Appendix D

Approach Guidelines

This appendix contains guidelines topics useful for framework setup.

D.1 Guideline - Grid Size Calculation

Note. This is done for specific type of the system and it is recommended to start with full boundary defined by sensor and then reduce the Avoidance grid according performance capabilities (reach set/computational)

The grid size calculation is done by hand. The following approach has been used in our work.

For *Sensor Field* there is *effective sensor boundary* given as set:

$$Boundary(Sensor \in SensorField) = \{points \in polarCoordinates\} \tag{D.1}$$

The *Boundary* for sensor fields is then given as *union of all singe sensor boundaries*:

$$Boundary(SensorField) = \bigcap_{\forall Sensors} Boundary(Sensor \in SensorField) \tag{D.2}$$

Depending on boundary properties it can be projected into maximal avoidance grid boundary values:

$$Boundary(SensorField) \rightarrow AvoidanceGrid : \begin{matrix} \max(distanceRange) \\ \max(horizontalRange) \\ \max(verticalRange) \end{matrix} \tag{D.3}$$

Our approach taken worst LiDAR performance into account [1] and following parameters for avoidance grid were calculated:

1. distance range $[0m, 10m]$,
2. horizontal range $]-180°, 180°]$,
3. vertical range $[-30°, 30°]$.

The *count of layers* is derived from *average distance traveled by one movement application*:

$$layerCount = \frac{|distanceRange|}{\text{avg.} \quad length(movement \in MovementSet)} \tag{D.4}$$

The *layer length* is based on *our movement set* (tab. **??**, **??**) the average movement length is 1 m; therefore the *layer count* is 10.
The *efficient boundary* is given by *Reach Set*. Estimate reach set coverage space using *ellipsoidal toolbox* [2] up to given *sensor field* maximal distance:

$$Boundary(ReachSet) = Ellipsoid(UASSystem, distance) \tag{D.5}$$

The values for *Reach Set Boundary* with distance 10 m was following:

1. distance range $[0m, 10m]$,
2. horizontal range $[-45°, 45°]$,
3. vertical range $[-45°, 45°]$,

The *Avoidance Grid* boundary is given as *intersection* of all boundaries:

$$Boundary(AvoidanceGrid) = Boundary(ReachSet) \cap Boundary(SensorField) \tag{D.6}$$

The values for *Avoidance Grid Boundary* for our UAS system (sec. **??**) following:

1. distance range $[0m, 10m]$,
2. horizontal range $[-45°, 45°]$,
3. vertical range $[-45°, 45°]$,
4. layer count 10, layer distance 1m.

The *horizontal cell count* and *vertical cell count* was estimated by the *rule of thumb* to have value 7 and 5.

D.2 Guideline - Safety Margin Calculation

Safety Margin Determination: To determine *safety Margin* the *Rule of Thumb* is used:

$$maximalBodyRadius \leq safetyMargin \leq 2 \times turningRadius \tag{D.7}$$

The *lower boundary* is given by *UAS* construction because the *UAS* body is considered as a *unit ball* with the radius given as *maximal body radius*.

The *upper boundary* is optional, The *double of* turning radius is used by the *conservative approach* [3].

Safety Margin Bloating: The *discretization* of *Reach Set*, *Operation Space* and *Decisions* imposes standard *mixed integer* problem considering *safety*. This section covers a *non-exhaustive* list of possible *Safety Margin Bloats* in our approach.

Own Position Uncertainty Bloat: The *sensor fusion* is precise, but not *exact* in own UAS position determination. The usual maximal disparity needs to be accounted into *Safety Margin*.

Intruder Position Uncertainty Bloat: The *sensor fusion* of Intruder is precise, but not *exact* in own UAS position determination. The usual maximal disparity needs to be accounted into *Safety Margin*.

Weather bloat: The *Weather* impact type may result in increased *safety margin*. Example: UAS is not humidity resistant, the clouds will be avoided from a greater distance.

Airspace bloat: The *Airspace* depending on cluster or *country* may require greater separation distances, depending on circumstances. The example can be UAS directive to keep minimal separation from obstacles. The *Safety Margin* is usually overridden by UTM directive value.

UTM Synchronization Bloat: *Both UAS* decision times were *synchronized*. The *intruder* can be offset for the *full decision frame*. This is not an assumption, but it shows critical performance. Usually, safety margin is bloated for (worst case offset):

$$safetyMarginBloat = \begin{pmatrix} intruderVelocity \times \dots \\ intruderDecisionFrame \end{pmatrix} [m, ms^{-1}, s] \tag{D.8}$$

Bibliography

[1] Roberto Sabatini, Alessandro Gardi, and Mark A Richardson. Lidar obstacle warning and avoidance system for unmanned aircraft. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 8(4):702–713, 2014.

[2] Alex A Kurzhanskiy and Pravin Varaiya. Ellipsoidal toolbox (et). In *Decision and Control, 2006 45th IEEE Conference on*, pages 1498–1503. IEEE, 2006.

[3] Johann Borenstein and Yoram Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.