Supplementary: Real-Time Line-of-Sight Maintenance in Multi-Robot Navigation with Unknown Obstacles

1 Formulation of Communication and Collision Avoidance Constraints

Note we adopted the formulations in [1] to derive the communication radius (C1) and collision avoidance constraints (C3) in this work.

1.1 C1: Communication Radius Constraints $\alpha_{ij}(\cdot)$

For two robots $i, j \in \mathcal{R}$, their relative distance is calculated as $d_{ij} = \|\boldsymbol{q}_i - \boldsymbol{q}_j\|$. The potential function quantifies the communication constraints between the two robots is defined as

$$\alpha_{ij} = \begin{cases} k_\alpha, & 0 \leq d_{ij} \leq d_{\text{min}}^{\text{com}} \\ \frac{k_\alpha}{2}[1 + \cos(\frac{d_{ij} - d_{\text{min}}^{\text{com}}}{d_{\text{max}}^{\text{com}} - d_{\text{min}}^{\text{com}}})\pi], & d_{\text{min}}^{\text{com}} < d_{ij} \leq d_{\text{max}}^{\text{com}} \\ 0, & d_{ij} > d_{\text{max}}^{\text{com}} \end{cases} \tag{1}$$

where $d_{\text{min}}^{\text{com}} \geq 0$ is the distance at which communication reliability starts to decrease; and the communication breaks when $d_{ij} > d_{\text{max}}^{\text{com}}$. The derivative of α_{ij} w.r.t. $\boldsymbol{q}_i$ is calculated as

$$\frac{\partial \alpha_{ij}}{\partial \boldsymbol{q}_i} = \frac{\partial \alpha_{ij}}{\partial d_{ij}} \cdot \frac{\partial d_{ij}}{\partial \boldsymbol{q}_i}. \tag{2}$$

1.2 C3: Collision Avoidance Constraints $\gamma_{ij}(\cdot)$

The potential function for collision avoidance constraints is defined as

$$\gamma_{ij}^* = \begin{cases} 0, & 0 \leq d_{ij} \leq d_{\text{min}}^{\text{coll}} \\ \frac{k_\gamma}{2}[1 - \cos(\frac{d_{ij} - d_{\text{min}}^{\text{coll}}}{d_{\text{max}}^{\text{coll}} - d_{\text{min}}^{\text{coll}}})\pi], & d_{\text{min}}^{\text{coll}} < d_{ij} \leq d_{\text{max}}^{\text{coll}} \\ k_\gamma, & d_{ij} > d_{\text{max}}^{\text{coll}} \end{cases} \tag{3}$$

Here $d_{\text{min}}^{\text{coll}}$ and $d_{\text{max}}^{\text{coll}}$ are the minimum allowed inter-robot distance and the threshold of the inter-robot distance to influence safety, respectively.

The collision avoidance constraints are different from other constraints, as the robot may fail once it collides with others. This edge weight γ_{ij} is defined as [1]:

$$\gamma_{ij} = \left(\prod_{k \in \mathcal{N}_i} \gamma_{ik}^{*} \right) \cdot \left(\prod_{k \in \mathcal{N}_j / \{i\}} \gamma_{jk}^{*} \right) = \gamma_i \cdot \gamma_{j/i}, \tag{4}$$

which is the product of weights of all edges (specifically that reflect the collision avoidance constraints) connected to robot i and j, without repetition. With such, if robot i collides with other robots, the weights of all outgoing edges will be zero, *i.e.*, robot i will be disconnected to $\mathcal{G}$. Moreover, it ensures that $\gamma_{ij} = \gamma_{ji}$.

As introduced in Sec. VII-A, we treat the closest obstacle point around a robot $i \in \mathcal{R}$ as a virtual neighbor, with a virtual index i^{obs}. The Eq. (4) is updated as:

$$\gamma_{ij} = \left(\prod_{k \in \mathcal{N}_i \cup \{i^{\text{obs}}\}} \gamma_{ik}^{*} \right) \cdot \left(\prod_{k \in \mathcal{N}_j \cup \{j^{\text{obs}}\} / \{i\}} \gamma_{jk}^{*} \right). \tag{5}$$

The derivative of γ_{ij} w.r.t. $\boldsymbol{q}_i$ is calculated as

$$\frac{\partial \gamma_{ij}}{\partial \boldsymbol{q}_i} = \gamma_{ij} \cdot \sum_{k \in \mathcal{N}_i \cup \{i^{\text{obs}}\}} \left(\frac{1}{\gamma_{ik}^{*}} \cdot \frac{\partial \alpha_{ik}^{*}}{\partial d_{ik}} \cdot \frac{\partial d_{ik}}{\partial \boldsymbol{q}_i} \right) \tag{6}$$

2 Proof of Propositions

2.1 Proof of Proposition 1

Proof. To compute the derivative in Eq. (8), each robot $i \in \mathcal{R}$ only needs local information from its one-hop neighbors $\mathcal{N}_i$. For each robot $j \in \mathcal{N}_i$, its pose $\langle \boldsymbol{q}_j, R_j \rangle$ and the convex hull $Conv(\mathcal{C}'_j)$ derived from point cloud measurements need to be communicated to robot i. Note robots do not need to share their raw point cloud, but only the flipped convex hull, which is more compact. The derivative of the other two weights α_{ij} (C1) and γ_{ij} (C3) can also be obtained through distributed calculation, as proved in [1]. Therefore, this method can be deployed in a distributed manner. The only concern is that the graph Laplacian matrix is global information for the robot team. However, it is shown in [2] that both λ_2 and v_{2i} can be estimated through distributed estimation. In conclusion, the connectivity force in Eq. (8) can be calculated distributedly. □

2.2 Proof of Proposition 2

Proof. According to the definition of the connectivity velocity $\boldsymbol{u}_i^{\text{c}}$ in Eq. (8), if λ_2 approaches $\lambda_2^{\min}$, the term $\frac{1}{(\lambda_2 - \lambda_2^{\min})^2}$ grows unbounded. When the navigation velocity $\boldsymbol{u}_i^{\text{n}}$ is bounded, the connectivity velocity $\boldsymbol{f}_i^{\text{c}}$ will dominate the movement of robot i, forcing robots to be connected. Note that Prop. 2 is also verified in experiments as shown in Fig. 7. □

Bibliography

[1] Paolo Robuffo Giordano et al. “A passivity-based decentralized strategy for generalized connectivity maintenance”. In: *The International Journal of Robotics Research* 32.3 (2013). Publisher: SAGE Publications Sage UK: London, England, pp. 299–323.

[2] Peng Yang et al. “Decentralized estimation and control of graph connectivity for mobile sensor networks”. In: *Automatica* 46.2 (2010). Publisher: Elsevier, pp. 390–396.