Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Supplementary Material
Rev: 1.5
September 22, 2023

AMSwarmX: Safe Swarm Coordination in CompleX Environments via Implicit Non-Convex Decomposition of the Obstacle-Free Space

Vivek K. Adajania
<vivek.adajania@robotics.utias.utoronto.ca>

1 Computing Directional Clearance

We can compute $d^*_{i,r}$ by calling algorithm 1 for each prediction step k. We refer to Fig. 1 (left) for a 2D illustration of the algorithm. First, we cast rays from $\mathbf{p}_{i,r}$ to the circle centred at $\mathbf{p}_i[k]$ in three directions. One ray points towards the center of the circle, and the rest are tangential. The next step is to obtain the first occupied position $\mathbf{p}_{hit}$ among all the directions. In Fig. 1 (left), the red dot is the first occupied position found. We use the `castRay` functionality from the Octomap package that casts a ray and returns the first occupied position or the ray's end position. The third step is to project $\mathbf{p}_{hit}$ perpendicular to the line connecting $\mathbf{p}_{i,r}$ and $\mathbf{p}_i[k]$ obtaining $\mathbf{p}_{proj}$. The final step is to compute the obstacle-free distance from $\mathbf{p}_{i,r}$ to $\mathbf{p}_{proj}$ accounting for the quadrotor's radius r. In the 3D scenario, all the steps remain the same, except we cast a total of five rays as shown in Fig. 1 (right). One ray points towards the center of the sphere, and the next two are tangential to a circle that lies in the plane of the center ray. The last two are tangential to a circle that lie perpendicular to the plane of the center ray.

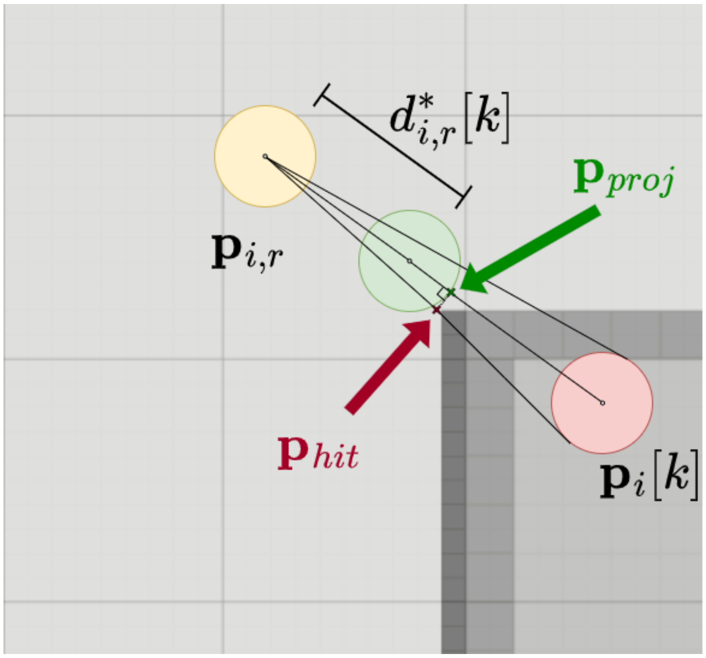
Algorithm 1 `freeDistFromAttractor`

Input Quadrotor position $\mathbf{p}_i[k]$, Attractor position $\mathbf{p}_{i,r}$,
Quadrotor radius r
Output obstacle-free distance d_{free}

1: $\mathbf{p}_{hit} \leftarrow \texttt{cast5RaysAndGetFirstHit}(\mathbf{p}_i[k], \mathbf{p}_{i,r})$
2: $\mathbf{p}_{proj} \leftarrow (\mathbf{p}_{i,r} - \mathbf{p}_{hit}).(\mathbf{p}_i[k] - \mathbf{p}_{i,r})/\|\mathbf{p}_i[k] - \mathbf{p}_{i,r}\|^2$
3: $d_{free} \leftarrow \|\mathbf{p}_{i,r} - \mathbf{p}_{proj}\| - r$
4: **return** d_{free}

2 Checking visibility

We check the visibility of one position from the other using the `castRay` functionality from the Octomap package. Referring to Fig. 1 (right), we cast two pairs of rays with appropriate lengths from position $\mathbf{q}_1$ that are tangential to two circles centred at $\mathbf{q}_2$. One circle lies in the plane of the line connecting $\mathbf{q}_1$ and $\mathbf{q}_2$. The other circle lies perpendicular to that plane. If the ends of the rays are obstacle-free, then we consider $\mathbf{q}_2$ is visible from $\mathbf{q}_1$.



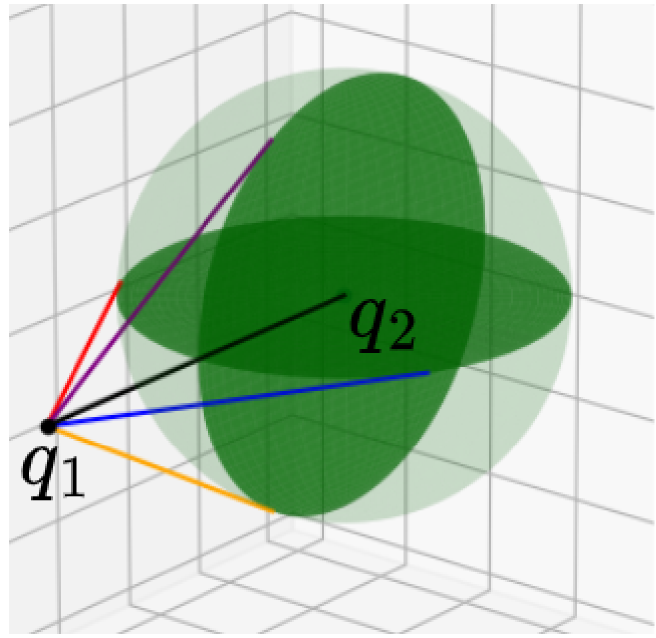


Figure 1: The figure on the left illustrates how to compute a distance from an attractor position that would make a quadrotor position obstacle-free. The right figure showcases a sphere centered at $\mathbf{q}_2$, and five rays being casted from $\mathbf{q}_1$. One ray points towards the center of the sphere, and the next two are tangential to a circle that lies in the plane of the center ray. The last two are tangential to a circle that lies perpendicular to the plane of the center ray.

3 Simulation Parameters

For the AMSwarmX and the AMSwarmED approach, we use a discretization level of $h = 0.075s$ and the length of the prediction horizon $K = 40$. Thus, a planning horizon of $3s$. The single Bernstein polynomial used is of degree $n = 10$. In the trajectory optimization problem formulated in the Sec. II-A of submitted paper, we set w_g=50000, w_s=150, κ=2 and penalize the jerk $(q$=$3)$ trajectory in the cost. In the constraints, we set $\overline{v}$=$1.73ms^{-1}$, $\underline{f}$=$0.3g$ and $\overline{f}$=$1.5g$. We set quadrotor radius $r = 0.13m$.

For the LSC-Planner, we use the open-source implementation [1] provided by the author. Here, we mention the important modifiable parameters. We set the planning horizon to be 3s. The duration of the Bernstein polynomial piece is set to 0.2s, which gives us a total of 15 pieces. By default, the degree of the Bernstein polynomial is set to 5, and the jerk trajectory is penalized in the cost function. Similar to AMSwarmX and AMSwarmED, we set similar quadrotor dimensions.

References

[1] J. Park, D. Kim, G. C. Kim, D. Oh, and H. J. Kim, "LSC-Planner." https://github.com/qwerty35/lsc_planner, 2022.