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*.

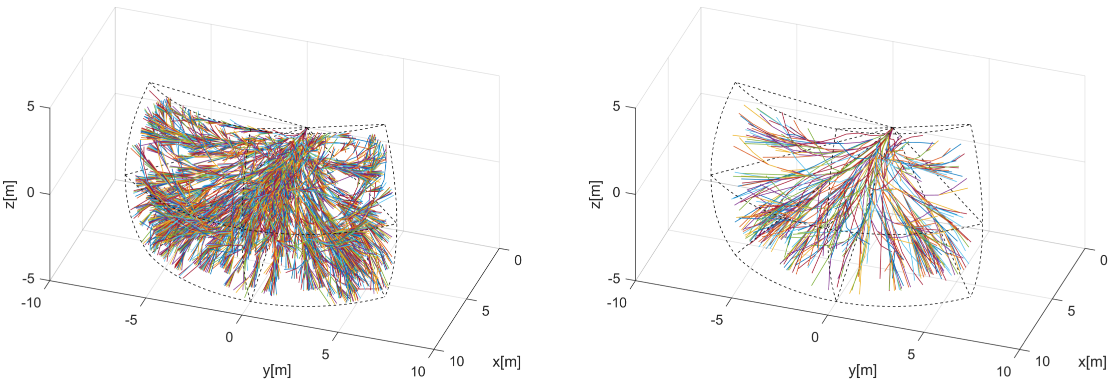
Testing Avoidance Grid with *Distance 10 m*, *Layer count 10*, *Horizontal range* $[-45^\circ, +45^\circ]$, *Horizontal Cell Count 7*, *Vertical range* $[-30^\circ, +30^\circ]$, and *Vertical Cell Count 5*.

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. 7.1a). The *Pruned* set of trajectories is given in (fig. 7.1b).

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).

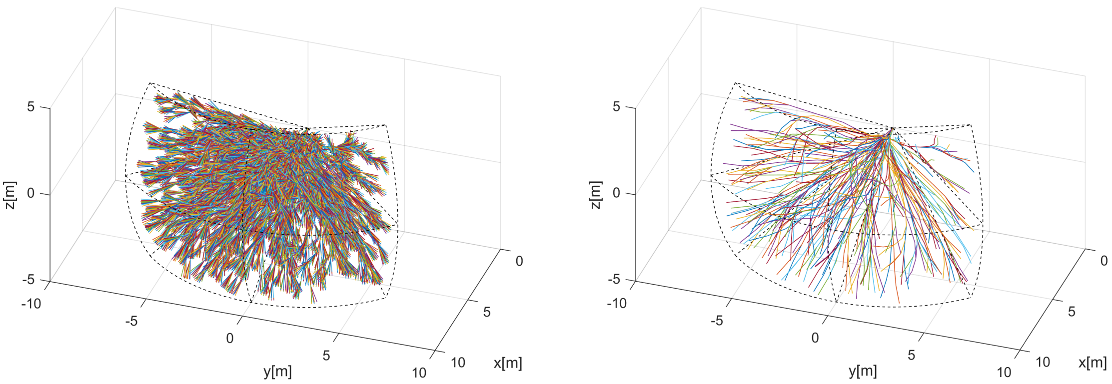


(a) Full. (b) Pruned.

Figure 7.1: 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. 7.2a). The *Pruned* set of trajectories is given in (fig. 7.2b).

Tuning parameter for *harmonic spread ratio* was set to 9 (which implies low coverage).

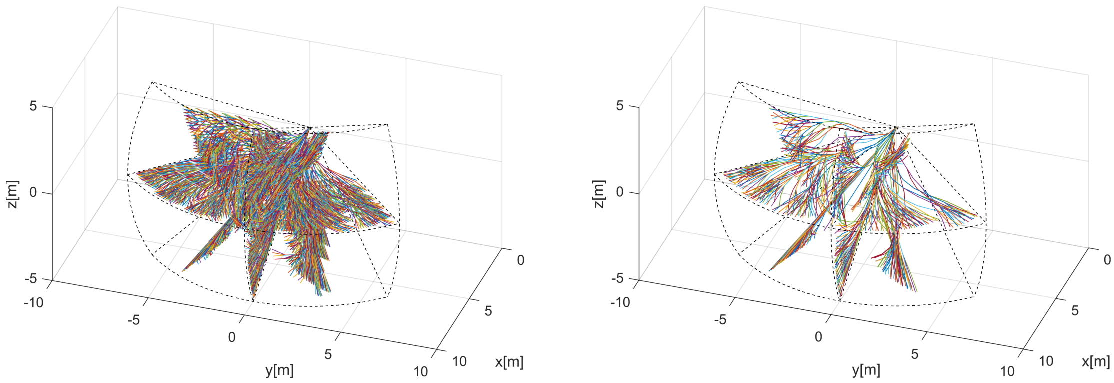


(a) Full. (b) Pruned.

Figure 7.2: Turn-minimizing reach set computation example.

Combined Reach Set (sec. **??**) is combination of *Coverage-Maximizing Reach Set* (fig. 7.1) and *Turn-Minimizing Reach Set* (fig. 7.2). 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. 7.3a). The *Pruned* set of trajectories is given in (fig. 7.3b).



(a) Full. (b) Pruned.

Figure 7.3: ACAS-like reach set computation example.

Computation Methods Performance Comparison (tab. 7.1) 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.7.1)

$$\begin{aligned} &\texttt{Nodes pruned} \\ &\quad CM-RSA &: 78.93\% \\ &\quad TM-RSA &: 18.50\% \\ &\quad ACAS-like &: 79.05\% \end{aligned} \tag{7.1}$$

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. 7.2)

$$\begin{array}{ll} \texttt{Trajectories pruned} & \\ \quad CM-RSA & : 91.24\% \\ \quad TM-RSA & : 88.21\% \\ \quad ACAS-like & : 89.43\% \end{array} \tag{7.2}$$

Calculation method	Node count full	Node count pruned	Trajectories full	Trajectories pruned	Coverage ratio	Parameters
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 7.1: *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.