *UAS* into an unsolvable situation. *Avoidance*

of traps is possible due *Reach set properties* because many scenarios for avoidance can be evaluated at once.

Obstacle		Body Margin			Safety Margin
position	type	min.	max.	avg.	
multiple (5)	hospital	[0.5, 1]	[2.2, 3.1]	[1.5, 3]	[1, 3]
multiple (12)	unusual	[0.3, 1]	[2.3, 3.5]	[2, 3]	[1, 4]
multiple (6)	square	[3, 4]	[4, 5]	[4, 5]	[1, 4]
multiple (7)	trap	[2, 4]	[3, 5]	[2, 4]	[1, 2]

Table 7.16: *Obstacle set* for *Maze* scenario.

Main Goal: Demonstrate static obstacle avoidance in closed space navigation. Focus on determinism of *avoidance run*. Demonstrate the possibilities of primitives *right-hand* maze solver incorporated into *Navigation-loop*.

Acceptance Criteria:

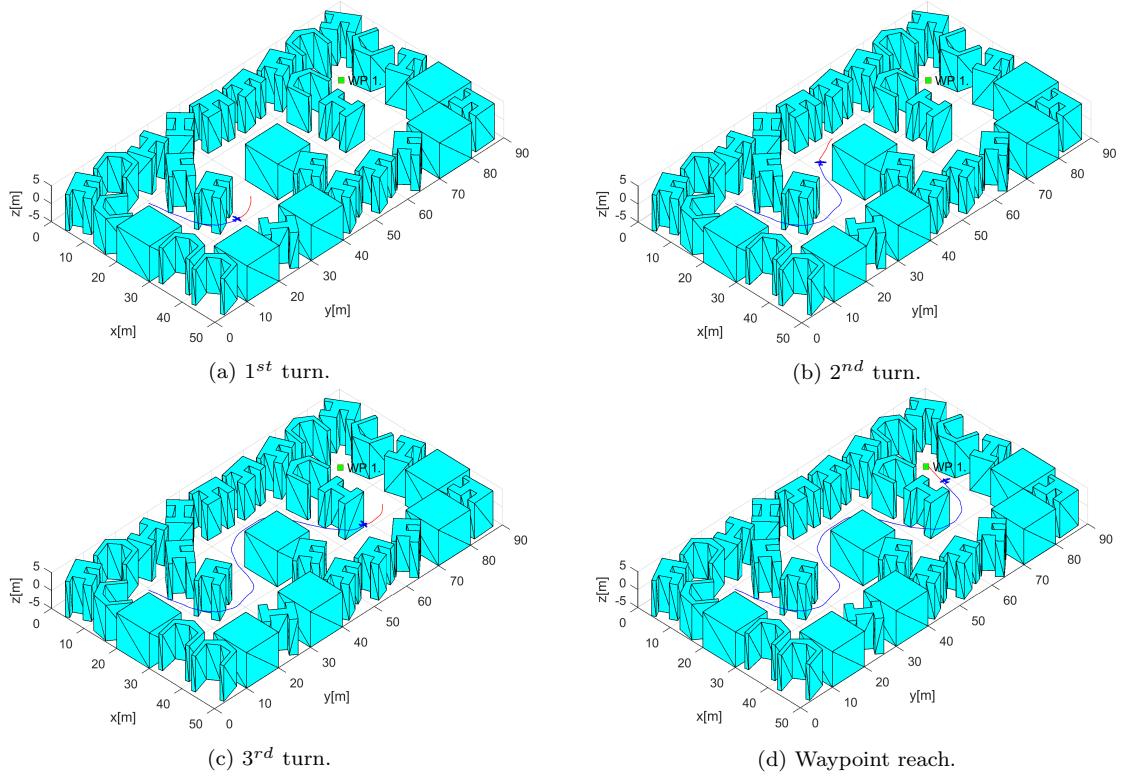
1. *Do not break top/bottom boundaries* - the UAS Z coordinate should not leave range -5 to $5m$. The boundary break occurs when there is no feasible horizontal path, and UAS needs to climb up to resolve the situation.
2. Minimal safety margin distance $\geq 0m$.
3. *Reach hidden goal waypoint* by solving simple maze (tab. 7.15).

Testing Setup: The *standard test setup* defined in (tab. 7.2, 7.3, 7.4, 7.5, 7.6) is used with following parameter override:

1. *Avoidance grid - type - ACAS-like* with *horizontal enabled maneuvers*

Simulation Run: Notable moments from the simulation run (fig. 7.9) are the following:

1. *The Maze* consists from heavy constrained turns: 1st turn (fig. 7.9a), 2nd turn (fig. 7.9b), and 3rd turn (fig. 7.9c). The hidden waypoint reach is given by (fig. 7.9d).
2. UAS is constantly in *Emergency Avoidance mode* because there is always a presence of an obstacle.
3. *The Navigation path* is located in a slim corridor with width only 3-6 meters. Mutual distance of obstacles is 20 meters, and combined margins take 14-17 meters.
4. *Maze scenario* was very close to the urban environment concerning obstacle density and computational complexity.
5. *Avoidance run* computational complexity scaled linearly with a count of active obstacles in Field of View.
6. *Hidden Goal Waypoint* has been reached as shown in (fig. 7.9d). This satisfy *reach hidden waypoint* acceptance criterion.

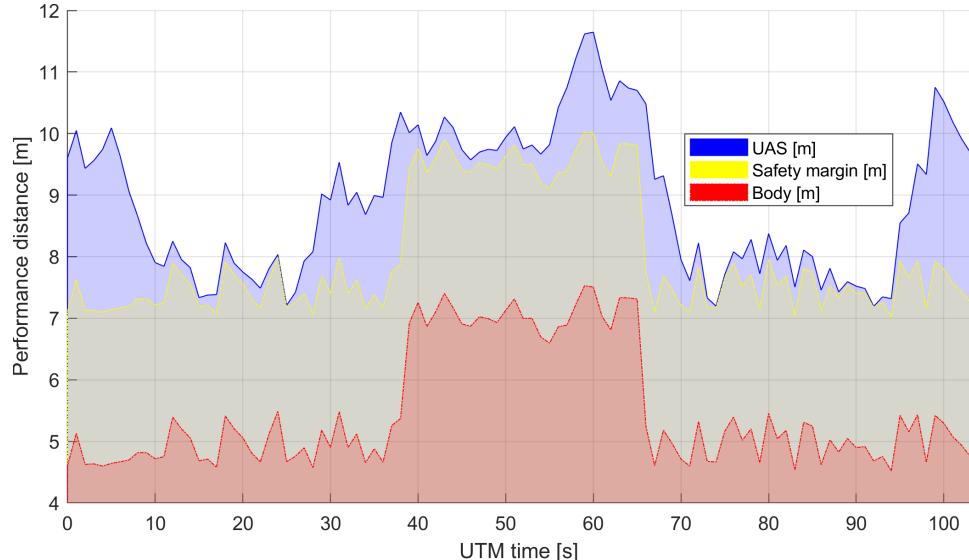
Figure 7.9: Test scenario for *Maze*.

Distance to Body/Safety Margin Evolution: The evolution of *body* and *safety margin* over time (x-axis, sec) given in meters distance (y-axis, m) is given in (fig. 7.10).

The *UAS* center distance to the nearest obstacle (blue line) does not break any *Safety Margin* (yellow line) of the closest obstacle. *Body Margin* of the closest obstacle (red line) has not been broken, because it always lies below of *Safety Margin* (yellow).

For *UTM period* 37 to 68 s, there is a *margin spike* due to avoidance of bloated *Rectangle buildings* (fig. 7.9b) during the 2nd turn. The *acceptance criterion* for *Safety Margin* is satisfied.

Note. The *body* and *safety margin* is changing depending on *UAS position* and *orientation*. The changes are reflected in (tab. 7.17).

Figure 7.10: Distance to body/safety margin evolution for *Maze scenario*.

Distance to Body/Safety Margin Peaks: The minimal and maximal values for *UAS distance to safety margin* based on performance (fig. 7.10) is summarized in (tab. 7.17).

The *minimal distance to safety margin* is $0.0131m$ which can be taken as $\sim 0m$ due to the numerical error. The *maximal distance to safety margin* is $2.9513m$ which is $5 \times$ *UAS radius*. The safety margin distance is $\leq 3m$ which means the scenario is tightly packed with obstacles. The *UAS* never left *Emergency Avoidance Mode* because of the condition: $safetyMarginDistance \geq avoidanceGridLength$ was never satisfied.

The *minimal body* distance is $5.0131m$, while the *maximal body* distance is $8.7117m$. The difference between minimal and maximal body distance is $\sim 4m$ which also indicates scenario packed with obstacles.

Parameter	UAS 1	
Distance to Safety Margin	min	0.0131
	max	2.9513
Distance to Body Margin	min	5.0131
	max	8.7117

Table 7.17: Distance to body/safety margin peaks for *Maze scenario*.

Path Tracking Performance: Reference path (green dashed) line is given as direct interconnection of *initial position* (green square with S marker) and *hidden waypoint* (green square with 1 marker). The *UTM Reference Time* is given on x-axis. The evolution of the real trajectory (solid blue line) for each axis is given as follow:

1. *X-axis path tracking* - reflects the maneuvering in the curves of the maze.
2. *Y-axis path tracking* - shows horizontal progress to the *hidden goal waypoint*. The expected linear tracking is not achievement due to the maneuvering delays on X-axis.
3. *Z-axis path tracking* - shows perfect linear tracking of the reference trajectory. The *altitude acceptance criterion*: $-5m \leq altitude \leq 5m$ have been fulfilled.

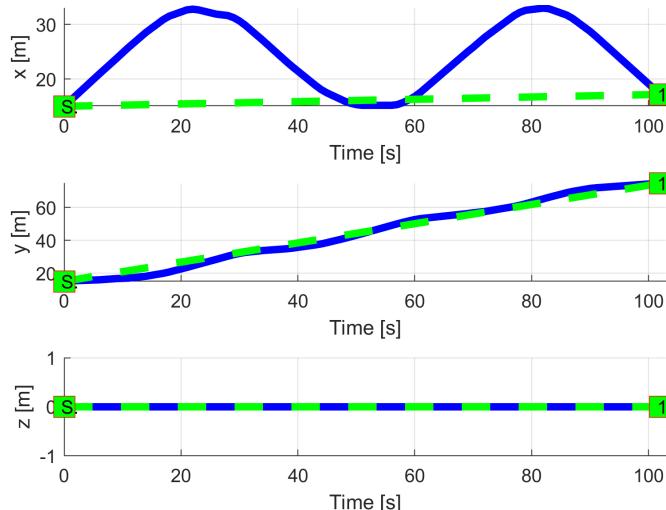


Figure 7.11: *Maze path tracking*.

Path Tracking Deviations: Deviations (tab. 7.18) from *reference trajectory* are in expected ranges considering the *mission plan* (tab. 7.15) and *obstacle properties* (tab. 7.16).

Param.	UAS 1
\mathcal{WP}_1	
$\max x $	27.32
$\max y $	2.41
$\max z $	0
$\max dist.$	28.06

Table 7.18: Path tracking properties for *Maze* scenario.

Computation Load: The *computation load* for *scenario* (fig.7.12) shows used time (y-axis) over decision frame (x-axis).

The UAS is constantly in *Emergency Avoidance Mode*; the *operational environment* is *cluttered* with obstacles. This causes very high *computation load*.

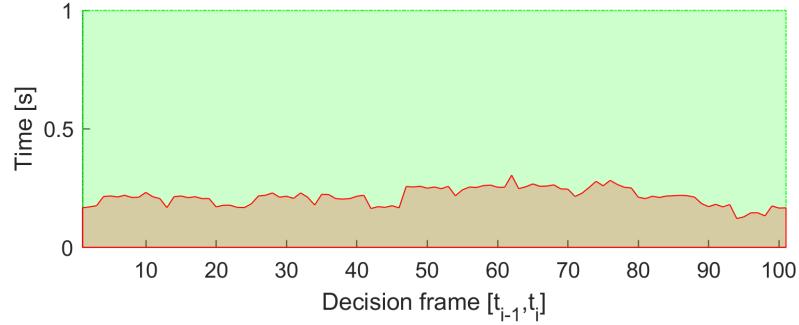


Figure 7.12: Computation time for *Maze* scenario.

7.3.4 Storm

Scenario: Small UAS is flying in open space in uncontrolled airspace (≤ 500 feet AGL (Above Ground Level)). A *Weather Service* notices UAS about *Dangerous Weather zone* (virtual constraint s. ??) which is moving in UAS direction. The *UAS* is executing mission given by (tab. 7.19).

Position		\mathcal{WP}_1
$[x, y, z]$	$[\theta, \varpi, \psi]$	
$[0, 0, 0]^T$	$[0^\circ, 0^\circ, 90^\circ]^T$	$[0, 60, 0]^T$

Table 7.19: Mission setup for *Storm* scenario.

Constraints: The *storm* is modeled as a *virtual constraint* with parameters given in (tab. 7.20). A constraint is modeled as a *convex polygon* for *horizontal boundary* and altitude for the *vertical boundary*.

The *Storm* is moving through an *operational region* with linear velocity $0.5ms^{-1}$. The *storm's center* was first detected at *decision frame* 0 at position $[0, 50, 0]^T$.

Constraint			Body Margin			Safety Margin
i. position	velocity	type	min.	max.	avg.	
$[0, 50, 0]^T$	$[0, -0.5, 0]$	polygon	9	10	9.5	5

Table 7.20: Constraint set for *Storm* scenario.

Assumption: Every *avoidable moving constraint* is usually slower than an *Approaching UAS*, or its radius is smaller than the turning radius of an *Approaching UAS*.

Note. Manned aviation receives a permit to operate in *controlled airspace* only if it has capability outmaneuver every known threat in requested airspace.

The *Constrained space portion* is usually very large, therefore in the majority of cases the assumption $uasSpeed >> constraintSpeed$ holds.

Main Goal: Show dynamic moving constraint avoidance capability in *uncontrolled airspace*.

Acceptance criteria:

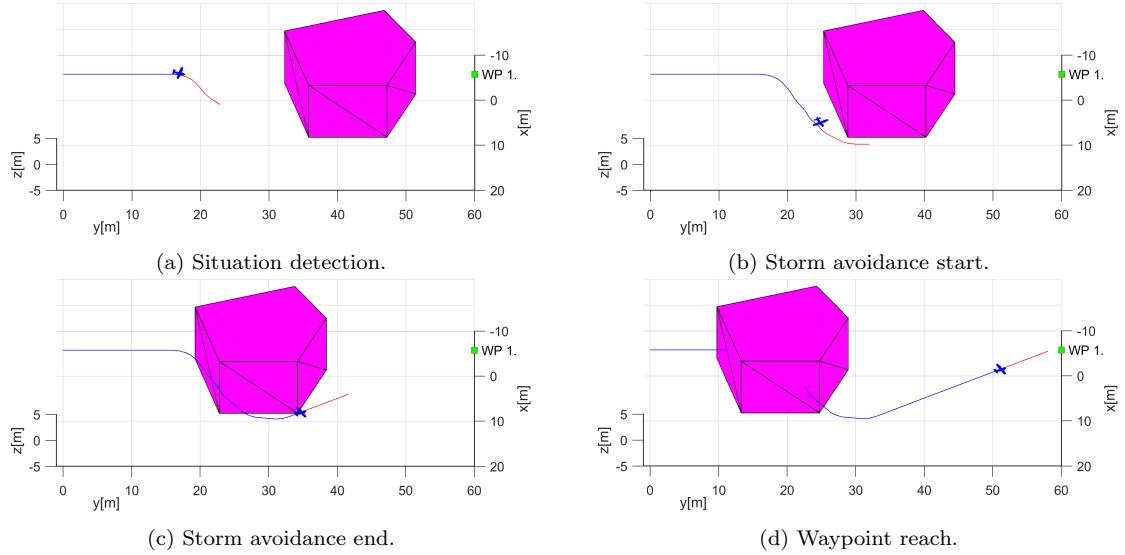
1. *Hard constraint avoidance* - the *UAS* must not cross the body margin: $distance(stormCenter, UAS) \geq bodyMargin$.
2. *Soft constraint avoidance* - the *UAS* cannot cross the safety margin to get into proximity of *Storms surrounding area*: $distance(stormCenter, UAS) \geq safetyMargin$.

Testing setup: The *standard test setup* defined in (tab. 7.2, 7.3, 7.4, 7.5, 7.6) is used with following parameter override:

1. *Avoidance grid - type* - *ACAS-like* with *horizontal enabled maneuvers*.

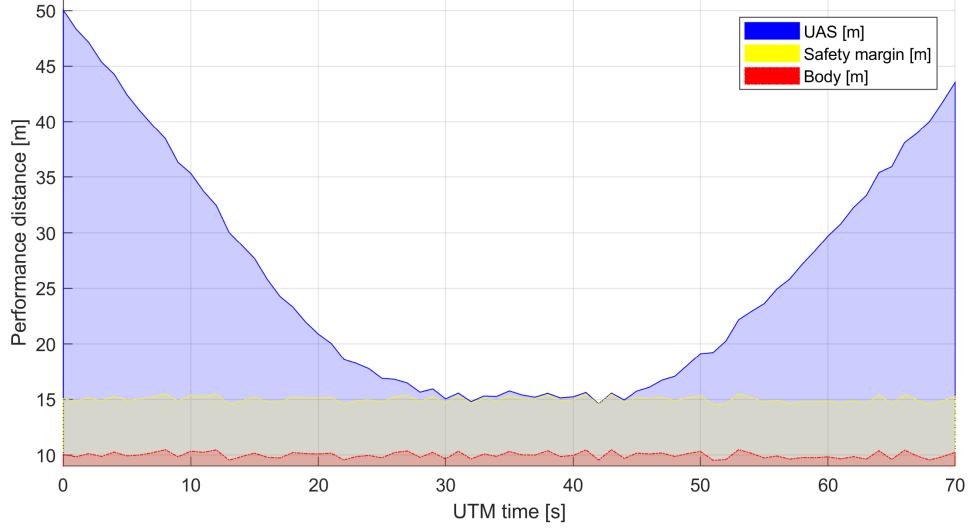
Simulation run: *Notable moments* from a *simulation run* (fig. 7.13) are the following:

1. *Detection* (fig. 7.13a) - the *Storm* (magenta polygon) is detected prior to the engagement (retrieved from associated weather service). The *UAS* (blue) stays in *Navigation mode*. *Trajectories in Navigation grid* are constrained by rule *Enforce safety margin* (tab. ??). The *Planned trajectory* (red) changes to avoid *Storm*.
2. *Avoidance start* (fig. 7.13b) - when UAS reaches optimal avoidance distance, the *navigation reach set* is constrained, forcing UAS to perform an evasive maneuver.
3. *Avoidance end* (fig. 7.13c) - navigation space is no longer constrained when the *minimal safe distance/heading* is achieved.
4. *Waypoint reached* (fig. 7.13d) - standard waypoint navigation procedure was used in this case.

Figure 7.13: Test scenario for *Storm* (Dynamic hard constraint).

Distance to Body/Safety Margin Evolution: The *body margin* (red line) and *safety margin* (yellow line) and *UAS distance to storm center* (blue line) evolution over *UTM time* (x-axis) are given in (fig. 7.14). The *body* and *safety* margin was changing according to the mutual position of the *storm* and the *UAS* (see tab. 7.20).

The acceptance criteria for the *hard constraint avoidance* and *soft constraint avoidance* have been fulfilled.

Figure 7.14: Distance to body/safety margin evolution for *Storm scenario*.

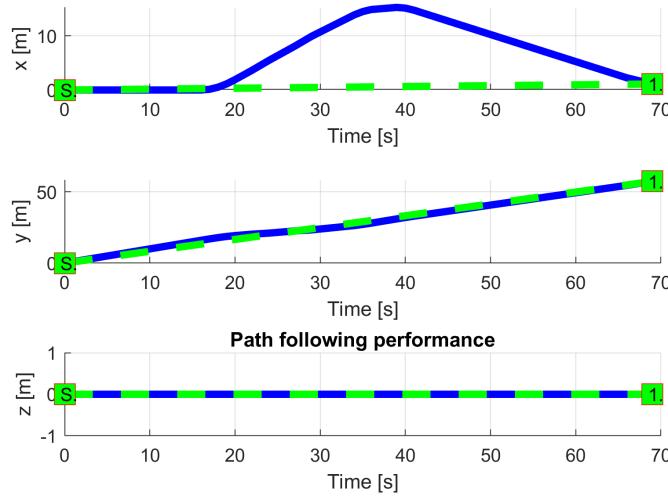
Distance to Body/Safety Margin Peaks: A *hard constraint* of *body margin* was not breached, because the $\text{distance}(\text{UAS}(t), \text{stormBody}(t))$ was all time greater than 0. Thus the *UAS* stayed well clear from *Storm*. The summary (tab. 7.21) shows that the *minimal body margin distance* was 5.0335 m, which proves *avoidance of hard constraint*.

A *soft constraint* represented as a *safety margin* (protective coating around storm body) was not breached, because the $\text{distance}(\text{UAS}(t), \text{stormBody}(t)) - \text{safetyMargin}(t)$ was all time greater than 0. The summary (tab. 7.21) show that the *minimal safety margin distance* was 0.0355 m, which proves *avoidance of soft constraints*.

Parameter		UAS 1
Distance to Safety Margin	min	0.0355
	max	34.9934
Distance to Body Margin	min	5.0355
	max	39.9934

Table 7.21: Distance to body/safety margin peaks for *Storm scenario*.

Path Tracking Performance: The *path tracking* (solid blue line) of *reference trajectory* (green dashed line) between *starting waypoint* (green square marked "S") and *final waypoint* (green square marked "1") is portrayed in (fig. 7.15). The *UAS* executes *horizontal right-side avoidance* of the *Storm* as is preferred.

Figure 7.15: *Storm* avoidance scenario path tracking.

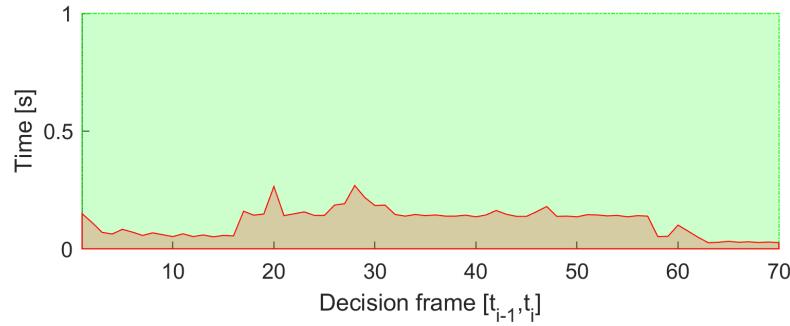
Path Tracking Deviations: *Deviations* (tab. 7.22) are in expected ranges considering the mission plan (tab. 7.19) and *body* and *safety* margins (tab. 7.20).

Param.	UAS 1
	\mathcal{WP}_1
$\max x $	15.26
$\max y $	1.32
$\max z $	0
$\max dist.$	15.76

Table 7.22: Path tracking properties for *Storm* scenario.

Computation Load: The *computation load* for *scenario* (fig.7.16) shows used time (y-axis) over decision frame (x-axis).

The *computation time* is low; it only increases slightly during avoidance maneuver.

Figure 7.16: Computation time for *Maze* scenario.

7.3.5 Emergency Converging

Scenario: Two *UAS* are flying in *uncontrolled airspace* (altitude ≤ 500 ft. Above the Ground Level) with missions defined in (tab. 7.23). Both *UAS* are in the *Navigation mode* with active *ADS-B-In/Out*, receiving position notification from each other. Cruising altitude is sufficient for horizontal separation (50-100 ft. Above the Ground Level). *Horizontal separation* is the preferred separation type for both *UAS*.

UAS	Position		WP_1
	$[x, y, z]$	$[\theta, \varpi, \psi]$	
1	$[0, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[40, 20, 0]^T$
2	$[20, 0, 0]^T$	$[0^\circ, 0^\circ, 90^\circ]^T$	$[20, 40, 0]^T$

Table 7.23: Mission setup for *Emergency converging* scenario.

Note. Collision point is expected at $\mathcal{C} = [20, 20, 0]^T$. The angle of approach is 90° which classifies the situation as *Converging maneuver* (fig. ??).

Main Goal: Show two *non-cooperative* UAS avoidance capability for *Converging maneuver* scenario in *uncontrolled airspace*.

Acceptance criteria:

1. *Proper mode invocation* - when an intruder intersects the UAS with *Right of the Way* navigation grid, both UAS will switch into *Emergency Avoidance Mode*.
2. *Minimal safety margin distance* $\geq 0m$.
3. *Each UAS* will reach own goal waypoint (tab. 7.23).

Testing setup: The *standard test setup* for each UAS defined in (tab 7.2, 7.3, 7.4, 7.5, 7.6) is used with the following without parameter override.

Intruder intersection model has been chosen depending on UAS (tab. 7.24). Each UAS is equipped with *ADS-B In/Out* sensor obtaining/distributing the following information:

1. *Position* - in operational section coordinate frame.
2. *Velocity* - vector representation in the given coordinate frame.

3. *Class size* - class body radius based on UAS propulsion and size.

4. *Safety margin set* - set of safety margins for different collision cases.

Avoidance parameters for the *Emergency converging scenario* are given in (tab. 7.24). Each UAS has the same speed set to 1ms^{-1} . Second UAS has the *Right of Way*.

The *safety margin* is considered as sum of both participants *near miss margins*. In this case, the default safety margin is considered as 1.2 m .

UAS	Parameters			Margins		Separation
	velocity	intruder model	ROW	body	safety	
1	1	body + spread	false	0.3	0.6	horizontal
2	1	body + spread	true	0.3	0.6	horizontal

Table 7.24: Avoidance parameters for *Emergency converging* scenario.

Note. Both UAS are using body (app. ??) and spread (app. ??) intersection models, reflecting both body volume and maneuverability of intruder. Both UAS have preferred separation mode as *horizontal*, typical for planes.

Simulation Run: Notable moments from the simulation run (fig. 7.17) are following

1. *Detection* (fig. 7.17a) - Intruder (UAS2 cyan) is approaching (UAS 1 blue) from the right side, Intruder (UAS2 cyan) has the right of way, because of $70^\circ \leq \text{angleOfApproach} < 130^\circ$. *Intruder intersection model* (for UAS 2) is created and propagated in *avoidance grid* (for UAS 1).
2. *Start Converging* (fig. 7.17b) - when *UAS 2 (cyan) parametric intruder intersection model* disables *trajectories*, converging maneuver for UAS 1 (blue) starts.
3. *Near miss case* (fig. 7.17c) - UAS 1 (blue) to UAS 2 (cyan) closest distance. The safety margin for *near miss* has not been breached. The safety margin for *well clear* in uncontrolled airspace is invalid.
4. *Waypoint reached* (fig. 7.17d) - the intruder intersection model for *UAS 2 (cyan)* is removed from UAS 1 (blue) *avoidance grid* after *converging maneuver competition*, standard navigation procedure is applied afterward.
5. Note that *UAS 2 (cyan)* has the *Right of way* in (tab. 7.24).
6. Note that *UAS 1 (blue)* used only horizontal separation (priority) in (fig. 7.19a).

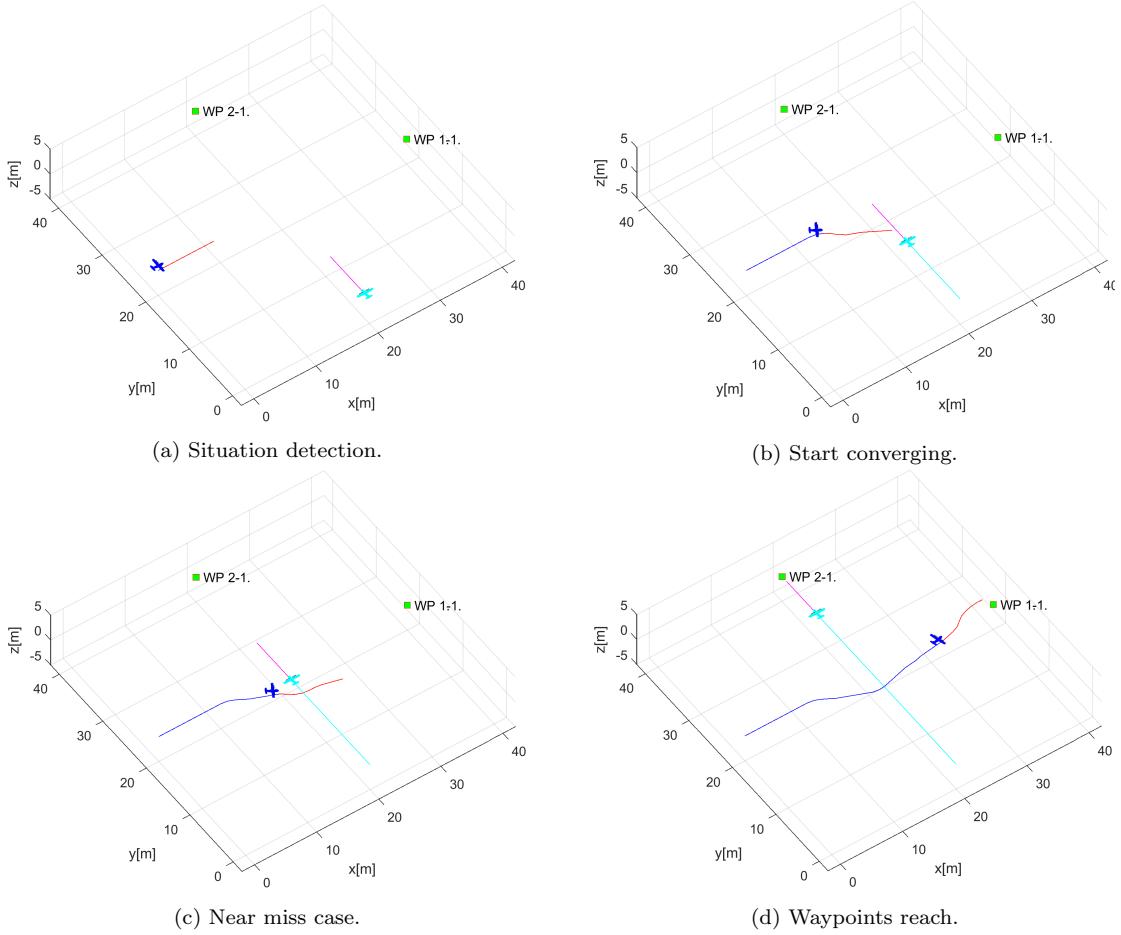


Figure 7.17: Test scenario for *Emergency converging* (Intruder avoidance).

Distance to Safety Margin Evolution: There is a need to compare the mutual distance between both UAS (y-axis [m]) and its evolution over UTM time (x-axis [s]). The *mutual distance* of *UAS 1* to *UAS 2* is given by *blue line*. The *Safety margin* value is denoted by the *red line* at a *constant value* of 1.2 m.

The *Proper avoidance Invocation* is shown when UAS systems are getting closer to each other, and they enter (Emergency Avoidance Mode) to provide *active separation*. The *Mutual distance evolution* (blue line) does not cross *safety margin* (red line).

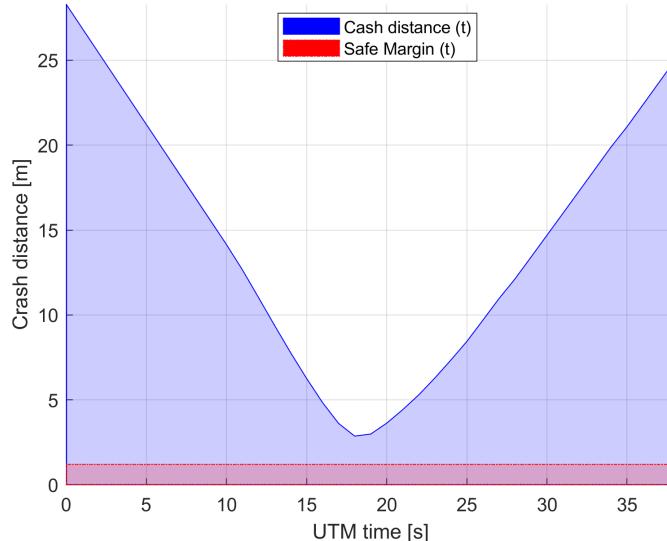


Figure 7.18: Distance to safety margin evolution for *emergency converging scenario*.

Distance to Safety Margin Peaks: Minimal and Maximal mutual distance to safety margin is summarized in (tab. 7.25). The closest to the collision are UAS systems when the distance to safety margin is $1.6676m$.

The *minimal distance to safety margin* ≥ 0 which means that the *safety acceptance criterion* is fulfilled.

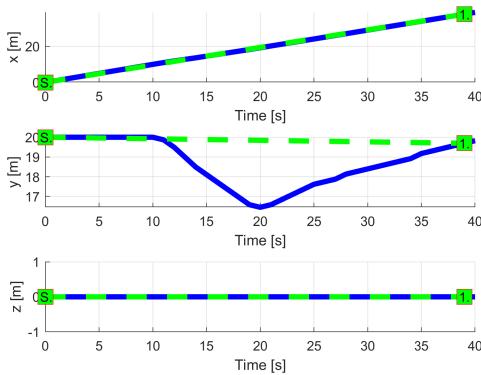
UAS:			1-2
Distance to Safety Margin	min	1.6676	
	max	27.0843	

Table 7.25: Distance to safety margin peaks for the *emergency converging scenario*.

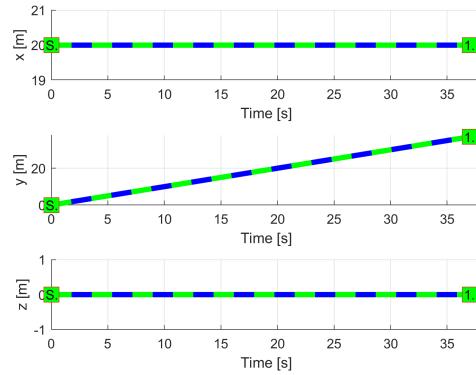
Path Tracking Performance: All waypoints (green numbered squares) for both UAS have been reached (fig. 7.19). *Reference trajectories* (green dashed lines), between the initial position (green square marked S) and goal waypoint (green square marked 1) are split into three XYZ values with respective figures. The tracked value is on y-axis [m] and time on x-axis [s]. The blue lines represent real parameter evolution over time.

Following observations can be made from path tracking (fig. 7.19) and preferred separations (tab. 7.24):

1. UAS 1 (fig. 7.19a) is using *horizontal separation* (y-axis). The UAS diverges from the reference trajectory to minimum necessary time.
2. UAS 2 (fig. 7.19b) has the right of way and is not using any active avoidance mechanism.



(a) UAS 1.



(b) UAS 2.

Figure 7.19: *Trajectory tracking for Emergency converging test case*.

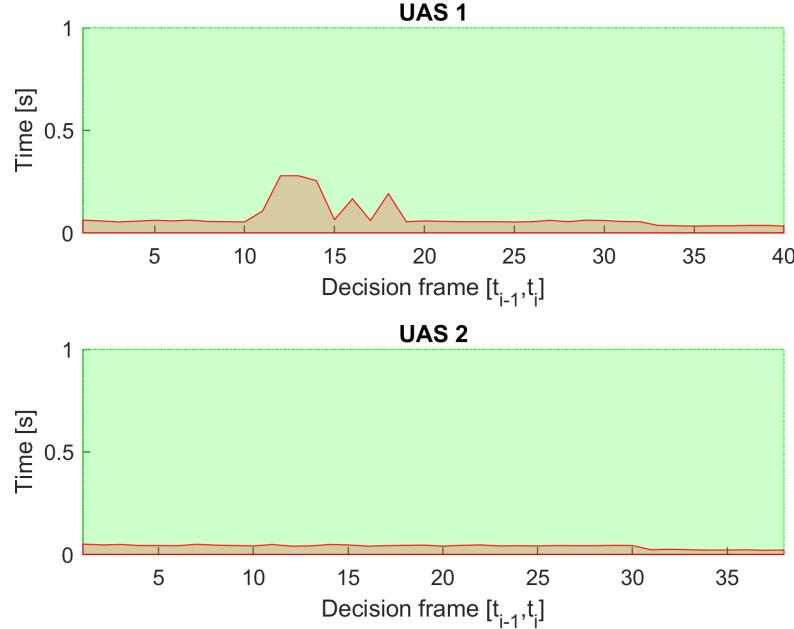
Path Following Deviations: *Deviations* (tab. 7.26) are in expected ranges considering the *mission plans* (tab. 7.23) and *separation safety margin* (tab. 7.24).

Param.	UAS 1	UAS 2
	\mathcal{WP}_1	\mathcal{WP}_1
$\max x $	0	0
$\max y $	3.25	0
$\max z $	0	0
$\max dist.$	3.25	0

Table 7.26: Path tracking properties for the *Emergency converging* scenario.

Computation Load: The *computation load* for *scenario* (fig.7.20) shows used time (y-axis) over decision frame (x-axis).

The *computation time* is increased only for UAS 1 during the avoidance period. The UAS 2 remains unaffected because it has the right of way.

Figure 7.20: Computation time for *Emergency converging* scenario.

7.3.6 Emergency Head-On

Scenario: Two *UAS* systems are flying in *uncontrolled airspace* (altitude ≤ 500 ft. Above the Ground Level) with missions defined in (tab. 7.27). Both *UAS* are in the *Navigation mode* with active *ADSB-In/Out*, receiving position notifications from each other. Cruising altitude is sufficient for horizontal separation (50-100 ft. Above Ground Level). *Horizontal separation* is preferred mode for both *UAS*.

UAS	Position		\mathcal{WP}_1
	$[x, y, z]$	$[\theta, \varpi, \psi]$	
1	$[0, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[40, 20, 0]^T$
2	$[40, 20, 0]^T$	$[0^\circ, 0^\circ, 180^\circ]^T$	$[0, 20, 0]^T$

Table 7.27: Mission setup for *Emergency head-on* scenario.

Note. *Collision point* is expected at $\mathcal{C} = [20, 20, 0]^T$. The *angle of approach* is 180° which

classifies the situation as *Head-on maneuver* (fig. ??).

Main Goal: Show two *non-cooperative* UAS avoidance for *Head-on approach scenario* in *uncontrolled* airspace.

Acceptance criteria:

1. *Proper mode invocation* - when an intruder intersects the opposing *UAS* Navigation grid, bot intruder and *UAS* will switch to *Emergency Avoidance Mode*. None of the *UAS* have the *Right of Way*.
2. *Minimal Safety Margin distance* $\geq 0m$. That means the mutual distance of both *UAS centers* does not go below-given *safety margin*.
3. *Both UAS* will reach own goal waypoint (tab. 7.27).

Testing setup: The *standard test setup* for each *UAS* defined in (tab 7.2, 7.3, 7.4, 7.5, 7.6) is used with the following without parameter override.

Intruder intersection model has been chosen depending on *UAS* (tab. 7.28). Each *UAS* is equipped with *ADS-B In/Out* sensor obtaining/distributing the following information:

1. *Position* - in operational section coordinate frame.
2. *Velocity* - vector representation in the given coordinate frame.
3. *Class size* - class body radius based on *UAS* propulsion and size.
4. *Safety margin set* - set of safety margins for different collision cases.

Avoidance parameters for the *Emergency head-on scenario* are given in (tab. 7.28). Each *UAS* has the same speed set to $1ms^{-1}$. None of them have the *Right of Way*.

The *safety margin* is considered as a sum of both participants *near miss margins*. In this case, the default safety margin is considered as $1.2 m$.

UAS	Parameters			Margins		Separation
	velocity	intruder model	ROW	body	safety	
1	1	body (timed)	false	0.3	0.6	horizontal
2	1	body (timed)	false	0.3	0.6	horizontal

Table 7.28: Avoidance parameters for *Emergency head on* scenario.

Note. Both *UAS* are using body (app. ??) intersection model, reflecting both body volume along the expected trajectory. Both *UAS* have a preference for *horizontal* separation mode, typical for planes.

Simulation Run: Notable moments from the simulation run (fig. 7.21) are the following:

1. *Situation detection* (fig. 7.17a) - *UAS 1* (blue) is approaching *UAS 2* (cyan) with $130^\circ \leq angleOfApproach \leq 180^\circ$, this is considered head-on approach. Head-on approach gives the *right of the way* neither to *UAS 1* nor *UAS 2*. An *intruder intersection model* for

opposite UAS is created in respective *avoidance grids*. *Head on emergency avoidance* starts independently in each UAS without intruders coordination. First *avoidance maneuver* is invoked when the *intruder intersection model* constraints any trajectory in the *avoidance grid*. When this happens *Navigation mode* switch to the *Emergency avoidance mode*.

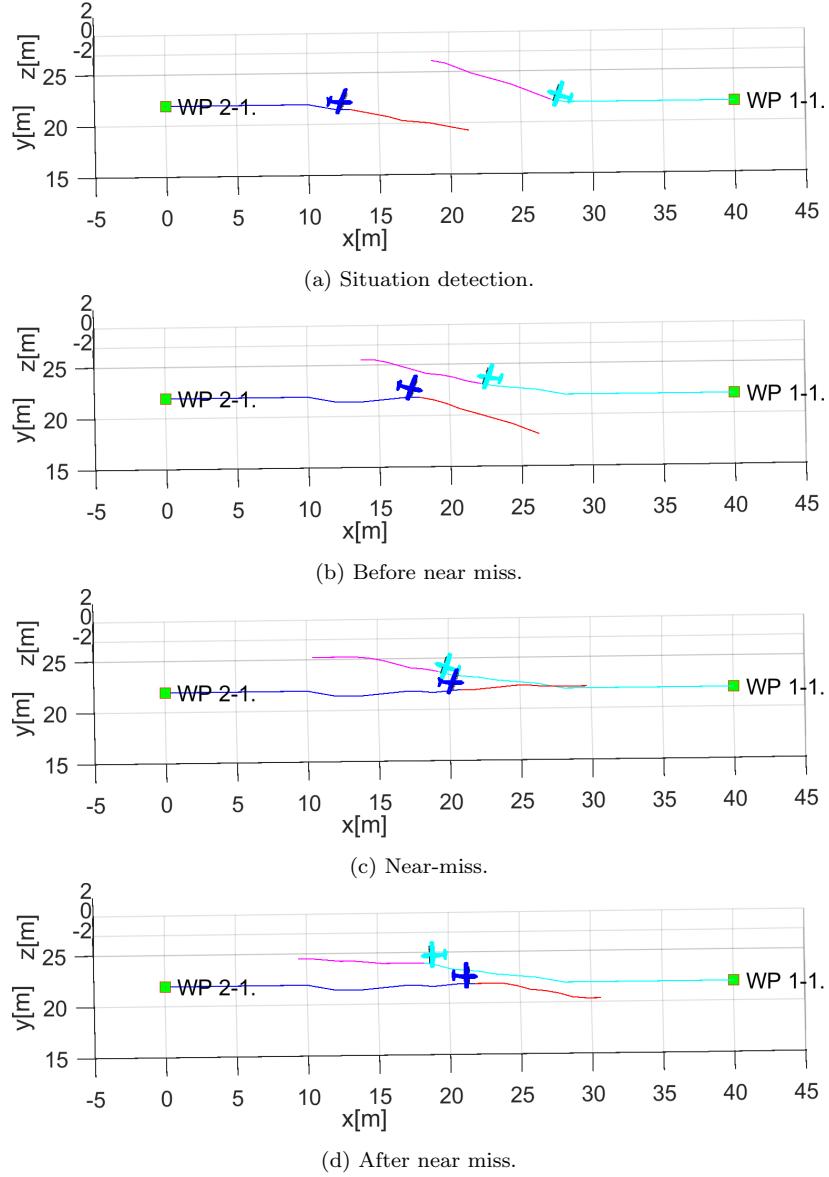


Figure 7.21: Test scenario for *Emergency head-on approach* (Intruder avoidance).

2. *Before near miss* (fig. 7.21b) - both *UAS* are in *emergency avoidance mode*, sticking to right side avoidance maneuver.
3. *Near miss case* (fig. 7.21c) - *UAS 1* to *UAS 2* closest distance. The safety margin for *near miss* has not been breached. The safety margin for *well clear* in uncontrolled airspace is invalid. Both *UAS* are using also *Horizontal separation* to avoid each other, *Emergency avoidance mode* is switched to the *Navigation mode* when the risk of an *aerial clash* is voided.
4. *After near-miss* (fig. 7.21d) - both *UAS* are tracking back to respective waypoint, correcting *altitude* (*Z-axis* in (fig. 7.23)) first.

Note. *Collision point* was expected at $\mathcal{C} = [20, 20, 0]^T$.

Note. Both UAS used horizontal (primary), vertical (secondary) separation (fig 7.23).

Note. Both UAS decision times were synchronized, this is not an assumption, but it shows critical performance. Usually, safety margin is bloated for (eq.??).

Distance to Safety Margin Evolution: There is a need to compare the mutual distance between both UAS (y-axis [m]) and its evolution over synchronized *UTM time* (x-axis [s].) The mutual distance between bodies of *UAS 1, UAS 2* (blue line) compared to *Safety Margin* (red line) is given in (fig. 7.22). The *Safety Margin* value was constant for all time at value 1.2 m which is double of *Near Miss Margin for UAS 1 UAS 2*.

The proper *Avoidance Invocation* is shown when *UAS* systems are getting closer to each other, and they start their *separation phase* (Emergency Avoidance Mode switch). The mutual distance (blue line) does not cross *safety margin* (red line).

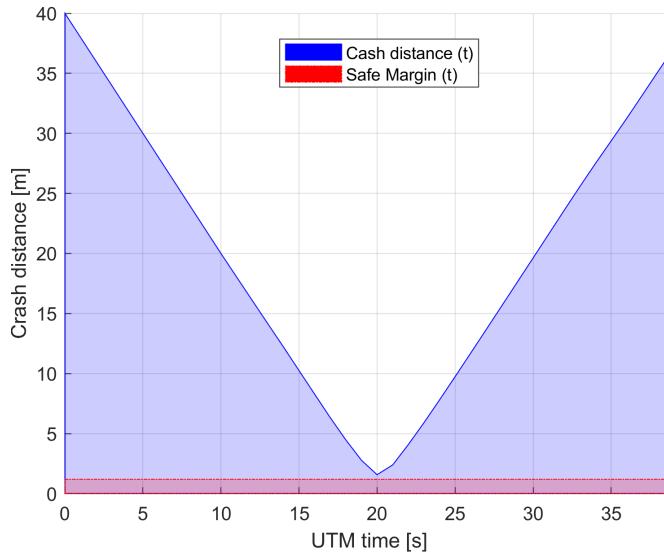


Figure 7.22: Distance to safety margin evolution for *emergency head-on scenario*.

Distance to Safety Margin Peaks: Minimal and Maximal mutual distance to safety margin is summarized in (tab. 7.29). The closest to the collision are UAS systems when the *distance to safety margin* is 0.3824m.

The *minimal distance to safety margin* ≥ 0 which means that the *safety acceptance criterion* is fulfilled.

UAS:	1-2	
Distance to Safety Margin	min	0.3824
	max	38.8000

Table 7.29: Distance to safety margin peaks for *Emergency head-on scenario*.

Path Tracking Performance All waypoints (green numbered squares) for both UAS have been reached (fig. 7.23). *Reference trajectories* (green dashed lines), between the initial position (green square marked S) and goal waypoint (green square marked 1) are split into three XYZ values with respective figures. The tracked value is on y-axis [m] and time on x-axis [s]. The blue lines represent real parameter evolution over time.

Following observations can be made from path tracking (fig.7.23) and preferred separations (tab. 7.28):

1. UAS 1 (fig. 7.23a) is using horizontal separation going to the right (y-axis) and a little bit up (z-axis).
2. UAS 2 (fig. 7.23b) is using horizontal separation going to the right (left in GCS, y-axis) and a little bit up (z-axis).

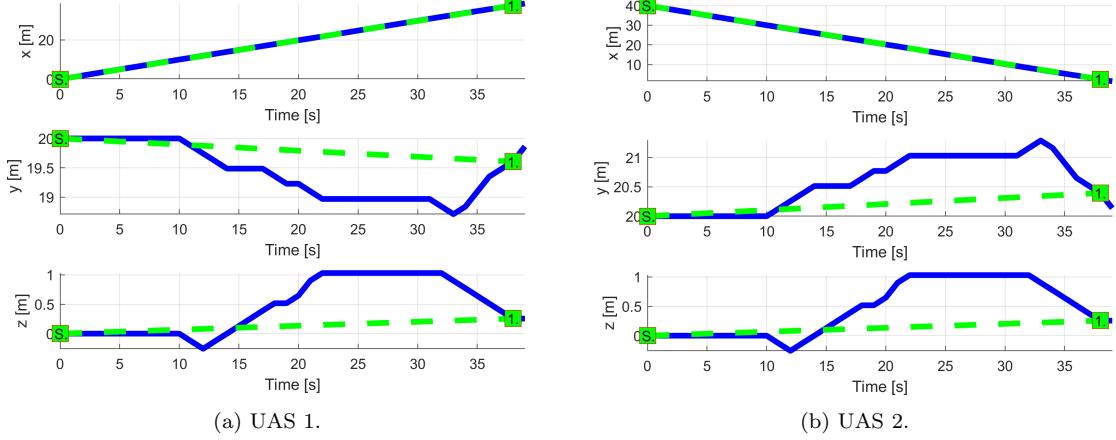


Figure 7.23: *Trajectory tracking for Emergency head-on test case.*

Path Following Deviations: *Deviations* (tab. 7.30) are in expected ranges considering the *mission plans* (tab. 7.27) and *separation safety margins* (tab. 7.28).

Param.		
	UAS 1	UAS 2
	\mathcal{WP}_1	\mathcal{WP}_1
$\max x $	0.05	0.06
$\max y $	1.37	1.48
$\max z $	1.03	1.05
$\max dist.$	1.39	1.52

Table 7.30: Path tracking properties for *Emergency head-on* scenario.

Computation Load: The *computation load* for scenario (fig.7.20) shows used time (y-axis) over decision frame (x-axis).

The *computation time* is increased only during the *avoidance phase*. The *load* is symmetric for both UAS systems.

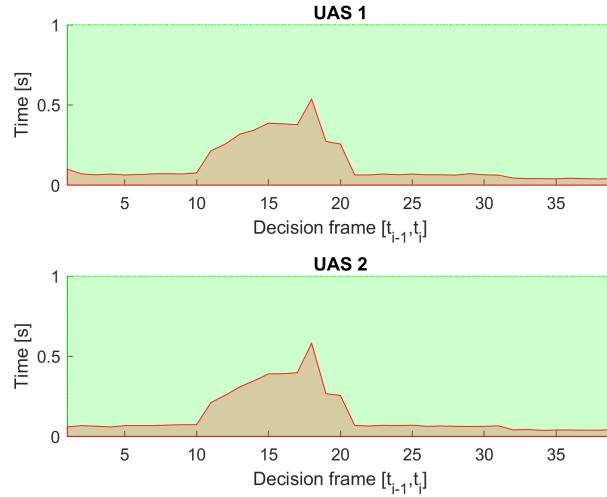


Figure 7.24: Computation time for *Emergency head-on* scenario.

7.3.7 Emergency Mixed Head-On with Converging

Scenario: Four UAS are flying in *uncontrolled airspace* (altitude ≤ 500 ft. Above the Ground Level) missions defined in (tab. 7.31). All UAS are in the *Navigation mode* with active *ADS-B In*, receiving *position notifications* from each other. Cruising altitude is sufficient for *horizontal separation* (50-100 ft. Above the Ground Level).

UAS	Position		\mathcal{WP}_1
	$[x, y, z]$	$[\theta, \varpi, \psi]$	
1	$[0, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[45, 20, 0]^T$
2	$[40, 20, 0]^T$	$[0^\circ, 0^\circ, 180^\circ]^T$	$[-5, 20, 0]^T$
3	$[20, 0, 0]^T$	$[0^\circ, 0^\circ, 90^\circ]^T$	$[20, 45, 0]^T$
4	$[20, 40, 0]^T$	$[0^\circ, 0^\circ, -90^\circ]^T$	$[45, 20, 0]^T$

Table 7.31: Mission setup for the *Emergency mixed* scenario.

Note. Collision point is expected at $\mathcal{C} = [20, 20, 0]^T$

Main Goal: Show *multiple non-cooperative intruders avoidance capability* in *uncontrolled airspace*.

Acceptance criteria:

1. Proper avoidance mode invocation - when an *intruder intersection model* impact the *Avoidance Grid*, UAS system will switch to an *Emergency avoidance mode*.
2. Minimal safety margin distance $\geq 0m$.
3. Each *UAS* will reach own goal waypoint (tab. 7.31).

Testing setup: The *standard test setup* for each UAS defined in (tab 7.2, 7.3, 7.4, 7.5, 7.6) is used with the following without parameter override.

Intruder intersection model has been chosen depending on UAS (tab. 7.32). Each UAS is equipped with *ADS-B In/Out* sensor obtaining/distributing the following information:

1. *Position* - in operational section coordinate frame.
2. *Velocity* - vector representation in the given coordinate frame.
3. *Class size* - class body radius based on UAS propulsion and size.
4. *Safety margin set* - set of safety margins for different collision cases.

Avoidance parameters for the *Emergency mixed scenario* are given in (tab. 7.32). Each UAS has different *intruder model* and separation combination. Each UAS has same the speed set to $1ms^{-1}$. None of UAS has the *Right of Way*.

The *safety margin* is considered as the sum of both participants *near miss margins*. In this case, the default safety margin is considered as 1.2 m.

UAS	Parameters			Margins		Separation
	velocity	intruder model	ROW	body	safety	
1	1	body + spread	false	0.3	0.6	horizontal
2	1	body (timed)	false	0.3	0.6	vertical
3	1	body (timed)	false	0.3	0.6	horizontal
2	1	body + spread	false	0.3	0.6	vertical

Table 7.32: Avoidance parameters for *Emergency mixed scenario*.

Note. Each *UAS* use different intruder intersection models and primary *separations* (defined in the tab. 7.32). UAS reactions are based on primary *Separation* mode, intruders intersection models this is reflected on major axial deviations in (fig. 7.27) and summarized in *path tracking* deviation (tab. 7.34).

Simulation Run: Notable moments from the simulation run (fig. 7.25) are the following:

1. *Situation detection* (fig. 7.25a) - UAS 1 (blue) is detecting UAS 2 (cyan), UAS 3 (green), and UAS 4 (black) as possible intruders. There are multiple converging and head on approaches depending on mutual positions (UAS and *angle of approach*). There exist at least one *converging case* where each UAS has the *Right of way*. Each UAS creates intruder intersection models depending on the intruder configuration (tab. 7.32). Each UAS enters into the *Emergency avoidance mode* independently, when at least one trajectory is constrained in the *avoidance grid*.
2. *Before near-miss* (fig. 7.25b) - all *UAS* are in *emergency avoidance mode*, using various *separation modes* and *intruder intersection models*. Each UAS is performing its own avoidance maneuver, constantly checking other intruders. If the same separation and the same intruder model were used, there would be a virtual roundabout.
3. *After near-miss* (fig. 7.25c) - all *UAS* avoided each other which is covered in *safety margin performance* (fig. 7.26) and (tab. 7.33).

4. *Situation resolution* (fig. 7.25d) - all *UAS* returns to *Navigation mode* correcting altitude first and continuing to assigned waypoints.

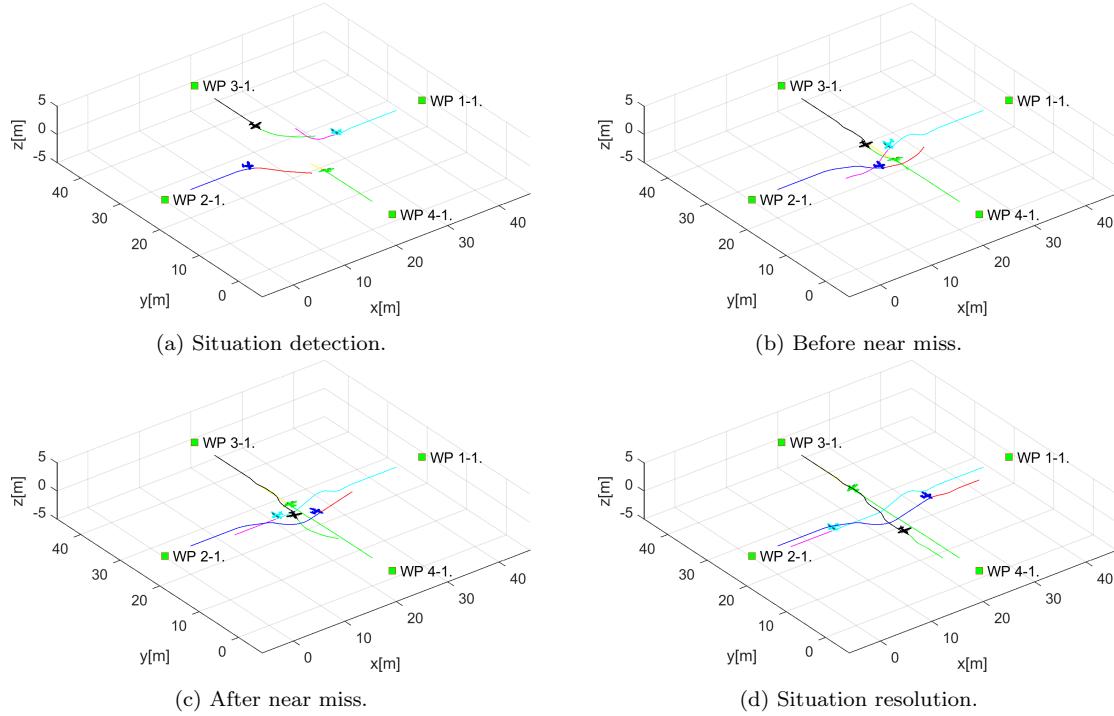


Figure 7.25: Test scenario for the *Emergency mixed* situation with the *self-separation mode*.

Distance to Safety Margin Evolution: There is a need to compare the mutual distance between each *UAS*. The graph (fig. 7.26) shows six figures for each *UAS systems* mutual distance (blue line) in this scenario. The *Safety Margin* (red line) (1.2 m) was not breached for any pair (case).

The *Proper avoidance invocation* is shown when *UAS* systems are getting closer to each other, and then they start separation phase (Emergency avoidance mode).

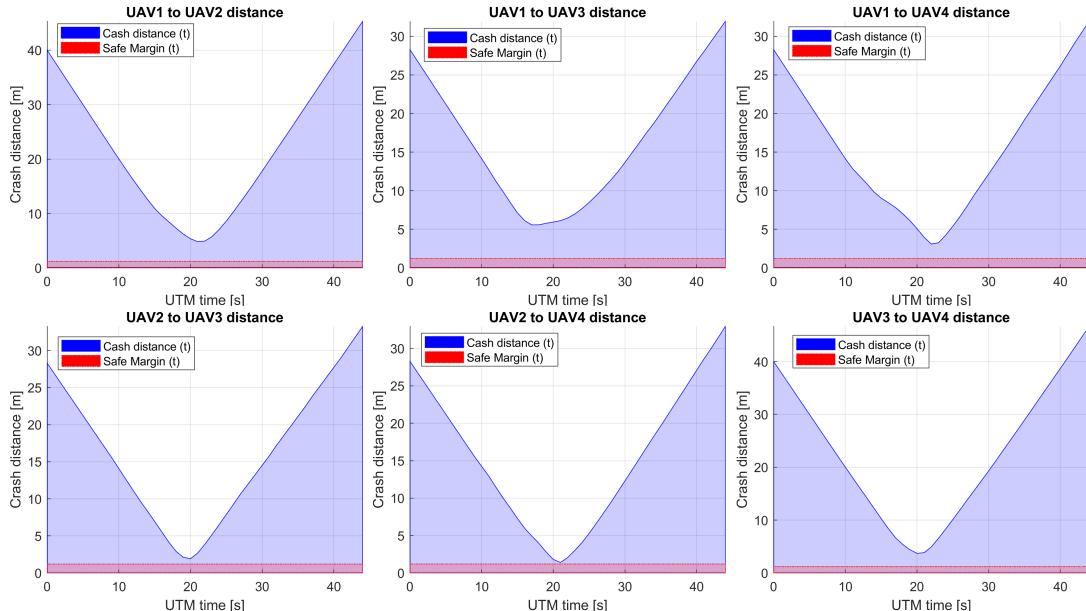


Figure 7.26: Distance to safety margin evolution for the *emergency mixed scenario*.

Distance to Safety Margin Peaks: Minimal and Maximal mutual distance to safety margin is summarized in (tab. 7.33). There is no detected breach for any combination.

The *closest to collision* is UAS pair 2 – 4 with mutual safety margin only 0.2019 m. On the other side is UAS pair 1 – 3 with mutual safety margin 4.3721 m.

The *minimal distance to safety margin* ≥ 0 which means that the *safety condition* is fulfilled.

UAS:	Distance to Safety Margin		
	min	max	breach
1-2	3.6231	44.0831	false
1-3	4.3721	30.7300	false
1-4	1.8959	30.7331	false
2-3	0.7331	32.0266	false
2-4	0.2019	31.7282	false
3-4	2.5171	45.4257	false

Table 7.33: Distance to safety margin peaks for the *emergency mixed scenario*.

Path Tracking Performance: All waypoints (Green numbered squares) for all UAS have been reached (fig. 7.27). *Reference trajectories* (green dashed line) have been tracked by *UAS real path* (solid blue line) almost all time.

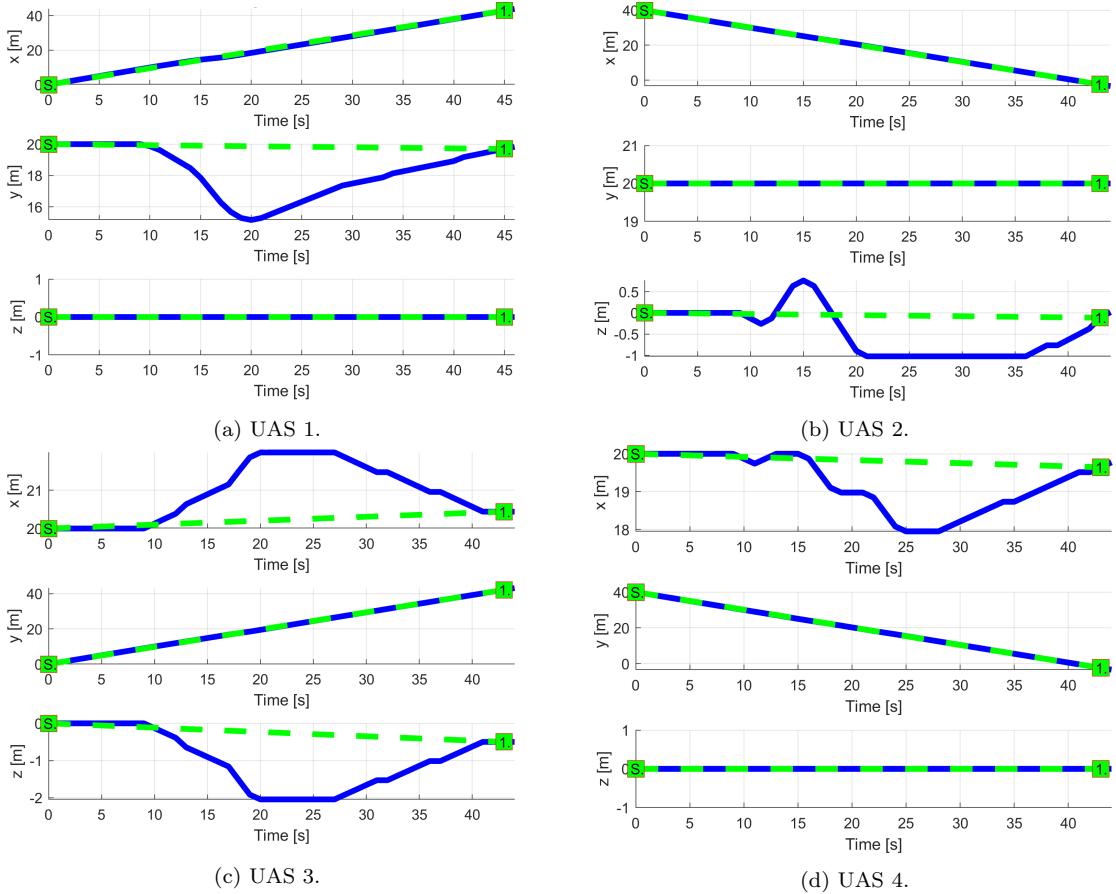


Figure 7.27: Trajectory tracking for the *Emergency mixed* situation test case.

Following observations can be made from *path tracking* (fig. 7.27) and *preferred separations* (tab. 7.32):

1. UAS 1 (fig. 7.27a) is using *horizontal separation* (y-axis right) having *preferred horizontal separation*.
2. UAS 2 (fig. 7.27b) is using vertical separation (z-axis up-down), having preferred vertical separation.
3. UAS 3 (fig. 7.27d) is using horizontal/vertical separation (x-right, z-down), having preferred horizontal separation. This UAS has used other than the preferred separation type.
4. UAS 4 (fig. 7.27c) is using horizontal separation (x-axis right/left), having preferred vertical separation. This UAS has used opposite separation type to preferred.

Path Tracking Deviations: *Deviations* (tab. 7.34) are in expected ranges considering the *mission plans* (tab. 7.31) and *separation safety margins* (tab. 7.32).

Param.	UAS 1	UAS 2	UAS 3	UAS 4
	\mathcal{WP}_1	\mathcal{WP}_1	\mathcal{WP}_1	\mathcal{WP}_1
$\max x $	0	0	1.98	2.05
$\max y $	4.84	0	0	0
$\max z $	0	1.23	2.43	0
$\max dist.$	4.84	1.23	3.45	2.05

Table 7.34: Path tracking properties for the *Emergency mixed* scenario.

Computation Load: The *computation load* for *scenario* (fig.7.28) shows used time (y-axis) over decision frame (x-axis).

The *computation time* increases during periods of *active avoidance*. The *shortest* period of avoidance has UAS 1 and the longest period of avoidance has UAS 4.

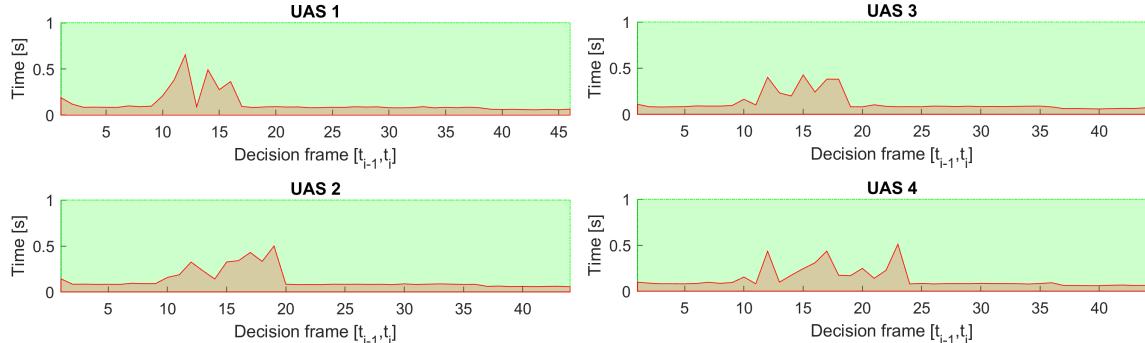


Figure 7.28: Computation time for *Emergency multiple* scenario.

7.4 Cooperative Test Cases

The *main goal* of this section is to show the operational capabilities of the *approach* under *UTM supervision*. The minimal UTM functionality set (sec. ??) has been implemented, including *position notifications mechanism, collision case calculation, resolution enforcement* components.

Test cases cover *well clear breach prevention, situation-based avoidance, and rules of the air enforcement*.

Coverage of *near miss situations, clash incidents* is given implicitly by *safety* and *body margins* (tab. ??).

1. *Rule-based converging* (sec. 7.4.1) covers *well clear breach* and the *converging rule of the air*, showing determinism and *UTM resolution execution*.
2. *Rule-based head-on* (sec. 7.4.2) covers *well clear breach* and the *head on rule of the air*, showing determinism and *UTM resolution execution*.
3. *Rule-based mixed head on with converging* (sec. 7.4.3) covers *well clear breach* and *head on and converging rules of the air*. The main focus is on a *virtual roundabout* concept when multiple collision cases are clustered into one avoidance maneuver.
4. *Rule-based overtake* (sec. 7.4.4) covers *well clear breach* during *overtaking* by faster UAS.

7.4.1 Rule-Based Converging

Scenario: Two *UAS* are approaching an *airway intersection* at the *same time* in *controlled airspace* (over 500 feet Above the Ground Level). The mutual position of *UAS* can be classified as *Side approach*. Following *collision hazards* are present:

1. *Active Converging Collision Hazard* - An *UAS* is approaching from the *right side*, which gives him *Right of the Way* and invokes the need to avoid *Intruder* actively.
2. *Passive Converging Collision Hazard* - An *UAS* is approaching from the *left side*, which gave us *Right of the Way* and imposes an obligation of *active avoidance* on other *UAS*.

Collision Hazards must be addressed by *UTM* service in the following manner:

1. *Each UAS* in particular *Controlled Space* periodically sends synchronized *Position Notification* messages (tab. ??).
2. *UTM* service receives *Position Notifications* and manages *Collision Case* (tab. ??) in *Controlled Space*.
3. *UTM* detects *Converging Collision Case* with *Collision Point* in the vicinity.
4. *UTM* service Sends *Mandate* to *UAS* without *Right of the Way* and implements *Normative Directive* on all *UAS* in the area.

Mission parameters for both UAS systems are defined in (tab. 7.35).

UAS	Position		WP_1
	$[x, y, z]$	$[\theta, \varpi, \psi]$	
1	$[0, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[40, 20, 0]^T$
2	$[20, 0, 0]^T$	$[0^\circ, 0^\circ, 90^\circ]^T$	$[20, 40, 0]^T$

Table 7.35: Mission setup for *Rule based converging* scenario.

Assumptions: Following assumptions are valid for this test:

1. *Controlled Airspace Airworthiness* - UAS system is equipped with necessary controlled airspace equipment like ADS-B In/Out, Radar, Transponder, etc. Moreover, airworthy *UAS* can precisely follow *UTM directives* (max. 5 % deviation).
2. *C2 (Command & control) Link Established* - necessary for $(UAS \leftrightarrow UAS)$ and $(UAS \leftrightarrow UTM)$ communication. If *C2* link is lost the *UAS* will enter into *Emergency avoidance mode*.
3. *Decision frame synchronization with UTM* - necessary in discrete C2 environment otherwise *safety margins* needs to be *bloated*.
4. *Both UAS have identical cruising speed* - simplification impacting *UTM* service implementation. *Obstacle Avoidance Framework* can comprehend various intruders speed, with proper *UAS* directives.

Main Goal: Show possibility of *Converging situation resolution* with *forced safety margin* by *UAS Traffic Management* system. The *Obstacle Avoidance Framework based on Reach Sets* is used as a *Navigation Module*.

Acceptance Criteria: Following criteria must be met:

1. *Well Clear Condition valid for both UAS* - Both *UAS* must have *minimal required distance* from *other UAS* for all *Converging Maneuver* enforcement time.
2. *Fulfillment of UTM Directives* - Both *UAS* must stay in a *Navigation mode* for all *Converging Maneuver* enforcement time. *UAS without Right Of the Way* must stay away for the necessary time, before returning to *Original Navigation waypoint WP₁* following.

Testing Setup: The *standard test setup* for each *UAS* defined in (tab. 7.2, 7.3, 7.4, 7.5, 7.6) is used with following parameter override:

1. *Navigation grid - type - ACAS-like* with horizontal enabled *maneuvers*

This *configuration* is based on the assumption that every *UAS* is in *controlled airspace* in *FL450* (flight level 45000 feet Above Sea Level), without permission for a *climb or descent maneuver*. The *rule engine* is initialized in standard *Rules of the air* configuration (fig. ??).

There is *UTM* service for given *airspace cluster* calculating *collision cases* (tab. ??) based on incoming *UAS position notifications* (tab. ??).

Simulation Run: Notable moments from the *simulation run* (fig. 7.29) are the following:

1. *Collision Case creation* (fig. 7.29a) following events happens in this step:
 - a. Two *UAS* are approaching *airway intersection*: UAS 1 (blue) from left and UAS 2 (cyan) from the bottom.
 - b. They are going to *collide* at point $\mathcal{C} = [20, 20, 0]^T$ of *Flight Level* (elevation is 45, 000 feet Above Mean Sea Level).
 - c. UTM service notices future *Collision Situation* and creates *Collision Case*.
 - d. *Converging Directive* for 8 m from *Collision point* is issued for UAS 1 (blue) because UAS 2 (cyan) has the *Right Of the Way*.
 - e. *Keep Velocity/Heading Directive* is issued for UAS 2 (cyan) to ensure avoidance maneuver success.
 - f. UAS 1 (blue) corrects its heading according to *UTM* directive.
 - g. UAS 2 (cyan) stays on claimed course and if its necessary adjust its speed.

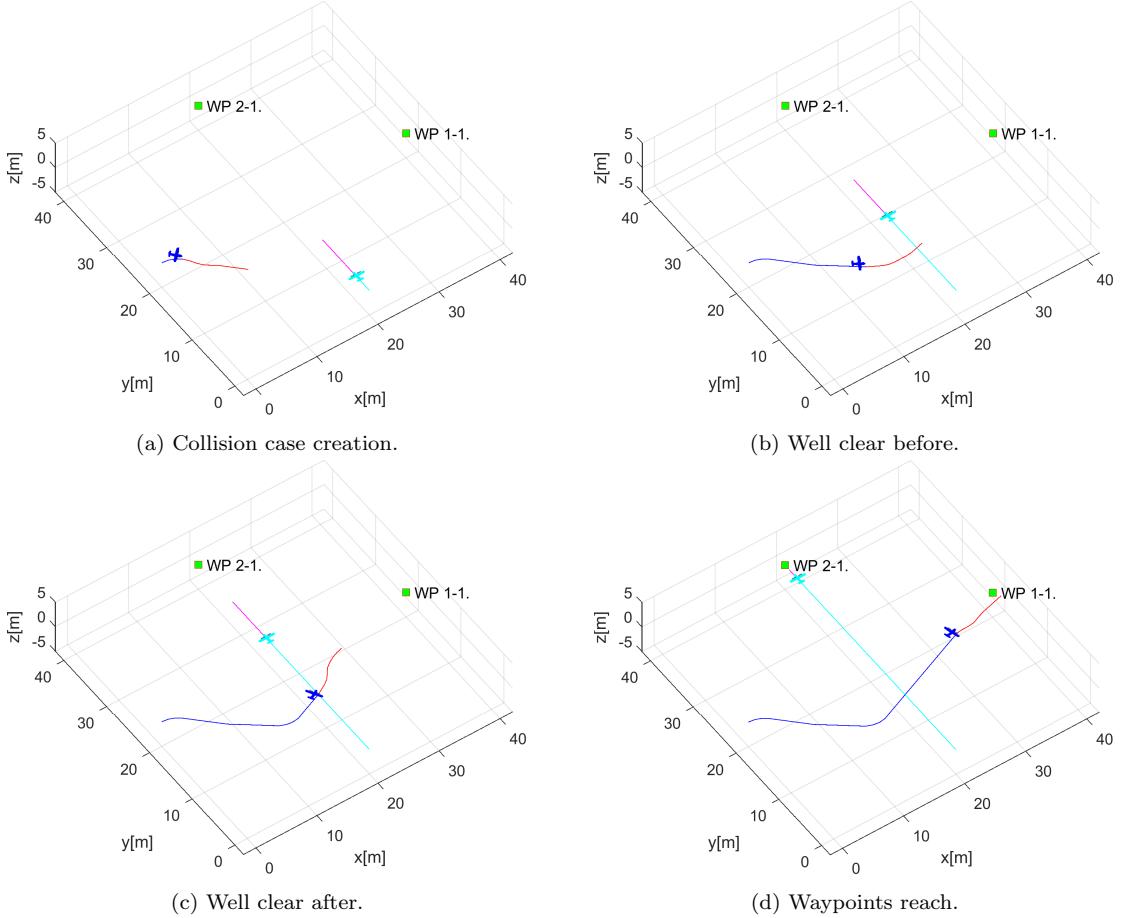


Figure 7.29: Test scenario for *Rule-based converging*.

2. *Well clear before* (fig. 7.29b) UAS 1 (blue) check the *Collision Point* distance and keeps safe distance given by safety margin. UAS 2 (cyan) checks if there is no intruder in *Avoidance Grid* and if not, stays in *Navigation Mode*.
3. *Well clear after* (fig. 7.29c) UAS 2 (cyan) is *after Collision Point*, it can start negotiations of new speed and heading with UTM. UAS 1 (blue) is still enforced to follow *Converging Maneuver* directive until the outer boundary of *Collision Zone* is reached.

4. *Waypoints reach* (fig. 7.29d) UAS 1 (blue) leaves the outer boundary of the *Collision zone*. Leaving *Converging Maneuver Directive*. UTM closes *Collision Case*.

Collision Case Calculation: For test scenario in (fig. 7.29) where UAS 1 (blue) is converging to avoid UAS 2 (cyan) the *Collision Case* (tab. 7.36) have been calculated.

The *Collision point* is at $[20, 20, 0]$ in *Flight Level FL450* coordinate frame.

The *angle of approach* was evaluated as 90° which indicates *converging maneuver* in range $70^\circ \leq \text{angleOfApproach} < 130^\circ$.

The *mutual position* of UAS 1 (blue) and UAS 2(cyan) is giving the roles: *Right Of the Way* for UAS 2 (cyan) and *Converging* for UAS 1 (blue).

The *safety margin* for *Well Clear* was determined as $3m$ for UAS 1 and $5m$ for UAS 2. (Note: Well Clear Margin is usually much greater than Near Miss margin). The *Combined Case* margin which was enforced was $8m$. The mutual distance cannot go below this threshold.

Collision Case						Margins	
id	UAS	role	collision point	angle of approach	type	safety	case
1-2	1	Converging	$[20, 20, 0]^T$	90°	Converging	3	8
	2	Right o. W.				5	

Table 7.36: Collision case for *Rule-based converging scenario*.

Distance to Safety Margin Evolution: The safety margin values (*well clear*) (fig. 7.30) in controlled airspace are much greater than in non-controlled airspace (*near miss*) (fig. 7.18)

The enforced rule was (rule ??) with parameters: Collision Point $[20, 20, 0]^T$ and *Safety Margin* 8 m as given by Collision Case (tab. 7.36).

The mutual *UAS distance* (blue line) does not go over *Safety Margin* (red line), which means UAS 1 well clear margin of 3 m and UAS 2 well clear margin of 5 m are not broken (fig. 7.30).

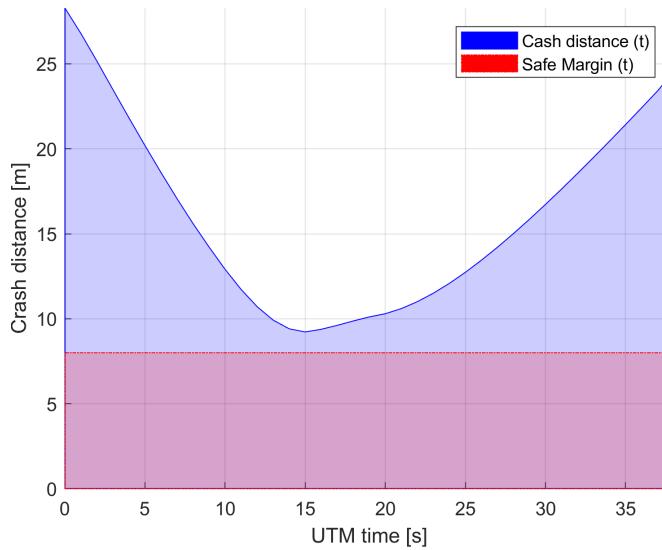


Figure 7.30: Distance to safety margin evolution for the *rule-based converging scenario*.

Distance to Safety Margin Peaks: *Distance to safety margin peaks* (tab. 7.37) represent the proximity on UAS mutual distance to *breach of well clear condition* (safety margin). The

breach of well clear condition was not achieved. The *minimal distance to the safety margin* was 1.2240 m. The *maximal distance to safety margin* was 20.2843 m which represent distance in a time of *Collision Case Creation*.

UAS:	Distance to Safety Margin		
	min	max	breach
1-2	1.2240	20.2843	false

Table 7.37: Distance to safety margin peaks for the *rule-based converging scenario*.

Path Tracking Performance: *Path tracking* is displayed in (fig. 7.31). The *UAS* trajectory is divided into *X, Y, Z axis tracking over UTM Time*. The *Reference Trajectory* (green dashed line) interconnect starting position of UAS (green square marked S) a goal waypoint (green square marked 1). The *Executed Trajectory* (solid blue line) reflects real UAS trajectory.

1. UAS 1. (fig, 7.31a) do steady right side *converging maneuver* (y-axis).
2. UAS 2. (fig. 7.31b) follows the reference trajectory precisely because it has the *Right Of Way*.

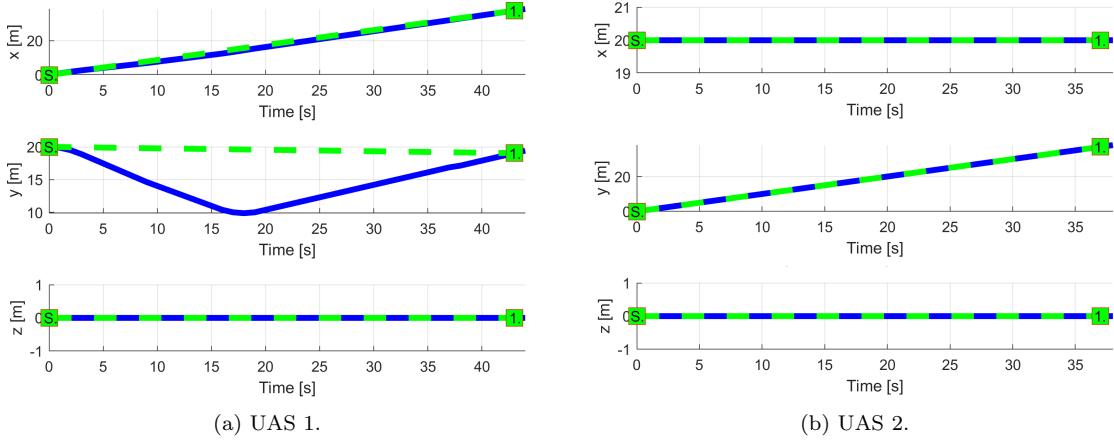


Figure 7.31: *Trajectory tracking* for *Rule-based converging* test case.

Path Tracking Deviations: Deviations (tab. 7.38) are in *expected ranges*, considering the *mission plans* (tab. 7.35) and *Collision Case* safety margin of 8m.

The minimal deviation distance was expected at the value of *safety margin* (8m). The maximal deviation was 10.22m which is acceptable due the space discretization, UAS dynamic, and, *dynamic decision time*.

Param.	UAS 1 UAS 2	
	\mathcal{WP}_1	\mathcal{WP}_1
$\max x $	0	0
$\max y $	10.22	0
$\max z $	0	0
$\max dist.$	10.22	0

Table 7.38: Path tracking properties for *Rule-based converging* scenario.

Computation Load: The *computation load* for scenario (fig.7.32) shows used time (y-axis) over decision frame (x-axis).

The *computation time* is slightly increased for avoiding UAS 1 during avoidance. The initial increase of computation time UAS 2 is caused by UTM communication demand.

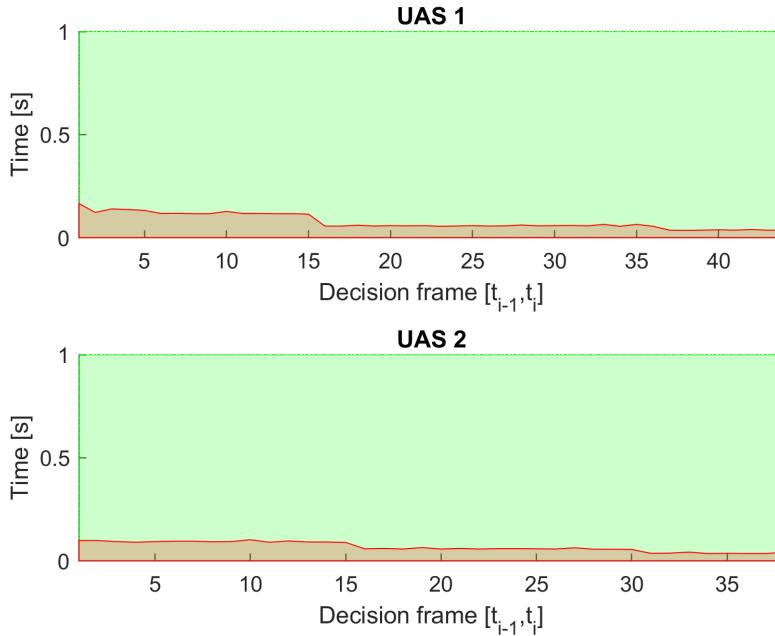


Figure 7.32: Computation time for *Rule-based converging* scenario.

7.4.2 Rule-Based Head-On

Scenario: Two *UAS* are going on the same *airway* in same *flight level* in the opposite direction in *controlled airspace* (over 500 feet Above the Ground Level). The *mutual position* of UAS can be classified as *Side Approach*. Following *collision hazard* is present:

1. *Head-on Collision Hazard* - An *UAS* is approaching from opposite direction which invokes need to avoid *Collision Point* actively.

Head on Collision Hazard must be addressed by *UTM* service in the following manner:

1. *Each UAS* in particular *Controlled Space* periodically sends synchronized *Position Notification* messages (tab. ??).
2. *UTM* service receives *Position Notifications* and manages *Collision Cases* (tab. ??) in *Controlled Space*.
3. *UTM* detects single *Head on Collision Cases* with *Collision Point* in the vicinity.
4. *UTM* service creates *Virtual Roundabout* and implements *Normative Directive* on both *UAS*.

Mission parameters for four UAS systems are defined in (tab. 7.39).

UAS	Position		\mathcal{WP}_1
	$[x, y, z]$	$[\theta, \varpi, \psi]$	
1	$[0, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[45, 20, 0]^T$
2	$[40, 20, 0]^T$	$[0^\circ, 0^\circ, 180^\circ]^T$	$[-5, 20, 0]^T$

Table 7.39: Mission setup for *Rule-based head-on* scenario.

Assumptions: Following assumptions are valid for this test:

1. *Controlled Airspace Airworthiness* - UAS system is equipped with necessary controlled airspace equipment like ADS-B In/Out, Radar, Transponder, etc. Moreover, airworthy *UAS* can precisely follow *UTM directives* (max. 5 % deviation).
2. *C2 (Command & control) Link Established* - necessary for $(\text{UAS} \leftrightarrow \text{UAS})$ and $(\text{UAS} \leftrightarrow \text{UTM})$ communication. If *C2* link is lost the *UAS* will enter into *Emergency avoidance mode*.
3. *Decision frame synchronization with UTM* - necessary in discrete *C2* environment otherwise *safety margins* needs to be *bloated*.
4. *Both UAS have identical cruising speed* - simplification impacting *UTM* service implementation. *Obstacle Avoidance Framework* can comprehend various intruders speed, with proper *UAS* directives.

Main Goal: Show possibility of *Head-on situation resolution* with *forced safety margin* by *UAS Traffic Management* system. The *Obstacle Avoidance Framework based on Reach Sets* is used as a *Navigation Module*.

Acceptance Criteria: Following criteria must be met:

1. *Well Clear Condition valid for both UAS* - Both *UAS* must have *minimal required distance* from *other UAS* for all *Virtual Roundabout* enforcement time.
2. *Fulfillment of UTM Directives* - Both *UAS* must stay in a *Navigation mode* for all *Virtual Roundabout* enforcement time. Both *UAS* must stay on *Virtual Roundabout* for the necessary time, before leaving for *Original Navigation waypoint* \mathcal{WP}_1 .

Testing Setup: The *standard test setup* for each *UAS* defined in (tab. 7.2, 7.3, 7.4, 7.5, 7.6) is used with following parameter override:

1. *Navigation grid - type* - *ACAS-like* with *horizontal enabled maneuvers*.

This *configuration* is based on the assumption that both *UAS* is in *controlled airspace* in *FL450* (flight level 45000 feet Above Sea Level), without permission for a *climb or descent maneuver*. The *rule engine* is initialized in standard *Rules of the air* configuration (fig. ??).

There is *UTM* service for given *airspace cluster* calculating *collision cases* (tab. ??) based on incoming *UAS position notifications* (tab. ??).

Simulation Run: Notable moments from the *simulation run* (fig. 7.33) are the following:

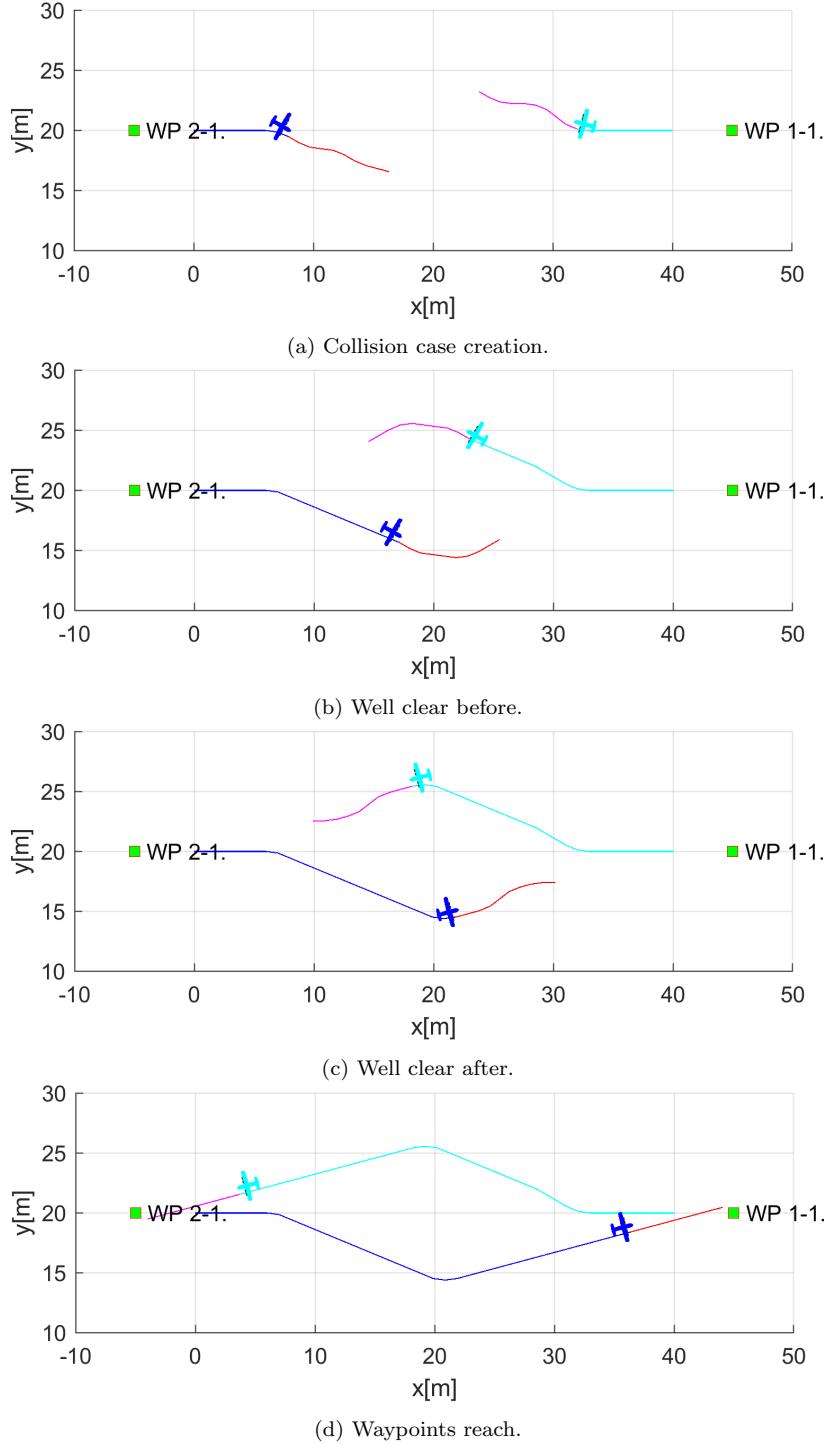


Figure 7.33: Test scenario for the *rule-based head-on approach* (virtual roundabout).

1. *Collision Case creation* (fig. 7.33a) following events happens in this step:
 - a. Two UAS are on the same airway approaching each other from the opposite direction, UAS 1 (blue) from the left, UAS 2 (cyan) from the right.
 - b. They are going to *collide* at point $C = [20, 20, 0]^T$ of *Flight Level* (Elevation is 45, 000 feet Above Mean Sea Level).
 - c. UTM service notices future *Collision Situation* and creates *Collision Case*.
 - d. *Virtual Roundabout* is created at *collision point* with radius 10m. UTM issues directive

for both UAS to avoid collision point from different sides.

- e. UAS 1 (blue) receives a directive to avoid *Collision Point* from the *right side* (downside in GCS). UAS 2 (cyan) receives a directive to avoid *Collision Point* from the *right side* (upside in GCS).
 - f. Both UAS enters into *Virtual Roundabout*.
2. *Well clear before* (fig. 7.33b) UAS 1 (blue) is keeping *enforced safety margin* (10 m) from *collision point* and *UAS 2 position*. The *Virtual Roundabout* is enforced until the (*Collision point*) is reached by both UAS. Both UAS stays in *Navigation Mode*.
 3. *Well clear after* (fig. 7.33c) UTM notices that *Collision point level* has been reached by both UAS. UTM renounce *Directives* and enables a return to *Original Waypoint* WP₁. Both UAS starts to converging to *Original waypoint* (because possible collision was averted).
 4. *Waypoint reach* (fig. 7.33d) Both UAS reaches respective goal points.

Collision Case Calculation: For test scenario in (fig. 7.33) where UAS 1 (blue) have head-on collision with UAS 2 (cyan), *Collision Case* have been calculated (tab. 7.40).

The *Collision point* is at $[20, 20, 0]^T$ in Flight Level *FL450* coordinate frame.

The *angle of approach* was evaluated as 180° which indicates *Head-on Approach* due to the $130^\circ \leq \text{angle of Approach} \leq 180^\circ$ conditions.

The *mutual position* of UAS 1 (blue) and UAS 2 (cyan) is giving the roles of *Roundabout* to *both* UAS.

The *safety margin* for *Well Clear* was determined as 5m for UAS 1 and UAS 2. The combined *Case Margin* is 10 m, which is sum of both. The *mutual distance* cannot go below this threshold.

Collision Case						Margins	
id	UAS	role	collision point	angle of approach	type	safety	case
1-2	1	Roundabout	$[20, 20, 0]^T$	180°	Head on	5	10
	2	Roundabout				5	

Table 7.40: Collision case for the *rule-based head-on* scenario.

Distance to Safety Margin Evolution: The safety margin values (*well clear*) (fig. 7.34) in controlled airspace are much larger than in non-controlled airspace (*near miss*) (fig. 7.22).

The enforced rule was (rule ??) with parameters: Collision Point $[20, 20, 0]^T$ and *Safety Margin* 10 m as given by Collision Case (tab. 7.40).

The mutual *UAS distance* (blue line) does not go over *Safety Margin* (red line) which means both UAS well clear margins are not broken by any means (fig. 7.33).

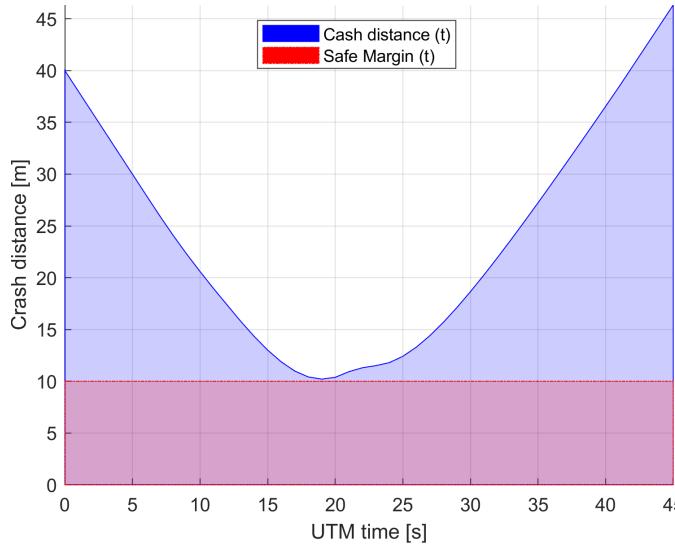


Figure 7.34: Distance to safety margin evolution for the *rule-based head-on scenario*.

Distance to Safety Margin Peaks: Given by (tab. 7.41) represents the proximity on UAS mutual distance to *well clear condition* breach. The breach of *well clear condition* was not achieved. The *minimal distance to the safety margin* was 0.2084 m. The *maximal distance to safety margin* was 36.3253m which represents distance at *Collision Case closing*.

UAS:	Distance to Safety Margin		
	min	max	breach
1-2	0.2084	36.3253	false

Table 7.41: *Rule-based head-on* safety margin distances.

Path Tracking Performance: *Path tracking* is displayed in (fig. 7.35). The *UAS* trajectory is divided into *X, Y, Z axis tracking over UTM Time*. The *Reference Trajectory* (green dashed line) interconnect starting position of UAS (green square marked S) a goal waypoint (green square marked 1). The *Executed Trajectory* (solid blue line) reflects real UAS trajectory.

1. UAS 1. (fig. 7.35a) do steady right side *roundabout maneuver* (y-axis).
2. UAS 2. (fig. 7.35b) do steady right side *roundabout maneuver* (y-axis).

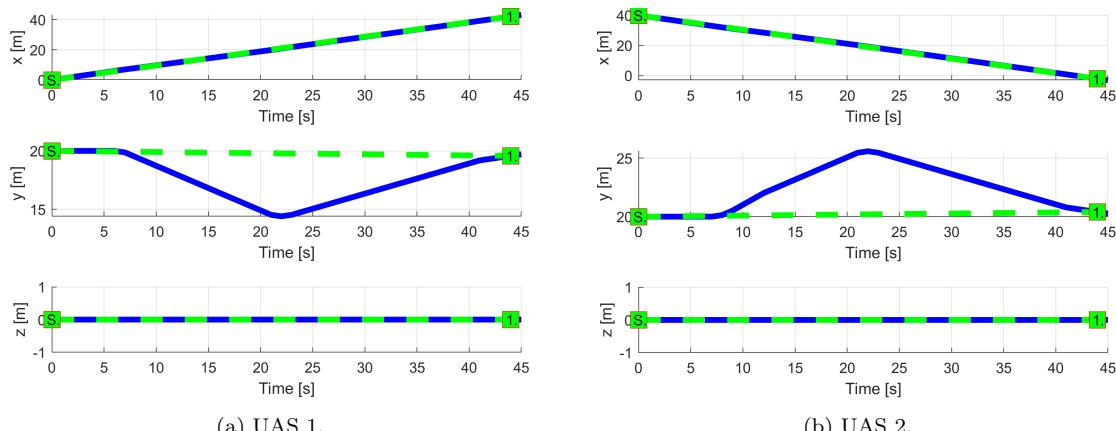


Figure 7.35: *Trajectory tracking* for *rule-based head-on* test case.

Path Tracking Deviations: Deviations (tab. 7.42) are in *expected ranges*, considering the *mission plans* (tab. 7.39) and *Collision Case* safety margin of 10m.

Param.	UAS 1	UAS 2
	\mathcal{WP}_1	\mathcal{WP}_1
$\max x $	0	0
$\max y $	5.40	5.40
$\max z $	0	0
$\max dist.$	5.40	5.40

Table 7.42: Path tracking properties for *rule-based head-on* scenario.

Computation Load: The *computation load* for *scenario* (fig.7.36) shows used time (y-axis) over decision frame (x-axis).

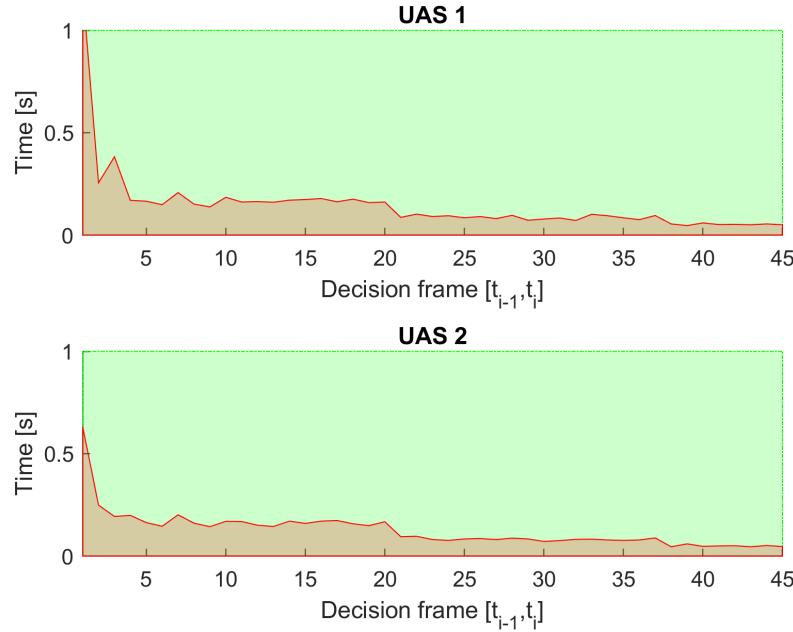


Figure 7.36: Computation time for *rule-based head-on* scenario.

7.4.3 Rule-Based Mixed Head-On with Converging

Scenario: Four *UAS* are approaching an airway *intersection* at the *same time* from *opposite direction* in *controlled airspace* (over 500 feet Above Ground Level). Each *UAS* have following *Collision Hazards*:

1. *Head on Collision Hazard* - An UAS is approaching from opposite direction which invokes need to avoid *Collision Point* actively
2. *Active Converging Collision Hazard* - An UAS is approaching from the *right side*, which gives him *Right of the Way* and invokes the need to avoid *Intruder* actively.
3. *Passive Converging Collision Hazard* - An UAS is approaching from the *left side*, which gave us *Right of the Way* and imposes an obligation of *active avoidance* on other *UAS*.

Note. Presented scenario is *the worst possible situation* in current *manned aviation ATM*.

Mentioned *Collision Hazards* must be addressed by *UTM* service in the following manner:

1. *Each UAS* in particular *Controlled Space* periodically sends synchronized *Position Notification* messages (tab. ??).
2. *UTM* service receives *Position Notifications* and manages *Collision Cases* (tab. ??) in *Controlled Space*.
3. *UTM* detects multiple *Collision Cases* with *Collision Points* in the vicinity.
4. *UTM* service creates *Virtual Roundabout* and implements *Normative Directive* on all *UAS* in the area.

Mission parameters for four UAS systems are defined in (tab. 7.43).

UAS	Position		WP_1
	$[x, y, z]$	$[\theta, \varpi, \psi]$	
1	$[0, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[45, 20, 0]^T$
2	$[40, 20, 0]^T$	$[0^\circ, 0^\circ, 180^\circ]^T$	$[-5, 20, 0]^T$
3	$[20, 0, 0]^T$	$[0^\circ, 0^\circ, 90^\circ]^T$	$[20, 45, 0]^T$
4	$[20, 40, 0]^T$	$[0^\circ, 0^\circ, -90^\circ]^T$	$[45, 20, 0]^T$

Table 7.43: Mission setup for *rule-based mixed scenario*.

Assumptions: Following assumptions are valid for this test:

1. *Controlled Airspace Airworthiness* - UAS system is equipped with necessary controlled airspace equipment like ADS-B In/Out, Radar, Transponder, etc. Moreover, airworthy *UAS* has capability to precisely follow *UTM directives* (max. 5 % deviation).
2. *C2 (Command & control) Link Established* - necessary for $(UAS \leftrightarrow UAS)$ and $(UAS \leftrightarrow UTM)$ communication. If *C2* link is lost the *UAS* will enter into *Emergency avoidance mode*.
3. *Decision frame synchronization with UTM* - necessary in discrete C2 environment otherwise *safety margins* needs to be *bloated*.
4. *Every UAS have identical cruising speed* - simplification impacting *UTM* service implementation. *Obstacle Avoidance Framework* can comprehend various intruders speed, with proper *UAS* directives.

Main Goal: Show possibility of *Virtual Roundabout* invoked by *UTM* directives where *Obstacle Avoidance Framework based on Reach Sets* is used as a *Navigation Module*.

Acceptance Criteria: Following criteria must be met:

1. *Well Clear Condition valid for every UAS* - Each *UAS* must have *minimal required distance* from *other UAS* for all *Virtual Roundabout* enforcement time.
2. *Fulfillment of UTM Directives* - Each UAS must stay in a *Navigation mode* for all *Virtual Roundabout* enforcement time. Each *UAS* must stay on *Virtual Roundabout* for the necessary time, before leaving for *Original Navigation waypoint WP_1* .

Testing Setup: The *standard test setup* for each UAS defined in (tab. 7.2, 7.3, 7.4, 7.5, 7.6) is used with following parameter override:

1. *Navigation grid - type - ACAS-like with horizontal enabled maneuvers*

This *configuration* is based on the assumption that every UAS is in *controlled airspace* in *FL450* (flight level 45000 feet Above Sea Level), without permission for a *climb or descent maneuver*. The *rule engine* is initialized in standard *Rules of the air* configuration (fig. ??).

There is *UTM* service for given *airspace cluster* calculating *collision cases* (tab. ??) based on incoming *UAS position notifications* (tab. ??).

Simulation Run: Notable moments from the *simulation run* (fig. 7.37) are the following:

1. *Collision cases created* (fig. 7.37a) following events happen in this step:
 - a. Four *UAS* are approaching airways intersection: *UAS 1* (blue) from left, *UAS 2* (cyan) from right, *UAS 3* (green) from the bottom, *UAS 4* (black) from the top.
 - b. They are going to collide at point $[20, 20, 0]^T$ of *Flight level* (elevation is 45, 000 feet Above Mean Sea Level).
 - c. *UTM service* notices future *Collision Situations* and creates *Collision Cases*.
 - d. There are many *Collision Cases* in the near vicinity. The *Virtual Roundabout* is created with *Safety margin* 15 m.
 - e. The *UTM service* then sends a new *Roundabout Directives* to involved *UAS* systems.
 - f. Each *UAS* starts *Roundabout Entry Maneuver* by correcting own *Heading* and *Speed* (if its necessary).
2. *Roundabout entry* (fig. 7.37b) - Each *UAS* enters into *Virtual Roundabout* while sending *Roundabout Entrance Notification* to *UTM service*.
3. *Roundabout leave* (fig. 7.37c) following events happens in this step:
 - a. Each *UAS* when is going to approach the level of *Original Goal Waypoint* sends *Roundabout Leave Request*.
 - b. *UTM system* will check if there is *Sufficient Free Space* to leave *Virtual Roundabout*.
 - c. The *UTM Service* then issues *Virtual Roundabout Leave Approval*.
 - d. Each *UAS* will correct own heading and speed in the range of received permit.
4. *Situation resolution* (fig. 7.37d) - Each *UAS* is heading away from *Roundabout Center*, there is no active user of *Virtual Roundabout*. *UTM* will remove *Virtual Roundabout* and closes underlying *Collision Cases*. Each *UAS* will reach respective *Original Goal Waypoint*.

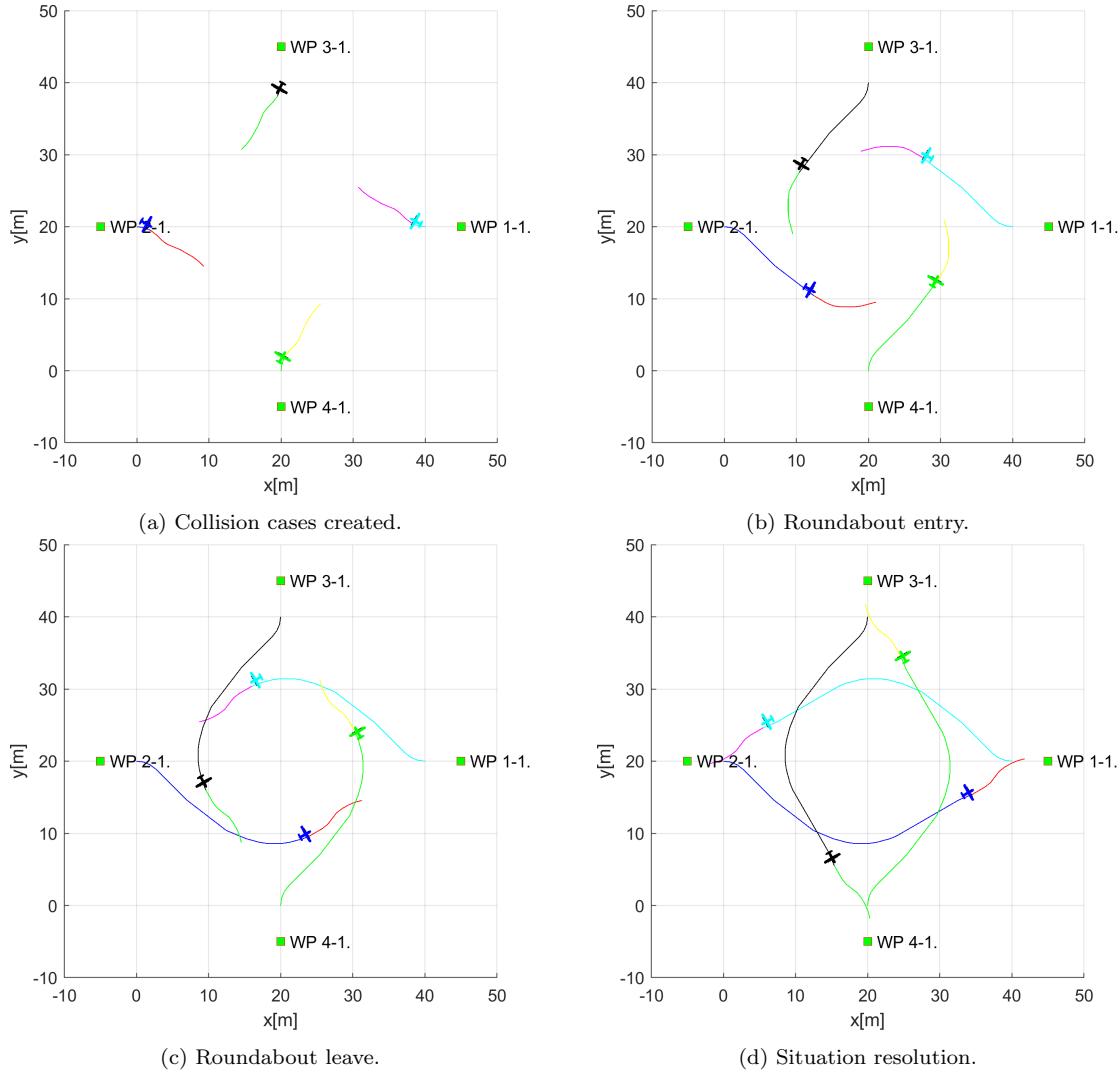


Figure 7.37: Test scenario for *rule-based mixed* situation with the *self-separation mode*.

Collision Cases Calculation: The set of original *Collision cases* is given in (tab. 7.44).

Each *UAS* has one *Head on*, *Converging passive*, *Converging active* collision hazard. For example *UAS 1* have a *head-on* with *UAS 2*, *converging passive* with *UAS 4*, *converging active* with *UAS 3*. For *UAS 2-4* check *role* in respective *Collision Cases*.

Note. *Collision cases* calculated by *UTM* are symmetric, which means that collision case for *UAS X*, *UAS Y* is identical to collision case calculated for *UAS Y*, *UAS X*, $X \neq Y$.

Safety margin representing *Well Clear Margin* for single *UAS* in *Collision Case* ranges 5 – 8 m. *Case margin* representing the minimal mutual distance between two *UAS systems* to remain well clear ranges 12 – 15 m.

Merged Collision Case is oversimplified for demonstration purposes. *Merge Case Procedure* is out of the scope of this work due to its extent. Every *Collision Case* shares same *Collision Point* $[20, 20, 0]^T$ in flight level coordinate frame. *Merged Collision Case* type was set as *Roundabout*, due the number of collision case *attendants* is greater than 2. Each *UAS role* has been set as *Roundabout*. The enforced *safety margin* is equal to 15 m, which is the maximum of all *single collision case combined margins*.

Collision Case						Margins	
id	UAS	role	collision point	angle of approach	type	safety	case
1-2	1	Roundabout	$[20, 20, 0]^T$	180°	Head on	8	15
	2	Roundabout				7	
1-3	1	Converging	$[20, 20, 0]^T$	90°	Converging	8	15
	3	Right o.W.				5	
1-4	1	Right o.W.	$[20, 20, 0]^T$	90°	Converging	8	15
	4	Converging				5	
2-3	2	Right o.W.	$[20, 20, 0]^T$	90°	Converging	7	12
	3	Converging				5	
2-4	2	Converging	$[20, 20, 0]^T$	90°	Converging	7	12
	4	Right o.W.				5	
3-4	3	Roundabout	$[20, 20, 0]^T$	180°	Head on	7	14
	4	Roundabout				7	
Merged cases						Safety	
id	UAS	role	collision point	type		Margin	
1-2- -3-4	1	Roundabout	$[20, 20, 0]^T$	Roundabout		15	
	2	Roundabout					
	3	Roundabout					
	4	Roundabout					

Table 7.44: Collision cases for *rule-based mixed* scenario.

Distance to Safety Margin Evolution: *Merged Collision Case Safety Margin* is 15 m , and it is valid for all *UAS mutual distances*. The simple condition for *Remain Well Clear* is:

$$\text{crashDistance}(\text{UAS}_X, \text{UAS}_Y, t) \geq 15\text{ m}, X \neq Y \in \{1, 2, 3, 4\}, t \in \text{utmTime}$$

Safety Margin Performance is given in (fig. 7.38). The mutual distance (Crash Distance [m]) between two UAS is denoted as the *blue line*. The enforced safety margin for *Remain Well Clear* condition is denoted as the red line.

Note. *Evolution of mutual crash distance* is symmetric. In any case, the mutual distance goes under *safety margin*. *Acceptance criterion* for *Well Clear condition* is fulfilled.

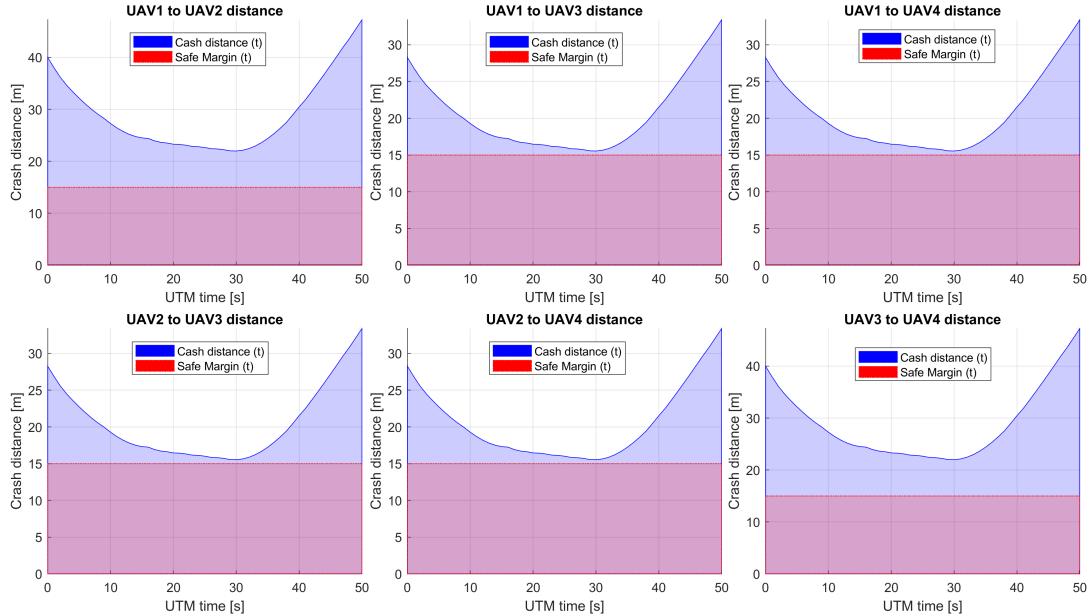


Figure 7.38: Distance to safety margin evolution for *rule-based mixed scenario*.

Distance to Safety Margin Peaks: *Distance to Safety Margin Peaks* (tab. 7.45) represents the proximity of *UAS mutual distance to breach well clear condition*. The *breach condition* was not fulfilled in any combination.

The *minimal distance to safety margin* was 0.5438 m between all four *UAS* systems. The *maximal distance to safety margin* ranges between $18 - 32\text{ m}$ which show advantages of the *virtual roundabout*.

UAS:	Distance to Safety Margin		
	min	max	breach
1-2	6.9823	32.2369	false
1-3	0.5438	18.4015	false
1-4	0.5438	18.4015	false
2-3	0.5438	18.4015	false
2-4	0.5438	18.4015	false
3-4	6.9823	32.2369	false

Table 7.45: Distance to safety margin peaks for *rule-based mixed scenario*.

Path Tracking Performance: Path tracking is displayed in (fig. 7.39). The UAS trajectory is divided into *X*, *Y*, *Z* axis tracking over *UTM Time*. The *Reference Trajectory* (green dashed line) is represented as the interconnection between *Start Waypoint* (green square marked S) and *Goal Waypoint* WP_1 (green square marked 1). The *Executed trajectory* (solid blue line) reflects real *UAS* movement.

1. *UAS 1* (fig. 7.39a) is using the bottom portion of *Virtual Roundabout* (-Y values), sticking to the boundary of the *Virtual Roundabout*.
2. *UAS 2* (fig. 7.39b) is using the upper portion of the *Virtual Roundabout*. (+Y values), sticking to the boundary of the *Virtual Roundabout*.
3. *UAS 3* (fig. 7.39c) is using the right portion of the *Virtual Roundabout*. (+X values), sticking to the boundary of the *Virtual Roundabout*.
4. *UAS 4* (fig. 7.39d) is using the left portion of the *Virtual Roundabout*. (-X values), sticking to the boundary of the *Virtual Roundabout*.

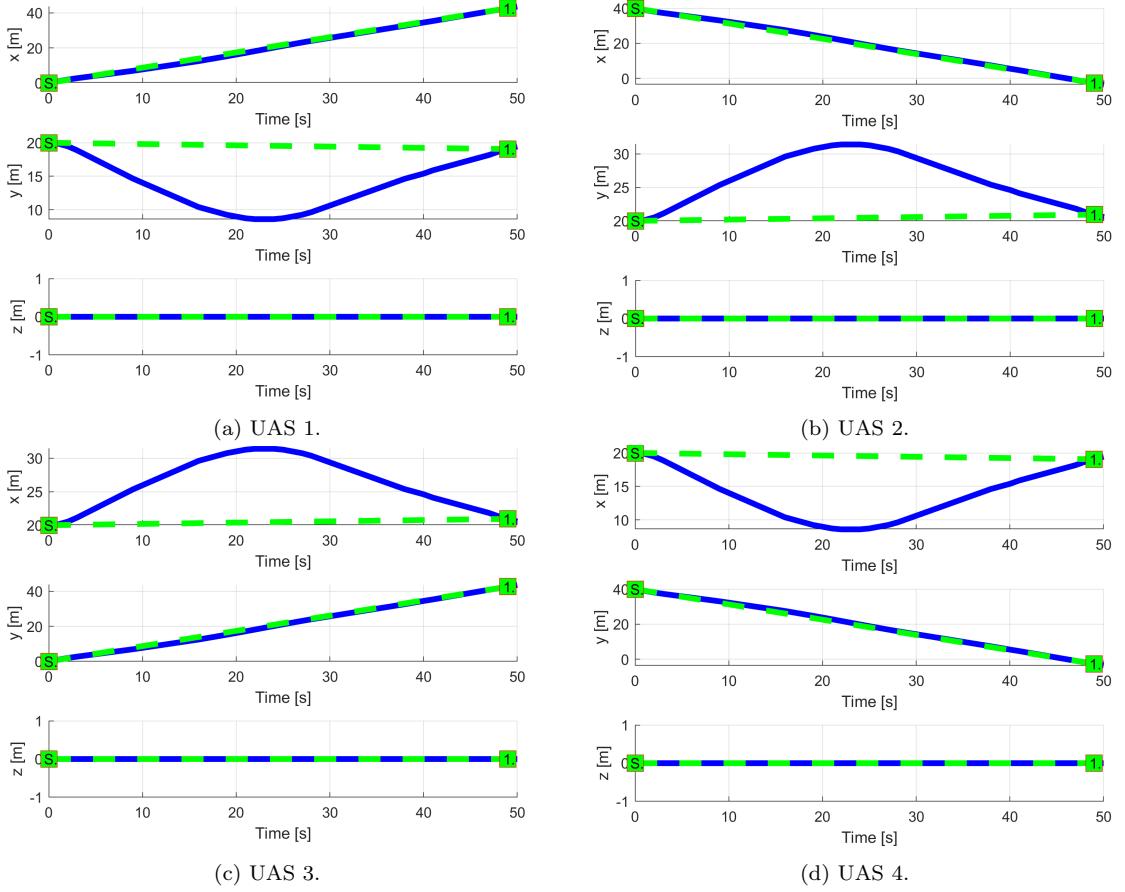


Figure 7.39: Trajectory tracking for *rule-based mixed situation test case*.

Path Tracking Deviations: *Deviations* (tab. 7.46) are in expected ranges, considering the mission plans (tab. 7.43) and *Merged Case Safety Margin* (15 m).

Param.	UAS 1	UAS 2	UAS 3	UAS 4
	WP_1	WP_1	WP_1	WP_1
max $ x $	0	0	11.40	11.40
max $ y $	11.40	11.40	0	0
max $ z $	0	0	0	0
max dist.	11.40	11.40	11.40	11.40

Table 7.46: Path tracking properties for *rule-based mixed* scenario.

Computation Load: The *computation load* for *scenario* (fig.7.40) shows used time (y-axis) over decision frame (x-axis).

The *computation time* for each UAS has the same evolution. The *load* is higher during avoidance maneuver on the *virtual roundabout*.

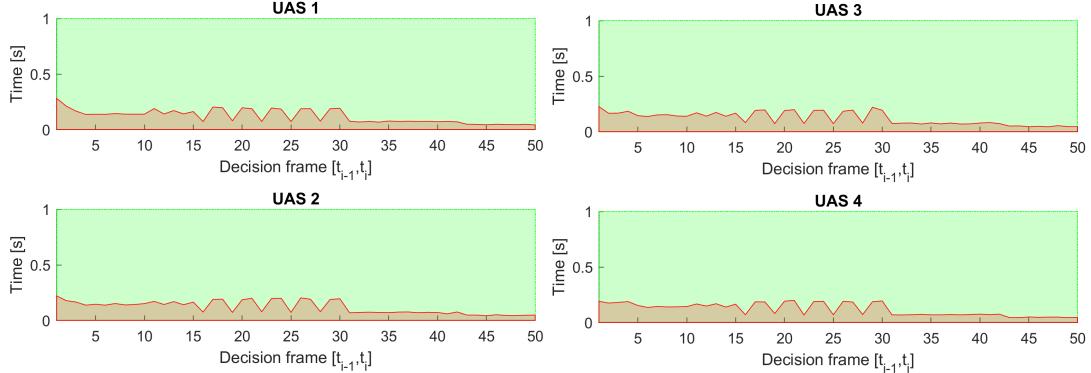


Figure 7.40: Computation time for *rule-based multiple* scenario.

7.4.4 Rule-Based Overtake

Scenario: Two UAS are flying in the *controlled airspace* (over 500 feet Above Ground Level) on the *airway* (in the same direction). *Slower UAS* is in front of *Faster UAS*. There is possibility of a *collision* or a *near miss incident* or a *well clear breach*. The *Faster UAS* (Overtaking) must contact *UTM service* and ask for *overtake permission*. Scenario steps:

1. *Faster UAS* (Overtaking) notices *UTM service* about *Slower UAS* (Overtaken). (This step is Optional.)
2. *UTM service* issues *Directives* to all *UAS* in the area.
3. *Overtake Directive* is received by *Faster UAS* (Overtaking) and *Slower UAS* (Overtaken).
4. *Faster UAS* (Overtaking) mission plan is altered to reflect *Overtake directive*, *Divergence Waypoint* and *Convergence Waypoint* are added.
5. *Faster UAS* (Overtaking) safely overtakes *Slower UAS* (Overtaken) without breaking *Well clear* condition.

Mission parameters for both *UAS* systems are defined in (tab. 7.47).

UAS	Position		\mathcal{WP}_1
	$[x, y, z]$	$[\theta, \varpi, \psi]$	
1	$[-40, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[110, 20, 0]^T$
2	$[-20, 20, 0]^T$	$[0^\circ, 0^\circ, 0^\circ]^T$	$[80, 20, 0]^T$

Table 7.47: Mission setup for all *Rule based overtaking* scenarios.

Assumptions: Following assumptions are valid for this test:

1. *Controlled Airspace Airworthiness* - UAS system is equipped with necessary controlled airspace equipment like ADS-B In/Out, Radar, Transponder, etc. Moreover, airworthy *UAS* has capability to precisely follow *UTM directives* (max. 5 % deviation).
2. *C2 (Command & control) Link Established* - necessary for $(\text{UAS} \leftrightarrow \text{UAS})$ and $(\text{UAS} \leftrightarrow \text{UTM})$ communication. If *C2* link is lost the *UAS* will enter into *Emergency avoidance mode*.
3. *Decision frame synchronization with UTM* - necessary in discrete C2 environment otherwise *safety margins* needs to be *bloated*.

Main Goal: Show possibility of *Overtake Maneuver* invoked by the *UTM Directive* (event-based flight constraint).

Acceptance Criteria: Following criteria must be met:

1. *Proper passing of Divergence/Convergence Waypoint* - a minimal distance of *UAS trajectory* to *Divergence/Convergence waypoint* must be below the passing threshold. Waypoints need to be passed in given order (Divergence 1st, Convergence 2nd).
2. *Slower UAS (Overtaken) keeps Right of the Way* - the UAS with lesser maneuverability does not stand a chance in avoidance situation, it needs to keep its *Right of the Way*.
3. *Both UAS does not breach Well Clear (safety) Margin* - mutual distance does not get through *calculated Safety Margin*.

Testing Setup: The *standard test setup* for each UAS defined in (tab. 7.2, 7.3, 7.4, 7.5, 7.6) is used with following parameter override:

1. *Navigation grid - type - ACAS-like with horizontal enabled maneuvers*.

This *configuration* is based on the assumption that every UAS is in *controlled airspace* in *FL450* (flight level 45000 feet Above Sea Level), without permission for a *climb or descent maneuver*. The *rule engine* is initialized in standard *Rules of the air* configuration (fig. ??).

There is *UTM* service for given *airspace cluster* calculating *collision cases* (tab. ??) based on incoming *UAS position notifications* (tab. ??).

Simulation Run: Notable moments from the *simulation run* (fig. 7.41) are the following:

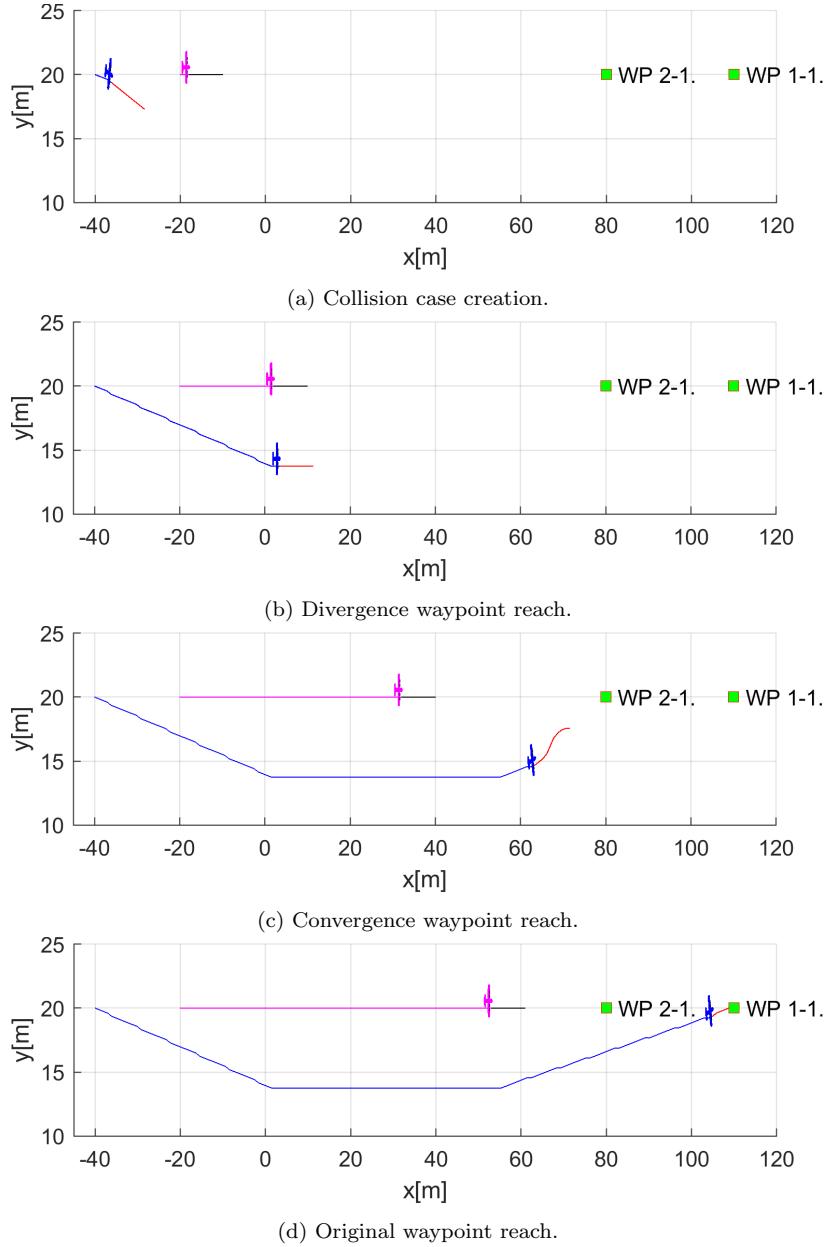


Figure 7.41: Test scenario for *rule-based Overtake* (double speed of overtaking aircraft).

1. *Collision case creation* (fig. 7.41a) - *Faster UAS* (blue) receives *UTM Directive* to invoke *Overtake Rule* (tab. ??). *Slower UAS* (magenta) receives *UTM Directive* to keep *Right of the Way* and warning that is going to be *Overtaken*. *Faster UAS* (blue) creates two *virtual waypoints*:
 - a. *Divergence waypoint* at position $[0, 14, 0]^T$.
 - b. *Convergence waypoint* at position $[24, 14, 0]^T$.

Faster UAS then sets *Divergence waypoint* as *Goal waypoint*, and It starts to overtake maneuver while checking mutual distance.
2. *Divergence waypoint reach* (fig. 7.41b) - *Faster UAS* (blue) successfully reached *Divergence Waypoint*, setting *Convergence Waypoint* as new *Goal waypoint*.

3. *Convergence waypoint reach* (fig. 7.41c) - *Faster UAS* (blue) successfully reached *Convergence Waypoint*, setting *Original Goal Waypoint* as new *Goal waypoint*. The *UTM service* is notified from *Faster UAS* (blue) that *Overtaken Maneuver* has been completed. *UTM acknowledges* maneuver competition and It sends a notification to *Slower UAS* (magenta) that *Overtake Maneuver* is finished. *Slower UAS* (magenta) was successfully overtaken.
4. *Original waypoint reach* (fig. 7.41d) - *Faster UAS* (blue) successfully reached *Original Waypoint*, Starting landing Sequence.

Collision Case Calculation: The *Collision Case* (tab. 7.48) was calculated according to the *Collision Calculation process* (sec. ??). *Faster UAS* (1) has *Overtaking* role, and *Slower UAS* has the *Right of Way*. *Collision Point* is direct type at $[0.20.0]^T$. *Collision case type* was set based on *angle of approach* 0° as *Overtake*. The *Safety Margin* was set as 5 m .

Collision Case						Margins	
id	UAS	role	collision point	angle of approach	type	safety	case
1-2	1	Overtaking	$[0, 20, 0]^T$	0°	Overtake	5	5
	2	Right o.W.				5	

Table 7.48: Collision case for *Rule-based Overtake* scenario 2x speed.

Overtake Speed: Divergence/Convergence Waypoints *Divergence waypoints* have been calculated according to (eq. ??), and, *Convergence Waypoints* have been calculated according to (eq. ??). Following *Speed Differences* were taken into account (Faster/Slower UAS speed ratio): $2x$, $3x$, $4x$. Following observations can be made:

1. *The distance between Divergence and Convergence waypoint* is decreasing with increasing *speed difference*.
2. *Divergence waypoint* is moving *back/right* in *UAS Local Coordinate Frame* with Increasing *speed difference*.
3. *Convergence waypoint* is moving like *Divergence waypoint* but a little bit faster.

Speed diff.	Divergence		Convergence		Final waypoint
	waypoint	difference	waypoint	difference	
2x	$[0, 14, 0]^T$		$[24, 14, 0]^T$		$[110, 20, 0]^T$
3x	$[-10, 13, 0]^T$	$[-10, -1, 0]^T$	$[16, 13, 0]^T$	$[-8, -1, 0]^T$	$[110, 20, 0]^T$
4x	$[-13.4, 12, 0]^T$	$[-3.4, -1, 0]^T$	$[14.7, 12, 0]^T$	$[-1.3, -1, 0]^T$	$[110, 20, 0]^T$

Table 7.49: Convergence and divergence waypoints for various speed differences.

Overtake Speed: Impact on Trajectory Overtake speed difference is visible in (fig. 7.42). The *Slower vehicle trajectory*(cyan) is following *standard mission waypoints*. The *Faster vehicle trajectory* for 2x (blue), 3x (green), 4x (black) are following *Divergence/Convergence waypoints* from (tab. 7.49).

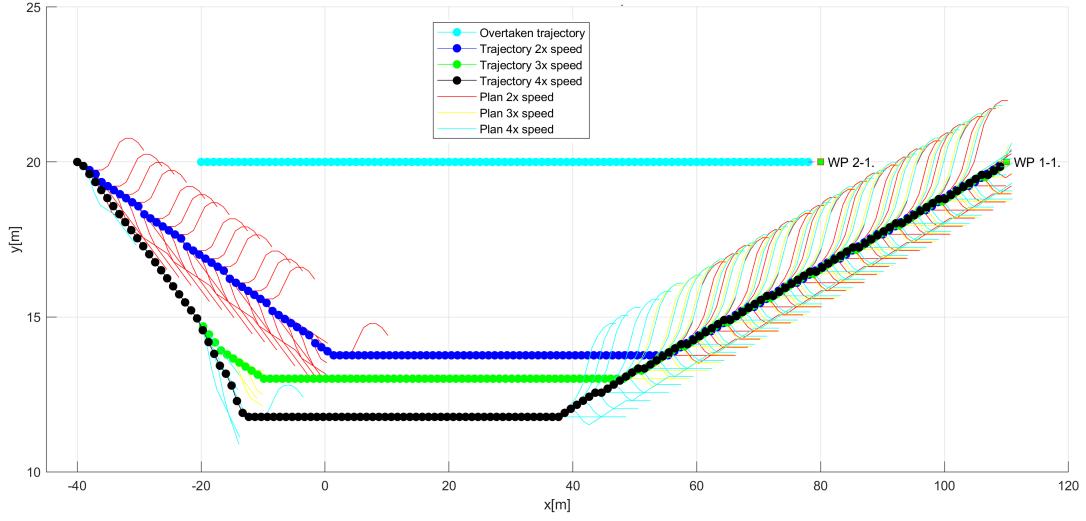


Figure 7.42: Rule-based overtake trajectories at a different speed.

Overtake Speed: Impact on Distance to Safety Margin Evolution *Safety margin* (red line) is set to 5 m. It is obvious that *Faster UAS* will take down *Slower UAS* if there was not for an *Overtake maneuver*. The distance of *Faster UAS* to *Slower UAS* evolution is depending on *Speed difference*. *Inflection point* (closest point of two UAS) is reached sooner with *Higher speed*. *Safety margin performance* was measured for the *UTM performance time* in the interval [0, 35] s and *Speed difference* of 2x (blue), 3x (green), 4x (black).

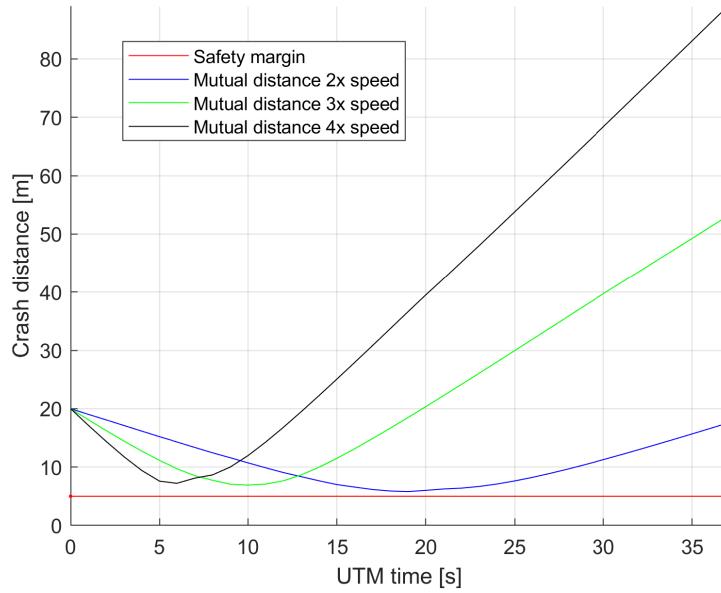


Figure 7.43: Overtake speed-dependent distance to safety margin evolution for rule-based overtake scenario.

Overtake Speed: Impact on Distance to Safety Margin Peaks There is summary table (tab. 7.50) for measurement of minimal and maximal values for *Distance to Safety Margin* over *UTM time* (fig.7.43). The minimal *Overtake Distance to Safety Margin* in 0.7991 m for 2x

Speed Difference. The minimal *Overtake closest point reach time* is 7 s for 4x *Speed Difference*.

For each *Speed difference* (2x, 3x, 4x), the *Well Clear Margin* (Safety Margin) was not reached by the *Faster UAS Body boundary*.

Speed diff.	Minimal		Maximal		Breach
	distance	time	distance	time	
2x	0.7991	20	48.8508	76	false
3x	1.9180	11	73.5336	51	false
4x	2.2154	7	84.0721	38	false

Table 7.50: Distance to safety margin peaks for various overtaking speed in *rule-based overtaking scenario*.

Path Tracking Performance: 2x Speed Performance was only evaluated for the case when *Faster/Slower UAS speed ratio* is 2x. All waypoints are marked as green numbered *squares* with a number. Initial waypoint is marked as a green square with *S*. Reference trajectory is annotated as *green dashed line*. The *executed trajectory* is annotated as *solid blue line*.

Following observations can be made from path tracking (fig. 7.44):

1. *UAS 2 has the Right of Way* (fig. 7.44b) - *reference trajectory* and *executed trajectory* are identical.
2. *UAS 1 is Overtaking* (fig. 7.44a) - the following waypoints are marked on reference trajectory:
 - a. *Collision Point* (WP 1.) - this is not used for navigation, it is marking of *Collision Point*.
 - b. *Divergence waypoint* (WP 2.) - there will *Faster UAS* navigate to avoid *Collision*.
 - c. *Convergence waypoint* (WP 3.) - there will *Faster UAS* navigate to gain *Safe Return Distance*.
 - d. *Original Goal Waypoint* (WP 4.) - there will *Faster UAS* continue until *original goal* is reached.

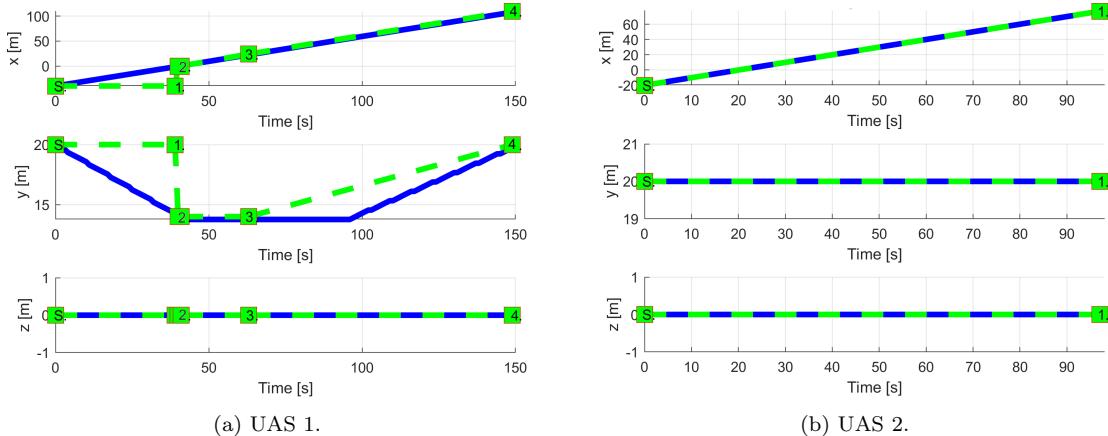


Figure 7.44: Trajectory tracking for *rule-based overtaking double speed* situation test case.

Path Tracking Deviations: 2x Speed Path tracking deviations (tab. 7.51) are interesting for an *Overtake Maneuver* performance.

Maximal deviation distance is for important waypoints: Divergence (WP 2.), Convergence (WP 3.) and Original Goal Waypoint (WP 4.), equal to 0 m. This is the *desired effect* for *Overtake maneuver*.

Collision point (WP 1.) is avoided at minimal distance 5.7991 m (tab. 7.50) and maximal distance 24.5 m (tab. 7.51).

Other *Speed Difference Ratios* yields similar results.

Param.	UAS 1				UAS 2
	WP ₁	WP ₂	WP ₃	WP ₄	WP ₁
	col.	div.	conv.	orig.	nav.
max x	20	0	0	0	0
max y	6	0	4	5	0
max z	0	0	0	0	0
max dist.	24.5	0	4	5	0

Table 7.51: Path tracking properties for *rule overtaking 2x speed scenario*.

Computation Load: The *computation load* for *scenario* (fig.7.45) shows used time (y-axis) over decision frame (x-axis).

The load is minimal on both UAS because the rule calculates only the divergence (eq. ??) and convergence (eq. ??) waypoints.

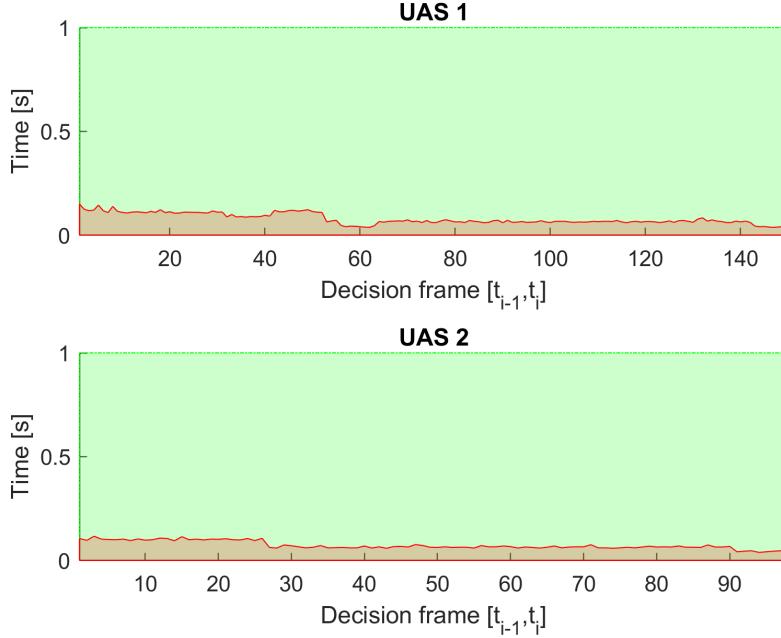


Figure 7.45: Computation time for *rule-based overtaking scenario*.

7.5 Test Cases Conclusion

This section contains summary of performance evaluation (sec. 7.1.3), adversary behavior impact on our approach (sec. 7.5.2), *calculation load* in (sec. 7.5.3).

7.5.1 Performance Evaluation

Performance of test cases was evaluated according to criteria given by (sec. 7.1.3). The performance for *test cases* from test plan (tab. 7.1) has been summarized in (tab. 7.52).

Scenario name	Safety Margin		Breach	Trajectory tracking			Pass / Fail		
	Distance			Waypoint Reach	Reference Deviation	Acceptable Deviation			
	min	max							
Building avoidance (sim. 7.1)	0.69 m UAS 1	24.98 m UAS 1	No (7.2)	Yes/UAS 1/(7.3)	$WP_1 : 107.05m$ $WP_2 : 86.20m$ $WP_3 : 28.70m$ $WP_4 : 32.84m$	Yes (7.10)	Pass		
Slalom (sim. 7.5)	0.09 m UAS 1	3.74 m UAS 1	No (7.6)	Yes/UAS 1/(7.7)	$WP_1 : 20.06m$	Yes (7.14)	Pass		
Maze (sim. 7.9)	0.01 m UAS 1	2.95 m UAS 1	No (7.10)	Yes/UAS 1/(7.11)	$WP_1 : 28.06m$	Yes (7.18)	Pass		
Storm (sim. 7.13)	0.04 m UAS 1	34.99 m UAS 1	No (7.14)	Yes/UAS 1/(7.15)	$WP_1 : 15.76m$	Yes (7.22)	Pass		
Emergency Converging (sim. 7.17)	1.67 m UAS 1-2	27.08 m UAS 1-1	No (7.18)	<u>Yes/UAS 1/(7.19a)</u> <u>Yes/UAS 2/(7.19b)</u>	$WP_1 : 3.25m$ $WP_1 : 0.00m$	Yes (7.26)	Pass		
Emergency Head On (sim. 7.21)	0.38 m UAS 1-2	38.00 m UAS 1-2	No (7.22)	<u>Yes/UAS 1/(7.23a)</u> <u>Yes/UAS 2/(7.23b)</u>	$WP_1 : 3.25m$ $WP_1 : 0.00m$	Yes (7.30)	Pass		
Emergency Multiple (sim. 7.25)	0.20 m UAS 2-4	45.46 m UAS 3-4	No (7.26)	<u>Yes/UAS 1/(7.27a)</u> <u>Yes/UAS 2/(7.27b)</u> <u>Yes/UAS 3/(7.27c)</u> <u>Yes/UAS 4/(7.27d)</u>	$WP_1 : 4.84m$ $WP_1 : 1.83m$ $WP_1 : 3.45m$ $WP_1 : 2.05m$	Yes (7.34)	Pass		
Rule-based Converging (sim. 7.29)	1.22 m UAS 1-2	20.28 m UAS 1-2	No (7.37)	<u>Yes/UAS 1/(7.31a)</u> <u>Yes/UAS 2/(7.31b)</u>	$WP_1 : 10.22m$ $WP_1 : 0.00m$	Yes (7.38)	Pass		
Rule-based Head On (sim. 7.33)	0.21 m UAS 1-2	36.33 m UAS 1-2	No (7.34)	<u>Yes/UAS 1/(7.35a)</u> <u>Yes/UAS 2/(7.35b)</u>	$WP_1 : 5.40m$ $WP_1 : 5.40m$	Yes (7.42)	Pass		
Rule-based Multiple (sim. 7.37)	0.54 m UAS 2-3	32.24 m UAS 1-2	No (7.38)	<u>Yes/UAS 1/(7.39a)</u> <u>Yes/UAS 2/(7.39b)</u> <u>Yes/UAS 3/(7.39c)</u> <u>Yes/UAS 4/(7.39d)</u>	$WP_1 : 11.40m$ $WP_1 : 11.40m$ $WP_1 : 11.40m$ $WP_1 : 11.40m$	Yes (7.46)	Pass		
Rule-based Overtake (sim. 7.41)	0.80 m UAS 1-2	48.85 m UAS 1-2	No (7.43)	Yes/UAS 1/(7.44a) <u>Yes/UAS 2/(7.44b)</u>	$WP_1 : 24.00m$ $WP_2 : 0.00m$ $WP_3 : 4.00m$ $WP_4 : 5.00m$ $WP_1 : 0.00m$	Yes (7.51)	Pass		

Table 7.52: Test cases *performance evaluation*.

Highlights: Each *scenario* contains the reference to notable simulation moments and results. The scenarios were grouped according to the *Operational Space* category, and each category is separated by strike line.

Non-cooperative test cases for the Rural/Urban environment:

1. *Static obstacle avoidance* (Building/Slalom/Maze) - the buildings were correctly avoided without security breach; navigation algorithm was sufficient for given scenarios and obstacle density.
2. *Weather avoidance* (Storm) - the moving *storm* have been avoided in both *soft constraint* and *hard constraint* state. The assumption of *early detection/notification* is key in successful weather avoidance.

Non-cooperative test cases for Intruder Avoidance - the key assumptions are early intruder detection in *Avoidance Grid* and *non-adversarial* behavior. Each UAS was running own instance of *Navigation loop* (fig. ??). The summary of test cases is going like follow:

1. *Emergency converging* - both UAS identified correct roles according to the rules of the air. The UAS 2 kept *right of the way*.
2. *Emergency head on* - both UAS identified correct roles according rules of the air, both of them uses full separation with *Combined Reach Set Approximation* (sec. ??).
3. *Emergency mixed* - all four UAS enters into emergency avoidance mode intermediately after intruders detection. The *non-cooperative* consensus of separation is reached (fig. ??)

Cooperative test cases with UTM supervision are working according to *UTM architecture* (fig. ??), where the *UTM* is considered as main authority. The key assumptions are UTM Resolution fulfillment and *non-adversary behavior*. Each UAS was running own instance of *Navigation loop* (fig. ??) with enabled *Rule Engine* (sec. ??). The summary of test cases is going like follow:

1. *Rule-based converging* - correct handling of *converging maneuver* (fig. ??), proper rule invocation (rule ??) on UAS side.
2. *Rule-based head on* - correct handling of *head on maneuver* (fig. ??), proper rule invocation (rule ??) on UAS side.
3. *Rule-based multiple* - proper *Collision case Merge* (tab. 7.44) with new collision point (eq. ??) and *safety margin calculation* (eq. ??).
4. *Rule-based overtake* - correct handling of *overtaking maneuver* (fig. ??), proper rule invocation (rule ??). Divergence/Convergence (eq. ??,??) for multiple waypoints calculation works for various speed difference (fig. 7.43).

7.5.2 Adversary Behaviour Impact

The *abuse* of UAS for *ill intentions* realization is expected. The *UAS* is cheap, disposable and does not have ethic boundaries.

One of the *assumptions* was that there are only intruders who do not actively look to harm our *UAS*. Breaking this assumption can be lethal for our system and also for other systems.

Let us take *Rule-based Head on* test case (sec. 7.4.2), changing only following aspects:

1. *UAS 2 position spoofing* - the adversarial vehicle is *faking its position* according to expected behavior.
2. *UAS 2 Navigation goal* - set as *UAS 1 position* from intercepted *position notifications* (tab. ??).

Simulation: The *simulation* (fig. 7.46) have been run with defined condition. UAS 2 (magenta) has been chosen as the *adversary*. UTM sees the expected trajectory of UAS 2 (grey plane/trajectory) based on spoofed *position notifications*. The *navigation/avoidance grid range* (black dashed line boundary) is shown. The notable moment of the simulation are:

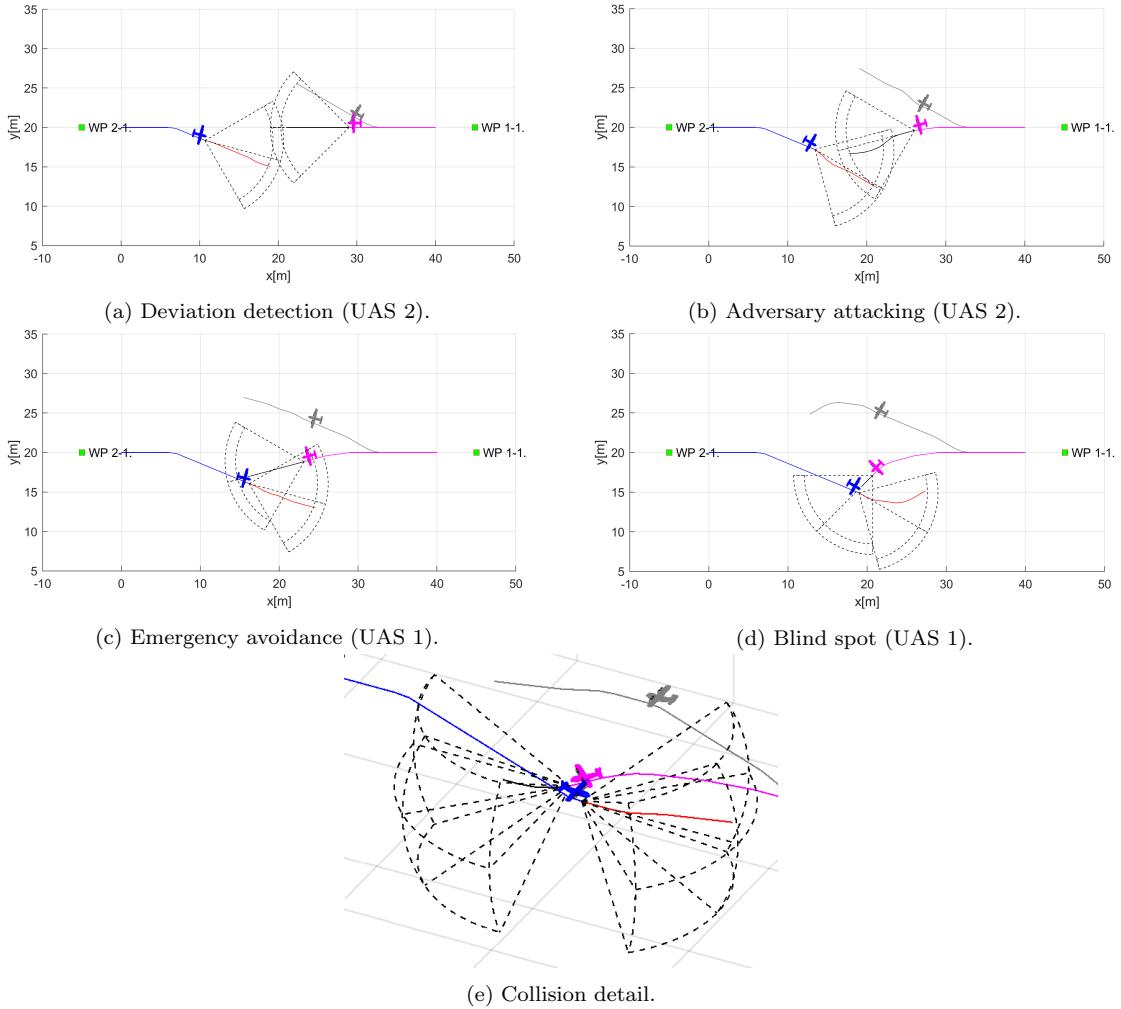


Figure 7.46: Adversarial behaviour of *UAS 2* (magenta) to compliant *UAS 1* (blue)

1. *Deviation detection ($UAS2 \leftrightarrow UTM$)* (fig. 7.46a) - the *collision case* (tab. 7.40) is active and *enforced* by UTM. The *adversary* UAS 2 (magenta) starts deviating from expected trajectory (grey). UAS 1 (blue) does not register any foreign object in *avoidance grid range* (black dashed line).
2. *Adversary attacking ($UAS2 \rightarrow UAS1$)* (fig. 7.46b) - the adversary UAS 2 (magenta) starts actively pursuing UAS 1 (blue) by changing the original heading. This can be considered as the beginning of *active pursuit*. UAS 1 (blue) does not detect any foreign

object in *avoidance grid* (black dashed line boundary). *UTM* is receiving expected UAS position (grey plane/line).

3. *Emergency avoidance* (*UAS1* → *UAS2*) (fig. 7.46c) following happens:
 - a. *Adversary UAS 2* (magenta) is spotted by *UAS 1* (blue), it entered into UAS 1 avoidance grid (black dashed line boundary).
 - b. *UAS 1* (blue) enters into *Emergency Avoidance Mode* because there is a *foreign object* in *avoidance grid*.
 - c. *UTM* notices a warning to *UAS 1* (blue) because it entered into *Emergency Avoidance Mode*. *UTM* is not aware of any breach, because of expected UAS 2 position (grey plane/line)
 - d. *Adversary UAS 2* (magenta) has UAS 1 (blue) locked in *navigation grid* as the goal (which guarantees optimal pursuit).
4. *Blind spot* (\circlearrowleft *UAS1*) (fig. 7.46d) following happens:
 - a. *UAS 1* (blue) returns to *Navigation Mode* because there is no *foreign object* in *avoidance grid* (black dashed line boundary).
 - b. *UTM* receives the mode change, and it starts enforcing *resolutions* for *collision case*, *Adversary UAS 2* is considered clear due to *expected position* (grey plane/line) compliance with the resolution.
 - c. *Adversary UAS 2* (magenta) is on UAS 1 blind spot. The target UAS 1 (blue) is locked in the UAS 2 navigation grid (black dashed line boundary).
5. *Collision detail* (*UAS1* ↔ *UAS2*) (fig. 7.46e) - Target *UAS 1* (blue) is hit by *Adversary UAS 2* (magenta) on left wing tip. Both UAS are going down. *UTM* will detect sudden loss of both UAS systems.

Performance Parameters Evaluation: Performance parameters (y-axis) are tracked over *UTM time* (x-axis). The evolution of *performance* (fig. 7.47) is tracking following parameters:

1. *Expected crash distance* (gray line) - defined as (eq . 7.1) between UAS 1 (blue) and expected UAS 2 position (grey plane/line) over mission time $t \in [0, 22]$.
2. *Crash distance* (blue line) - defined as (eq . 7.1) between UAS 1 (blue) and real UAS 2 position (magenta plane/line) over mission time $t \in [0, 22]$.
3. *Safety margin* (yellow line) - constant value according to *collision case* (tab. 7.40) as the value of 10 m. The safety margin is considered as a *soft constraint*.
4. *Body margin* (red line) - constant value according to (tab. 7.28) as value of 1.2 m. The body margin is considered as a *hard constraint*. The breaking of *body margin* means an effective *collision* UAS 1 and UAS 2.

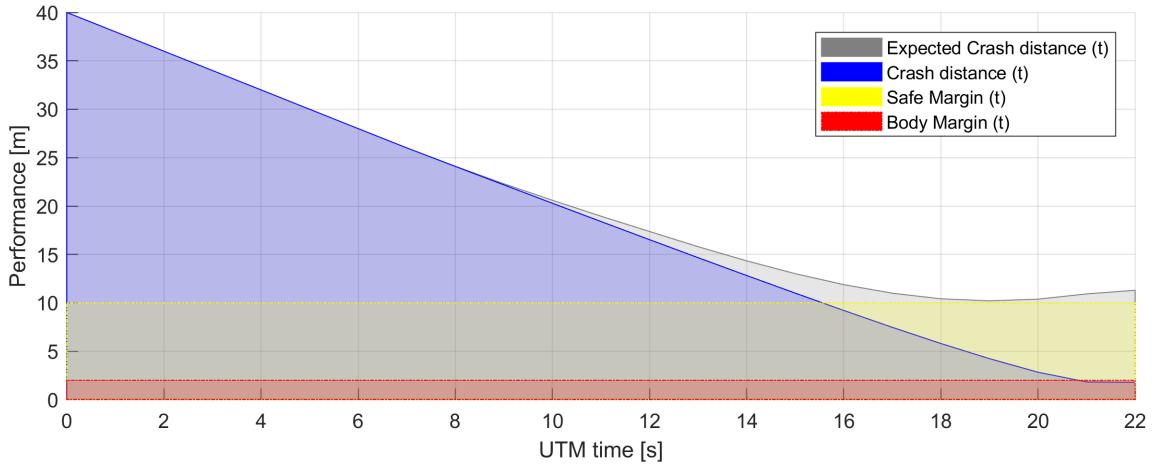


Figure 7.47: Expected/Real Distance to body/safety margin evolution for *adversarial behavior* of UAS 2.

Safety criteria for both *body* and *safety margins* in case of *expected behavior* are satisfied (eq. 7.10). This means that *UAS 1* fulfilled the *UTM directive* even though it entered *Emergency Avoidance Mode* (fig. 7.46c).

$$\begin{aligned} \text{expectedDistanceToSafetyMargin}(t) &\geq 0, & \forall t \in [0, 22] \\ \text{expectedDistanceToBodyMargin}(t) &\geq 0 & \forall t \in [0, 22] \end{aligned} \quad (7.10)$$

Safety Margin is broken at UTM time 15 s, *body margin* is broken at UTM time 21 s, the collision happens at UTM time 22 s. This is summarized in *Distance Condition Breach* (eq. 7.11).

$$\begin{aligned} \text{distanceToSafetyMargin}(t) &< 0, & \forall t \in [21, 22] \\ \text{distanceToBodyMargin}(t) &< 0 & \forall t \in [15, 22] \end{aligned} \quad (7.11)$$

Note. An *adversary behavior* needs to be addressed on:

1. *UAS Traffic Management Level* - our UTM implementation failed to detect *deviation* (fig. 7.46a) and *start of attack* (fig. 7.46b). UAS 2 (magenta) had clean intention from the beginning and did not change pursuit even when *safety margin* was breached.
2. *Emergency Avoidance Level* - our *navigation loop implementation* does not consider the *ill-intentions*. The UAS 1 (blue) properly switched to *Emergency avoidance mode* (fig. 7.46c) after detection of UAS2 (magenta). UAS 2 (magenta) then used the blind spot to exploit UAS 1 vulnerability.

7.5.3 Computation Footprint

The *computation footprint* is summarized in computation load (tab. 7.53). The *computation load* (eq. 7.5) was calculated for each *time-frame* in scenarios. There is the summary of *minimal*, *maximal*, *average* and *median* values.

The *computational load* never exceed more than 55.95% in case of *emergency Head On* (eq. 7.6), which means that *every path* was calculated on time.

Scenario	Computation load			
	min.	max.	avg.	med.
Building avoidance (fig. 7.4)	2.20%	27.40%	12.11%	13.20%
Slalom (fig. 7.8)	12.20%	30.50%	21.42%	21.50%
Maze (fig. 7.12)	24.90%	46.10%	31.51%	30.80%
Storm (fig. 7.16)	2.60%	26.90%	11.57%	13.90%
Emergency Converging (fig. 7.20)	2.75%	16.50%	5.84%	4.95%
Emergency Head On (fig. 7.24)	3.90%	55.95%	13.19%	6.90%
Emergency Multiple (fig. 7.28)	5.90%	52.35%	12.77%	8.56%
Rule-based Converging (fig. 7.32)	3.60%	13.50%	7.32%	5.97%
Rule-based Head on (fig. 7.36)	4.65%	41.60%	13.64%	9.30%
Rule-based Multiple (fig. 7.40)	4.37%	23.30%	11.96%	10.93%
Rule-based Overtake (fig. 7.45)	3.85%	13.40%	7.62%	6.70%

Table 7.53: *Computation load statistics* for all test cases.

Following observations can be made:

1. *Building avoidance*, *Slalom*, and *Maze* scenarios - the computation load is increasing with the *number of static obstacles*. The *average load* for *Emergency avoidance mode* in *clustered environment* is 31.51% (*Maze*).
2. *Storm scenario* - the overall *computation load* is very low due to the *moving constraint implementation* (def. ??).
3. *Emergency Converging/Head On/Multiple* scenarios - the *overall computation load* is quite high due to the ineffective *body volume intersection* (app. ??) implementation.
4. *Rule-based Converging/Head On/Multiple* scenarios - the *median computational load* is low, because of the linear *rule implementation* (sec. ??)
5. *Rule-based Overtake* - the *average computation load* is very low because only *divergence/convergence* (rule. ??) waypoints are calculated and UAS stays in *navigation mode*.

7.6 Reduced Reach Sets Performance

Constrained Expansion Method (alg. ??) is creating *Reach Sets* from the *Root Node* as a tree expansion using *Expansion Constraint function* (depending on type).

The *Reach set creation procedure* is creating the following artifacts:

1. *Nodes* - tree *Node* containing necessary data for discrete Trajectory portion, notably *System State Evolution*, *buffer*, and, *Reachability Rating*.
2. *Trajectories* - leaf *Node* containing *unique buffer* which is not *prefixed* in others *Node buffer*.

The *Reach Set Computation Time* depends strongly on *Movement Automaton* prediction complexity and Node count. The *Constrained Expansion Method* (alg. ??) is separating all nodes entering into $cell_{i,j,k}$ into two distinctive groups: *Candidates for expansion* and *Leftover Nodes*.

The *Leftover Nodes* are thrown away every expansion. The *Leftover Nodes* are not expanded in the next *Wave-front* iteration, but they leave a notable *computation* and *memory* footprint.

Note. *Average Trajectory Smoothness Rate* (def. ??) is important only in *Navigation Mode*; this aspect has been covered over (sec. ??, ??, ??).

Approach: For the same conditions (*Testing Avoidance Grid*, *UAS initial state*, *Movement Automaton*) compare the performance of *Reach Set Approximations* created by various methods for the following parameters:

1. *Coverage Ratio* - defined in (def. ??) shows how versatile *Reach Set Approximation* is (up to 100% of complete reach set coverage).
2. *Node count* - count of Nodes in *Reach Set Approximation* counted like:
 - a. full - all active nodes existing over computation time,
 - b. pruned - active nodes for real-time use.
3. *Count of Trajectories* - count of Trajectories (leaf Nodes) counted like:
 - a. full - all active trajectories existing over computation time,
 - b. pruned - active trajectories are leading to coating cells of *Avoidance Grid*.

Testing Avoidance Grid with *Distance 10 m*, *Layer count 10*, *Horizontal range* $[-45^\circ, +45^\circ]$, *Horizontal Cell Count 7*, *Vertical range* $[-30^\circ, +30^\circ]$, and *Vertical Cell Count 5*.

Note. The sizing of the *Avoidance Grid* was chosen a small scale because the property of *Coverage Ratio* can be calculated exactly up to some scale, after that it can be only assumed. Various sizes of *Avoidance Grid* was tested in [1].

The UAS is at *Back-side* of *figure* (the initial state is at all *Trajectory Origins*). The *black dashed line* marks *Avoidance Grid* space boundary. Each trajectory has own color and ends at *Front-side* of *Avoidance Grid Boundary*.

Coverage-Maximizing Reach Set (sec. ??) is used in *Emergency Avoidance Mode* for *Non-Controlled Airspace*. The *full* set of trajectories is given in (fig. 7.48a). The *Pruned* set of trajectories is given in (fig. 7.48b).

Tuning parameters were selected like follow: *Spread Ratio* is 15 (unique footprint trajectories in the cell), and *trajectory footprint length* is 3 (last three unique passing cells).

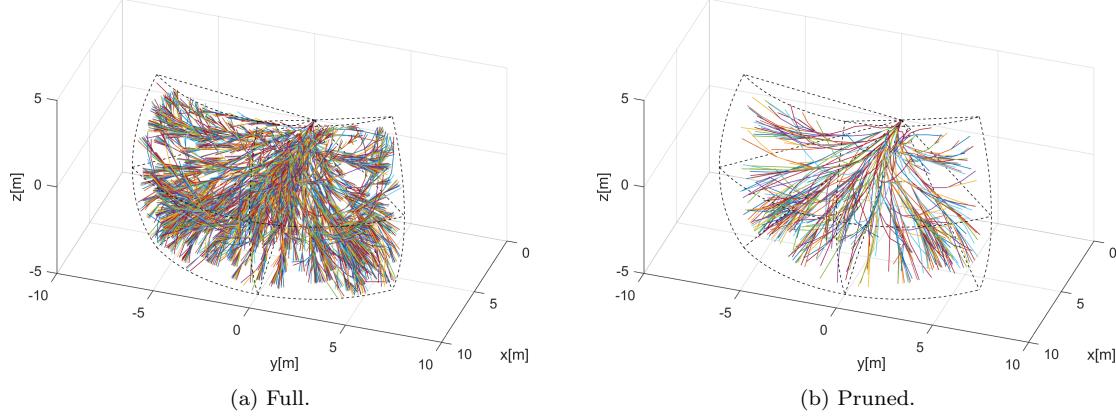


Figure 7.48: Coverage-maximizing reach set computation example.

Turn-Minimizing Reach Set (sec. ??) is used in *Navigation Mode* for *Non Controlled Airspace*. The *full* set of trajectories is given in (fig. 7.49a). The *Pruned* set of trajectories is given in (fig. 7.49b).

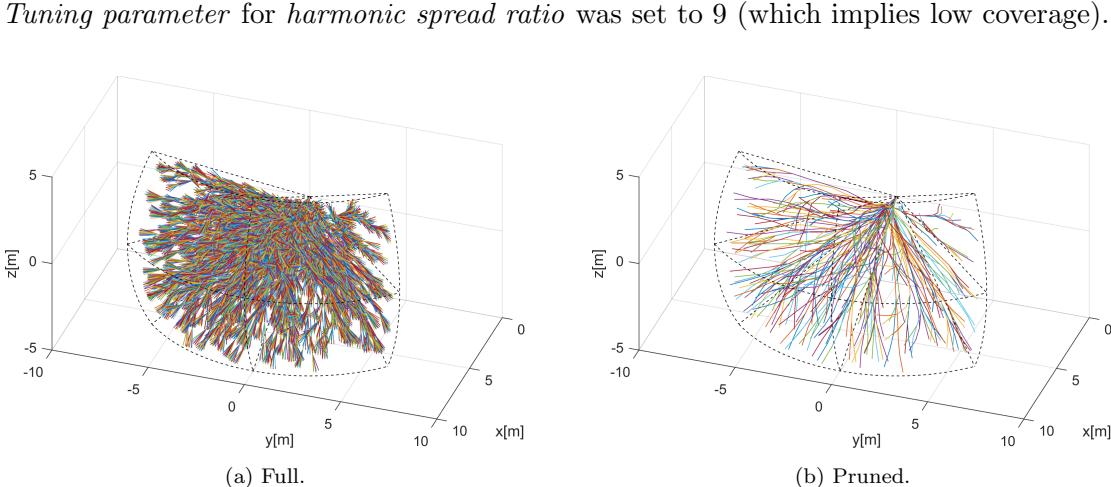


Figure 7.49: Turn-minimizing reach set computation example.

Combined Reach Set (sec. ??) is combination of *Coverage-Maximizing Reach Set* (fig. 7.48) and *Turn-Minimizing Reach Set* (fig. 7.49). The tuning parameters are the same for the respective methods. It is used for both *Emergency Avoidance* and *Navigation*.

ACAS-like Reach Set (sec. ??) is used in *Navigation Mode* for *Controlled Airspace*. The separations used are *Horizontal*, *Vertical*, *Slash*, and, *Backslash*, to give the worst possible nodes and trajectories count. The *full* set of trajectories is given in (fig. 7.50a). The *Pruned* set of trajectories is given in (fig. 7.50b).

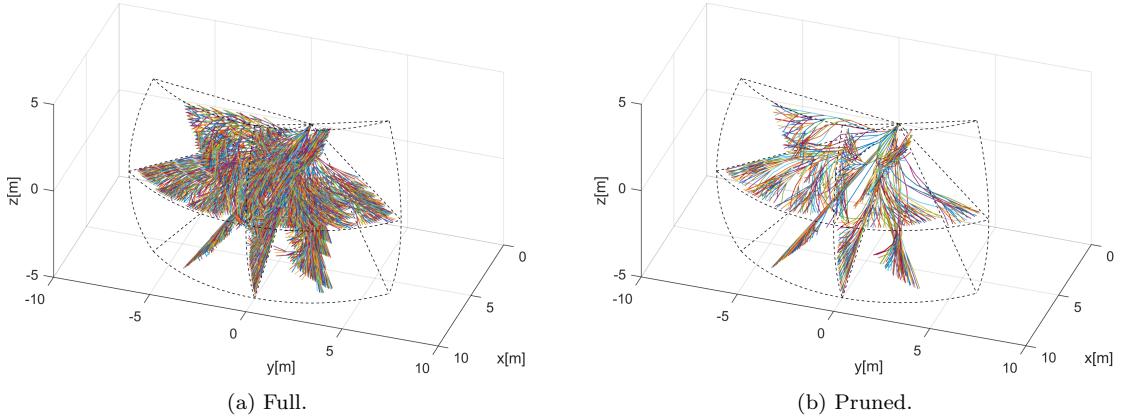


Figure 7.50: ACAS-like reach set computation example.

Computation Methods Performance Comparison (tab. 7.54) gives overview of memory consumption and *Coverage Ratio*.

Node count: *Full Node Count* shows how much memory it takes to compute *Reach set*. *Pruned Node Count* shows how much *memory* is needed for storage.

Note. The total size of *full/pruned Reach Set* depends on Node implementation. The Object-oriented prototype implementation in Matlab for example avoidance grid took up to 1 megabyte of system memory. The effective implementation would take up to 100 kilobytes.

Constrained expansion (alg. ??) have different selection rate, depending on the method. The survival rate directly reflects strictness of selection criteria. The rate of *node pruned* is summarized in (eq.7.12)

$$\begin{aligned}
 & \text{Nodes pruned} \\
 CM - RSA & : 78.93\% \\
 TM - RSA & : 18.50\% \\
 ACAS - like & : 79.05\%
 \end{aligned} \tag{7.12}$$

The *interpretation of results* for each reach set estimation method is like follow:

1. *Coverage-Maximizing* - the main exploration drive is *Coverage Rate*, the *Trajectory* segments are not usually smooth. For our *Movement Automaton*, there is only one *Smooth Movement*: Straight. Other eight are considered *Chaotic Movements*. Impact of this fact is significant because 4/5 of nodes were pruned.
2. *Turn-Minimizing* - the main exploration drive is *Smoothness* of contained *Trajectories*. The *Trajectory segments* which are getting further away from *cell center* are not feasible. If *Smooth Movements* set size is considered, the Smooth/Chaotic movement ratio is 1/8 for our *Movement Automaton* implementation. The low node count was expected in this approach. Another Contributing factor is *Trajectory Footprint Length* for uniqueness selection, which is not a tuning parameter in this method, and it is set to the most strict selection.
3. *ACAS-like* - the main drive is to create set consisting from *multiple 2D separation planes*. The expansion method applies full movement set on the *candidate node*. The *Separation*

plane movement subset is determining, which node will be selected for further expansion. The size of separation plane subset to the size of movement set rate is 1 : 3. There are four separation planes: horizontal, vertical, slash and backslash each containing full 2D plane reach set approximation which caused high node prune rate. Nodes used rate should get lower with increasing grid size.

Trajectories count: *Full trajectories count* shows how many *leaf nodes* were existing during the calculation process without pruning. The difference between *full node count* and *full trajectories count* is count of inner tree nodes.

Pruned trajectories count shows how many *leaf nodes* are used in run-time of *avoidance algorithm*. The difference between *pruned node count* and *pruned trajectories count* shows the count of inner nodes in active reach set.

The most of *waste leaf nodes* are removed during *layer pruning*: function *reachSet.purgeSameFootprint()* (alg. ??). The *Waste trajectories* or *unused leaf nodes count* have significant impact. Because *leaf nodes* are a side product of *Node Expansion procedure* the amount of *pruned trajectories* is around 90 % regardless of the used method. The results are summarized in (eq. 7.13)

$$\begin{array}{ll}
 \text{Trajectories pruned} \\
 \hline
 CM - RSA & : 91.24\% \\
 TM - RSA & : 88.21\% \\
 ACAS - like & : 89.43\%
 \end{array} \tag{7.13}$$

Calculation method	Node count		Trajectories		Coverage ratio	Parameters
	full	pruned	full	pruned		
CM-RSA	6727	1417	4557	399	90%	spread:15
TM-RSA	1724	1405	1528	180	30%	spread:9
combined	-	2405	-	435	95%	CH spread:15 H spread:9 tree comb.
ACAS-like	11294	2366	7437	786	74.95%	Separations: H/V/S/BS Coverage pruning: disabled

Table 7.54: Reduced reach set computation methods performance

Coverage ratio: (def. ??) is showing how much maneuvering versatility of *Reach Set*. *Full Reach Set Approximation* have coverage ratio of 100 %. It is possible to construct *Reference Reach Set* without constrained expansion method which contains all possible *trajectory footprints*. Following observations for *coverage ratio* can be made:

1. *Coverage-maximizing* reach set estimation method by design select *Nodes* which have the high probability of *trajectory footprint* diversification. The high coverage ratio was achieved at values around 90 %.

2. *Turn-Minimizing* reach set estimation method by design selects most smooth trajectories which cause low *trajectory footprint* diversity. The fairly high coverage ratio of 30 % has been achieved.
3. *Combined* reach set estimation method takes two reach set and combines their trajectory trees into a single trajectory tree. It is given that *Coverage ratio* will achieve at least maximal coverage ratio of original reach sets. Harmonic reach set supplemented narrow smooth trajectories which were throw away previously; this increased overall *coverage ratio* to 95 %.
4. *ACAS-like* reach set estimation method contained four separation planes, which caused that it was similar to *Coverage-Maximizing Reach Set Approximation* for given *Avoidance Grid*, concerning of performance. The coverage ratio For 2D plane was 100 %.

Bibliography

- [1] Alojz Gomola, Pavel Klang, and Jan Ludvik. Probabilistic approach in data fusion for obstacle avoidance framework based on reach sets. In *Internal publication collection*, pages 1–93. Honeywell, 2017.