Seminar - Collective navigation of a multi-robot system in an unknown environment

Gregor Germerodt
Hochschule Trier - Trier University of Applied Sciences
E-Mail: grgr3403@hochschule-trier.de
Themensteller und Betreuer: Prof. Dr. Jürgen Graf
Cognitive Systems and Robotics

***Abstract*—This paper is a summary of the work of Olcay (2020). All content is based on this work and its referenced sources have been incorporated where necessary.**

Various methods for the movement of robots in unknown environments are presented, for collision avoidance and target achievement using virtual forces. First, the methods for a single robot are explained and then extended to a multi-robot system with the help of a communication network.

I. Introduction

[1] Multi-robot systems, such as autonomous robots collaborating in a swarm, offer applications in unknown environments like mapping, exploration, or rescue missions [2]. To ensure collision-free paths, this work addresses collective navigation under limited sensor and communication ranges. To enhance system robustness, a nature-inspired swarm behavior is applied, based on three simple rules: cohesion, separation, and alignment [3]. While global motion planning assumes an initial map with static obstacles, this work focuses on local motion planning that can dynamically respond to environmental changes. Instead of using Simultaneous Localization and Mapping (SLAM), where robots build a map while estimating their own position, this approach relies on a decentralized, potential field-based navigation method using only local sensor data and neighbor-to-neighbor communication. This enables the robots to make early, collective decisions for collision avoidance while ensuring scalability and flexibility of the system.

II. Theoretical background

In the multi-robot system under consideration, the robots are modeled as point masses (i.e., without spatial extension, only by position and mass point) in a two-dimensional environment. The system dynamics are based on potential field-based forces, whereby only the position and velocity of each robot i are taken into account, $(p_i, v_i) \in \mathbb{R}^2 \times \mathbb{R}^2$. The motion dynamics of each robot i are described by the following differential equations:

$$\dot{p}_i = v_i \tag{1}$$

$$\dot{v}_i = u_i \tag{2}$$

The position of the robot therefore changes with its velocity (1), which in turn changes due to a control force (2). Each robot can only interact bidirectionally with neighboring robots within a fixed communication radius r_c. The time-dependent and undirected neighborhood is defined by

$$\mathcal{N}_i^\alpha = \{j \in \mathcal{V} \mid \|p_j - p_i\| < r_c\} \tag{3}$$

with $\mathcal{V} = \{1, \ldots, N\}$, $N \in \mathbb{N}$, the set of all robots. To generate swarm behavior, distance values are calculated between two robots, which determine the deviation energy, the degree of interaction, and connectedness ((16) to (18)).

The control force u_i of a robot i is calculated using the equation

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma \tag{4}$$

where u_i^α is for swarm behavior, u_i^β is for obstacle avoidance, and u_i^γ for navigation. The swarm behavior u_i^α consists of distance and speed adjustments to neighbors ((19)) and their effectiveness depending on the distance ((20) to (21)).

Obstacle avoidance u_i^β is defined by the obstacle points detected by a robot ((22)), how strongly it is repelled by them ((23)) and how this repulsive force is calculated as a function of distance ((24)). The control force for navigation u_i^γ attracts the robot to the target ((25)).

The problem statement

Many existing navigation approaches for multi-robot systems are based on artificial potential fields, in which robots are attracted to target points and repelled by obstacles . However, a central problem with these methods is local minima, in which attractive and repulsive forces balance each other out and robots get stuck. Methods for individual robots, such as camera- or elimination-based approaches, reach their limits in difficult environmental conditions or obstacle configurations. Furthermore, the challenge for multi-robot systems is to navigate collision-free and efficiently in unknown and complex environments, despite limited sensor and communication ranges. For this reason, this study deals with communication and motion planning for multiple robots, whereby noise effects, time delays, or packet loss are not taken into account.

III. Single robot navigation

Before a robot can act in a swarm, it must be able to cope on its own. To this end, three methods are presented in this section:

- Tangential navigation, inspired by [4],

- Corner avoidance, and
- Motion planning at obstacle extremities.

Basically, a robot wants to move to a specified destination while watching out for obstacles within a certain sensor radius $r_{\tan}$. If it detects an obstacle (represented by the point $\hat{p}_{i,k}$), it avoids it by moving parallel to the edge of the obstacle, i.e., tangentially. This is done taking into account the angles α_i and β_i depending on the direction of movement and the resulting rotation angles γ_i. The robot's position $\hat{p}_{i,k}$ is determined by the coordinates of the sensor $\hat{p}_{i,k}$ and the distance $r_{\tan}$ from the obstacle $\hat{p}_{i,k}$. The robot's This is done taking into account the movement direction-dependent angles α_i and β_i and the resulting rotation angle γ_i. These angles determine the evasion direction ((26) to (28)) and a virtual target ((29)). Using an adjusted control force ((30) similar to (25)) , the robot is accelerated toward the virtual target p_i^v.

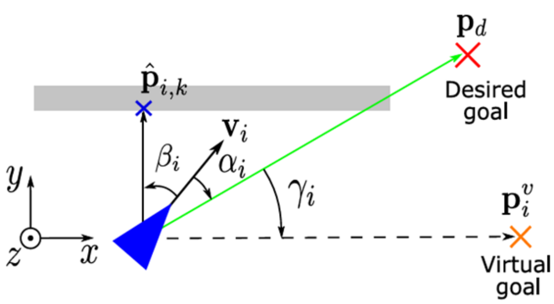
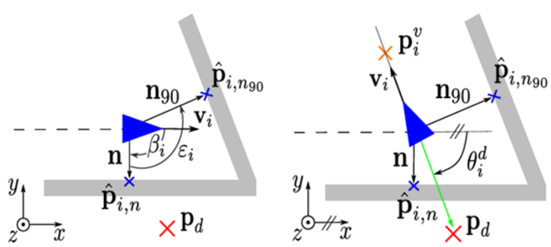


Fig. 1. Strategy for tangential navigation.

If the robot detects two obstacles (represented by two points $\hat{p}_{i,n}$ and $\hat{p}_{i,n90}$) that form a corner, it applies the *Corner Avoidance* maneuver. In this maneuver, the rotation angle γ_i is supplemented by an additional angle ε_i , which is calculated from the tangent directions n and n_{90} to both obstacle points relative to the robot ((31) to (32)). Then, a virtual target p_i^v is determined again, which guides the robot out of the corner ((33)).



(a) Corner detection. (b) Corner avoidance maneuver.

Fig. 2. Strategy for corner avoidance.

When the robot reaches the end of an obstacle, it performs a circular movement around the end. To do this, it remembers the last obstacle point $\hat{p}_{i,e}$ that was registered within its sensor radius. Using equation ((34)) and the angle β_i, which was obtained by tangential navigation , the robot calculates virtual targets at equal intervals (calculated by equation ((35)) and represented by the points p_i^{v1}, p_i^{v2} and p_d) at equal intervals and rotates by a fixed angle δ until the angle between the motion vector and the virtual target is less than or equal to δ ((36)).

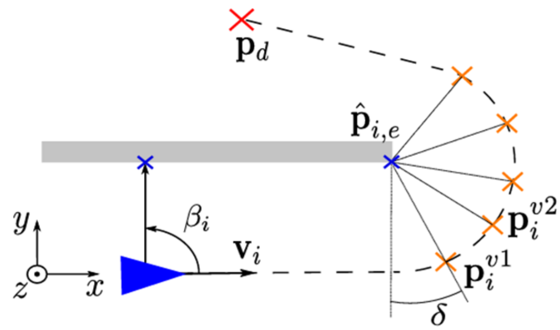


Fig. 3. Circular Motion at obstacle endpoint.

IV. The Proposed Navigation Approach for a Multi-Robot System

The approach presented here extends the navigation algorithm for individual robots with a communication interface for the collective motion planning of multiple robots. The cohesion of the group could be jeopardized if each robot pursues its own virtual goal. Therefore, information about virtual targets and critical points (as shown in Fig. 4) is exchanged between them. This information is prioritized according to relevance, timeliness, and the principle of "detection before communication" (detected obstacles take precedence over transmitted information from the communication network). The communication network consists of information packets, which are divided into three types : orientation, endpoints, and corners.

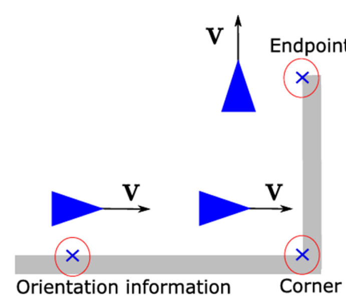


Fig. 4. The critical points of obstacles.

Orientation (Fig. 5): Contains the angle θ_i to the current target, the detected point $\hat{p}_{i,k}$ on the obstacle, the current status of the robot (all statuses explained in Table 1), and the time of obstacle detection.

Endpoints (Fig. 6 (a)): Consists of the last detected point $\hat{p}_{i,e}$, the angle $\omega_{i,e}$ between the normal vector from the robot to the obstacle and the x-axis of the inertial coordinate system[1] and the target angle $\theta_{i,e}$ at the moment of endpoint detection.

[1]unaccelerated and subject to the law of inertia coordinate system

Corners (Fig. 6 (b)): Includes the target angle at entry ($\theta_{i,\text{ent}}$) and exit ($\theta_{i,\text{ex}}$) from the corner, as well as the corner point $\hat{p}_{i,c}$, which is calculated as the intersection of the detected obstacle lines.

Before a robot performs an action based on the orientation information provided by the communication network, its relevance is calculated using the equation rel, which lies in the interval $]-\infty, 10]$. Here, $\text{rel} = 10$ stands for highly relevant information and $\text{rel} \leq 0$ for information to be ignored. The distinction is made as follows:

- **Age of information:** The equation
$$\text{rel}_t = 10 - \frac{10 \cdot (t_k - \hat{t}^c)}{d_t} \tag{5}$$
determines: the older the information, the less important it is, with the current time t_k and the registration time $\hat{t}^c$ of the information from another robot and $d_t \in \mathbb{R}$ a constant that makes rel_t negative after a certain time.
- **Distance from the obstacle:** If the obstacle is in the direction of movement, the relevance of the distance is determined by
$$\text{rel}_\text{dist} = 10 - \frac{10 \cdot \|\hat{p}^c - p_i\|}{d_x} \tag{6}$$
where $d_x \in \mathbb{R}$ is the maximum distance and $\|\hat{p}^c - p_i\|$ is the distance between the robot (p_i) and the obstacle ($\hat{p}^c$). If it is behind, then with
$$\text{rel}_\text{dist} = -\frac{10 \cdot \|\hat{p}^c - p_i\|}{d_x} \tag{7}$$
and if no orientation information is given, then $\text{rel}_\text{dist} = 0$.
- **Relevance of orientation based on the previous time step:** Each robot expects only small changes in tangential navigation for each time step. The equation
$$\text{rel}_\text{exp} = 10 - \frac{10 \cdot |\theta_i(t_k) - \theta^c|}{d_\theta} \tag{8}$$
describes the relevance of the expected orientation, where $|\theta_i(t_k) - \theta^c|$ is the comparison with the current target orientation ($\theta_i(t_k)$) and the new orientation (θ^c) and d_θ is the maximum allowed angle difference, so that $\text{rel}_\text{exp} > 0$.
- **Evaluation of the sender:** Received information is more relevant if the sender is the owner of this information.
$$\text{rel}_o = \begin{cases} 10, & \text{if sent directly} \\ 0, & \text{if only forwarded} \end{cases} \tag{9}$$
- **Evaluation based on status:** Actions that realign the robot are rated as having the highest relevance. All *status* can be found in Table 1 .
$$\text{rel}_\text{type} = \begin{cases} 10, & \text{if } status^c = 4 \vee status^c = 3 \\ 5, & \text{if } status^c = 1 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The final relevance is calculated using the equation
$$\overline{\text{rel}}_n = \frac{c_\text{type} \cdot \text{rel}_\text{type} + c_o \cdot \text{rel}_o + c_\text{exp} \cdot \text{rel}_\text{exp} + c_\text{dist} \cdot \text{rel}_\text{dist} + c_t \cdot \text{rel}_t}{c_\text{type} + c_o + c_\text{exp} + c_\text{dist} + c_t} \tag{11}$$
where the maximum relevance is calculated by
$$\text{rel}_\text{max} = \underset{\text{rel}_n \in \mathcal{R}_i}{\text{argmax}}(\text{rel}_n) \tag{12}$$
with $n \in \mathcal{S}_i$, the set of all information packets received from the network, and $\mathcal{R}_i$ the set of all relevance values extracted from the information packets for robot i.

An information packet is ultimately relevant for a robot if the condition
$$\text{rel}_n \geq 0.95 \cdot \text{rel}_\text{max} \tag{13}$$
is satisfied. It then adjusts its direction of movement for the next time step. If several information packets satisfy this condition, the mean value is determined from them.

For transmitted corners or endpoints, the robot checks its distance d_s with
$$\begin{aligned} d_s \leq \|p_i - \hat{p}_c^c\| \leq R_\text{rel}, \\ d_s \leq \|p_i - \hat{p}_e^c\| \leq R_\text{rel}, \end{aligned} \tag{14}$$
and then considers these points as its own obstacle points if they lie within the maximum distance $R_\text{rel} \in \mathbb{R}$.

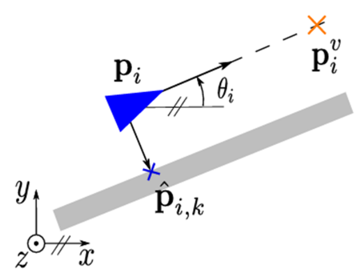
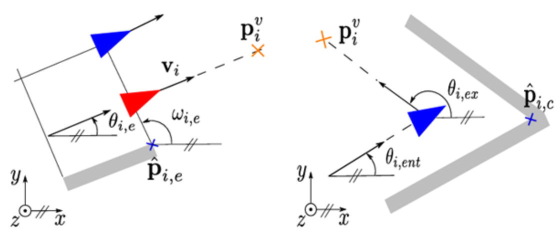


Fig. 5. Information about orientation.



(a) Information about an endpoint.

(b) Information about a corner.

Fig. 6. Endpoint and corner Information.

V. Collective Navigation Using Shared Information

This section explains how the procedures of a single robot can be extended to a multi-robot system. For this purpose, seven **statuses** are defined in which a robot can be found, with each robot starting in **status 0**.

status_i	Definition
0	Motion toward the desired goal position
1	Obstacle detection and tangential navigation
2	Handling the endpoint of an obstacle
3	Corner avoidance maneuver
4	Orientation phase
5	Tangential navigation based on received information
6	Waiting mode

TABLE I
DEFINITION OF ACTION STATUSES.

Collective tangential navigation

Collective tangential navigation comprises **status 1**, **4** and **5**. If a robot detects an obstacle within its perception range $r_{\tan}$, it initiates tangential navigation, as with a single robot. This action is referred to as **status 1**. The information obtained in this process is forwarded to neighboring robots. Since the control variable u_i^α according to equation ((19)) can cause robots to move away from each other by repelling each other perpendicular to the obstacle, which could unintentionally trigger a change from **status 0** to **1**, an ignorance condition (15) was developed so that robots can ignore obstacles.

$$\begin{aligned}(|\upsilon_i| > 90^\circ \wedge |\beta_i| > 91^\circ)\vee \\ \|\hat{p}_{i,k} - p_i\| > 0.3 \cdot (r_{\tan} - d_s) + d_s\end{aligned} \tag{15}$$

If new information from the communication network causes a robot to be misaligned and the difference between its current movement and target orientation is greater than 45°, it switches to status 4 to correct its orientation. If a robot calculates a virtual target using equation ((37)) based on the information it has received, it is in **status 5**. To prevent robots from detecting obstacle endpoints too late if they have only been detected by other neighboring robots, a distance estimation with case differentiation ((40)) is performed. To prevent robots from detecting obstacle endpoints too late when these have only been detected by other neighboring robots, a distance estimation is performed with a case distinction ((30)). This allows the robot to adjust its adjust its target orientation in good time and thus avoid delays in the movement sequence.

Collective corner avoidance

If a robot approaches a corner, it can navigate around it as described in Section III, while also using information from the communication network. This action is referred to as **status 3**.

To ensure that no possible passages (e.g., between two obstacles) are overlooked, the robot first checks the condition ((39)). Only if this condition is met is the corner recognized as such. The robot then defines a new virtual target and switches to **status 4** to realign itself. At the same time, it sends information about the recognized corner to the neighboring robots. However, if the condition ((39)) is not met, the robot initially remains in waiting state (**status 6**). In this state, it monitors the communication network of its neighbors. If information with **status 3** or **4** is classified as relevant , the robot adopts it and leaves the waiting state. If, instead, there is relevant information with **status 1** or **2**, it adapts accordingly and either follows the tangential navigation or the maneuver at an obstacle endpoint.

Collective motion at obstacle extremities

If a robot detects the endpoint of an obstacle, it follows virtual target positions on a circular path (**status 2**), as described in Section III. Each robot sets individual targets on this path and shares information about the endpoint in its network (Fig. 7).

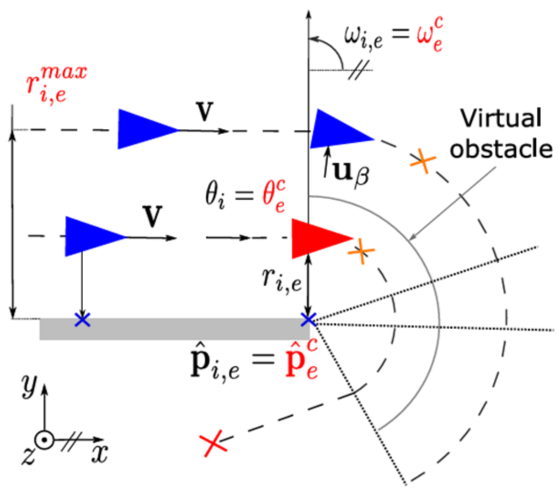


Fig. 7. Collective maneuver at an obstacle endpoint.

After receiving the endpoint data, a robot checks its position by orthogonal projection onto a virtual start line before beginning the circular motion. If the robot has not yet reached this start line ((40)), it sets further tangential targets and adjusts its speed to the minimum speed for circular motion ((41) to (42)). Once it reaches the start line, circular motion begins according to the scheme presented in Section III.

During the movement, a component from equation ((19)) ensures that the robot reaches its target on the circular path. If a robot exceeds its virtual target, a new target position is calculated.

The robots move on individual circular paths with different radii, which means that inner circles result in shorter paths. The speed consensus for circular motion is based on equal angular velocities. For this purpose, a minimum angular velocity ((43)) is specified and the control variable from equation ((19)) is replaced by an adapted control variable from equations ((44)) and ((45)). This allows the robots to circle the end of the obstacle uniformly.

Conditional Braking

When several robots in a group perform a circular movement around an obstacle, differences in the timing of leaving the circular path can cause speed differences that lead to the group breaking apart. To prevent this, robots that have already passed the obstacle are slowed down. This deceleration remains active until the last robot in the group has left the circular path. To determine this last robot, a new coordinate system is defined whose origin is at the end point of the circular path. This deceleration remains active until the last robot in the group has left the circular path.

To determine this last robot, a new coordinate system is defined, whose origin lies at the end point of the obstacle. In this system, the position of each robot is projected onto the x-axis. Using these projected positions and equation ((46)), the robots observe their neighborhood and identify the robot that is furthest behind.

Via gossip-like communication[2] , the entire group is informed about this candidate until all robots have determined the global rear robot. Once this robot has completed its circular movement, it sends a reset signal to release the brakes of the remaining robots and resynchronize the group movement.

Watchdog timer

If the speed of a robot falls below a defined threshold value, it activates a self-monitoring mechanism called the *watchdog timer*. The robot stores its current position and the current time. The timer is deactivated again as soon as the robot moves more than a specified threshold value within the following time steps (the robot assumes that it has left a possible deadlock[3]). However, if the robot hardly moves for a period of 15 seconds and the timer has not been deactivated, it automatically switches to **status 0** and moves toward the desired goal. Although this measure can lead to fragmentation of the group, it ensures that each robot reaches the desired goal.

VI. Simulation results

The simulation results are presented using two scenarios: one with a zigzag obstacle and one with a corridor with obstacles. In both cases, twelve robots are randomly positioned within a starting area, each without initial velocity. In addition, a desired goal (marked with a red cross) is defined.

The robots are represented as black triangles in the figures. Their movement is visualized by a colored line behind the triangles, with the direction of movement corresponding to the orientation of the triangle's tip. The black lines between the robots indicate the active communication network. The figures show snapshots at specific points in time or relevant events during the simulation.

In the first scenario (Fig. 8), the following sequences are shown:

- (a): Detection of an obstacle and start of tangential navigation
- (b): Detection of a corner point and execution of the corner avoidance maneuver
- (c) and (d): Detection of an end point and initiation of circular movement around the obstacle
- (e): Approximation of the robots to the obstacle to optimize navigation using equation ((46))
- (f): Reaching the desired destination point

[2]local, decentralized, and iterative information exchange
[3]hopeless situation

The same procedures are used in the second corridor scenario (Fig. 9).

Guideline for parameter choice

The parameters of the relevance function ((11)) should be selected depending on the respective scenario. A global behavior of the multi-robot system, in which all robots perform identical actions almost simultaneously , is suitable for smaller systems with six to twelve robots if their distribution in relation to the obstacles is low. In this case, the relevance of the time of the information packet (`relt`) and the relevance of the action status (`reltype`) should be weighted higher than the other values so that robots react uniformly and as synchronously as possible to current events in the communication network.

However, with an increasing number of robots or a denser obstacle constellation, such global behavior can be problematic, as individual robots could receive several relevant information packets at the same time. In these situations, robots should pay more attention to their own position and local information. To this end, the relevance of the distance to the obstacle (`reldist`), the relevance of the expected orientation for the next action (`relexp`), and the relevance of the sender of the information (`relo`) are weighted more heavily than `relt` and `reltype`.

The effects of these different weightings are shown in the following figures:

- Fig. 10 shows a *squeezing maneuver* in which the group of robots moves between two circular obstacles.
- Fig. 11 shows how a semicircular obstacle divides the group. The robots then correct their direction of movement by checking and adjusting their current action status.
- Fig. 12 illustrates that prioritized obstacle avoidance causes fragmentation of the group. Nevertheless, all robots ultimately reach the desired destination.

VII. Conclusion

In this work, a method for collective navigation of autonomous robots in unknown environments was developed. The approach combines a tangential avoidance strategy with the exchange of local environmental information via a communication network. Collision-free and coordinated movements of the robots are enabled by artificial potential forces that generate swarm-like behavior. Targeted adjustment of the relevance function parameters and weighting can prevent fragmentation of the group. Individual robots retain the ability to reach their destination even without group connection. For holonomic[4] robots, a sufficiently large safety distance and reduced speed are particularly crucial. Future work will address real-time implementation, optimization of wireless communication, and alternative network structures for the robots.

[4]Position can be changed independently of orientation

APPENDIX

Equations, functions and conditions

$$\|z\|_\sigma = \frac{1}{\epsilon}\left(\sqrt{1+\epsilon\|z\|^2}-1\right) \tag{16}$$

$$\rho_h(z) = \begin{cases} 1, & z \in [0,h) \\ \frac{1}{2}\left(1+\cos\left(\pi\frac{z-h}{1-h}\right)\right), & z \in [h,1) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$a_{ij}(p) = \rho_h\left(\frac{\|p_j - p_i\|_\sigma}{r^\alpha}\right), \quad j \neq i \tag{18}$$

$$u_i^\alpha = c_{\alpha 1}\sum_{j\in N_i^\alpha}\phi^\alpha\left(\|p_j-p_i\|_\sigma\right)n_{ij} + c_{\alpha 2}\sum_{j\in N_i^\alpha}a_{ij}(p)(v_j - v_i) \tag{19}$$

$$\phi^\alpha(z) = \rho_h^\alpha\left(\frac{z}{r^\alpha}\right)\varphi(z-d^\alpha) \tag{20}$$

$$\varphi(z) = \frac{1}{2}\left((a+b)\sigma_1(z+c)+(a-b)\right) \tag{21}$$

$$N_i^\beta = \left\{\hat{p}_{i,k} \mid \|\hat{p}_{i,k}-p_i\| < r_s\right\} \tag{22}$$

$$u_i^\beta = c_{\beta 1}\sum_{k\in N_i^\beta}\phi^\beta\left(\|\hat{p}_{i,k}-p_i\|_\sigma\right)\hat{n}_{i,k} + c_{\beta 2}\sum_{k\in N_i^\beta}b_{i,k}(p)(\hat{v}_{i,k} - v_i) \tag{23}$$

$$\phi^\beta(z) = \rho_h^\beta\left(\frac{z}{d^\beta}\right)\left(\sigma_1(z-d^\beta)-1\right) \tag{24}$$

$$u_i^\gamma = -c_{\gamma 1}\,\sigma_1(p_i - p_d) - c_{\gamma 2}\,v_i \tag{25}$$

$$\gamma_i = \begin{cases} \beta_i - \alpha_i - 90^\circ, & \text{if } \beta_i \geq 0^\circ \\ \beta_i - \alpha_i + 90^\circ, & \text{if } \beta_i < 0^\circ \end{cases} \tag{26}$$

$$c_r = \begin{cases} -1, & \text{for clockwise rotation} \\ 1, & \text{for counter-clockwise rotation} \end{cases} \tag{27}$$

$$\gamma_i = \beta_i - \alpha_i + c_r \cdot 90^\circ \tag{28}$$

$$p_i^v = p_i + \begin{pmatrix} \cos(\gamma_i) & -\sin(\gamma_i) \\ \sin(\gamma_i) & \cos(\gamma_i) \end{pmatrix}(p_d - p_i) \tag{29}$$

$$u_i^\gamma = -c_{\gamma 1}\cdot\left(c_n \cdot \frac{p_i - p_i^v}{\|p_i - p_i^v\|}\right) - c_{\gamma 2}\,v_i \tag{30}$$

$$\gamma_i = \begin{cases} \beta_i - \alpha_i - \tau_i, & \text{if } \epsilon_i \geq 0^\circ \\ \beta_i - \alpha_i + \tau_i, & \text{if } \epsilon_i < 0^\circ \end{cases} \tag{31}$$

$$\tau_i = |\epsilon_i| + 90^\circ \tag{32}$$

$$p_i^v = p_i + \begin{pmatrix} \cos(\gamma_i) & -\sin(\gamma_i) \\ \sin(\gamma_i) & \cos(\gamma_i) \end{pmatrix}\begin{pmatrix} 0.5\cdot d_s\cdot\cos(\theta_i^d) \\ 0.5\cdot d_s\cdot\sin(\theta_i^d) \end{pmatrix} \tag{33}$$

$$\gamma_i = \begin{cases} +\delta, & \text{if } \beta_{i,e} \geq 0^\circ \\ -\delta, & \text{if } \beta_{i,e} < 0^\circ \end{cases} \tag{34}$$

$$p_i^v = \hat{p}_{i,e} + \begin{pmatrix} \cos(\gamma_i) & -\sin(\gamma_i) \\ \sin(\gamma_i) & \cos(\gamma_i) \end{pmatrix} n_{i,e} \tag{35}$$

$$|\alpha_i| \leq \delta \tag{36}$$

$$p_i^v = p_i + \begin{pmatrix} \cos(\theta^c) & -\sin(\theta^c) \\ \sin(\theta^c) & \cos(\theta^c) \end{pmatrix}\cdot e_x \cdot s \tag{37}$$

$$\theta_i(t_{k+1}) = \begin{cases} \theta^c - 20^\circ, & \text{if } \|\tilde{p}_{i,k} - p_i\| < 1.5\cdot r \wedge \beta_i < 0^\circ \\ \theta^c + 20^\circ, & \text{if } \|\tilde{p}_{i,k} - p_i\| < 1.5\cdot r \wedge \beta_i \geq 0^\circ \end{cases} \tag{38}$$

$$\|\tilde{p}_{i,k} - p_i\| < 3\cdot d_s \wedge \|\tilde{p}_{i,k} - \hat{p}_k^c\| < 3\cdot d_s \tag{39}$$

$$(p_i^* - p_i) = -k\cdot(p_i^v - p_i) \tag{40}$$

$$v_i^{\text{ref}} = \frac{p_i^v - p_i}{\|p_i^v - p_i\|}\cdot\frac{d_s}{r_{i,e}^{\max}}\cdot v_{\max} \tag{41}$$

$$u_{i,e}^\gamma = -c_{\gamma e}(v_i - v_i^{\text{ref}}) \tag{42}$$

$$\omega_{\min} = \frac{v_{\max}}{r_{i,e}^{\max}} \tag{43}$$

$$v_i^{\text{ref}} = \frac{p_i^v - p_i}{\|p_i^v - p_i\|}\cdot\omega_{\min}\cdot r_{i,e} \tag{44}$$

$$u_{i,e}^\gamma = -c_{\gamma e}(v_i - v_i^{\text{ref}}) \tag{45}$$

$$d(i) = \begin{cases} \|p_{i,E} - 0_E\|, & \text{wenn } (p_{i,E} - 0_E) = k\cdot e_x^E \\ -\|p_{i,E} - 0_E\|, & \text{wenn } (p_{i,E} - 0_E) = -k\cdot e_x^E \end{cases} \tag{46}$$

Figures and tables

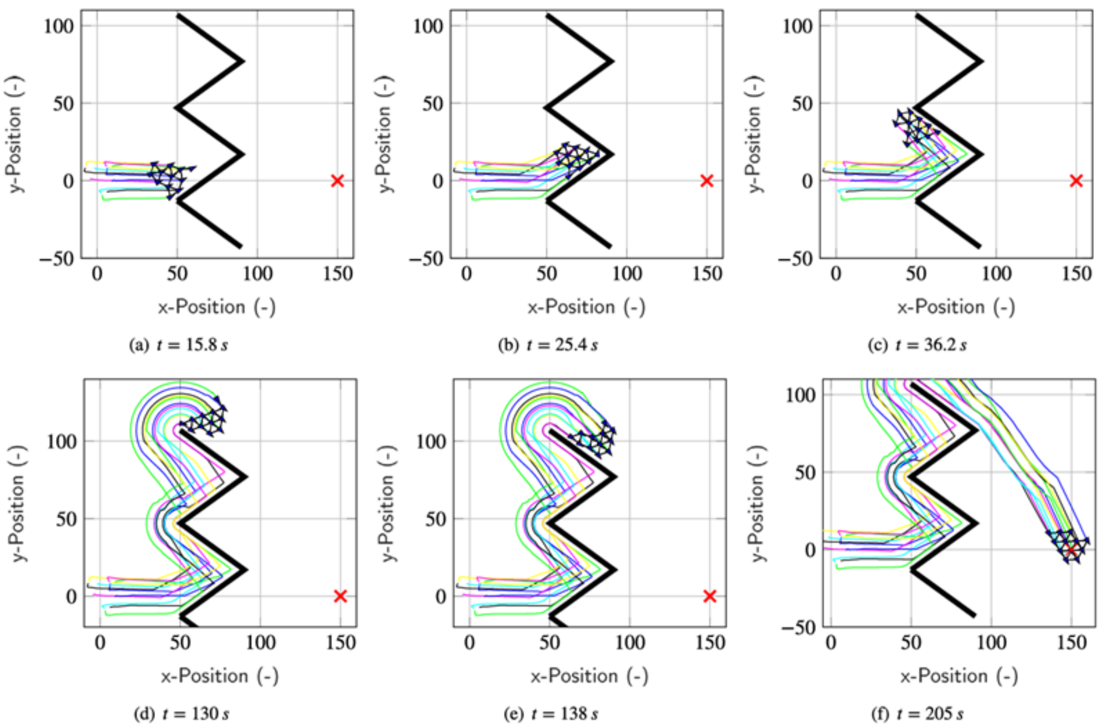


Fig. 8. Sequential snapshots of 12 robots collectively navigating through a zigzag obstacle.

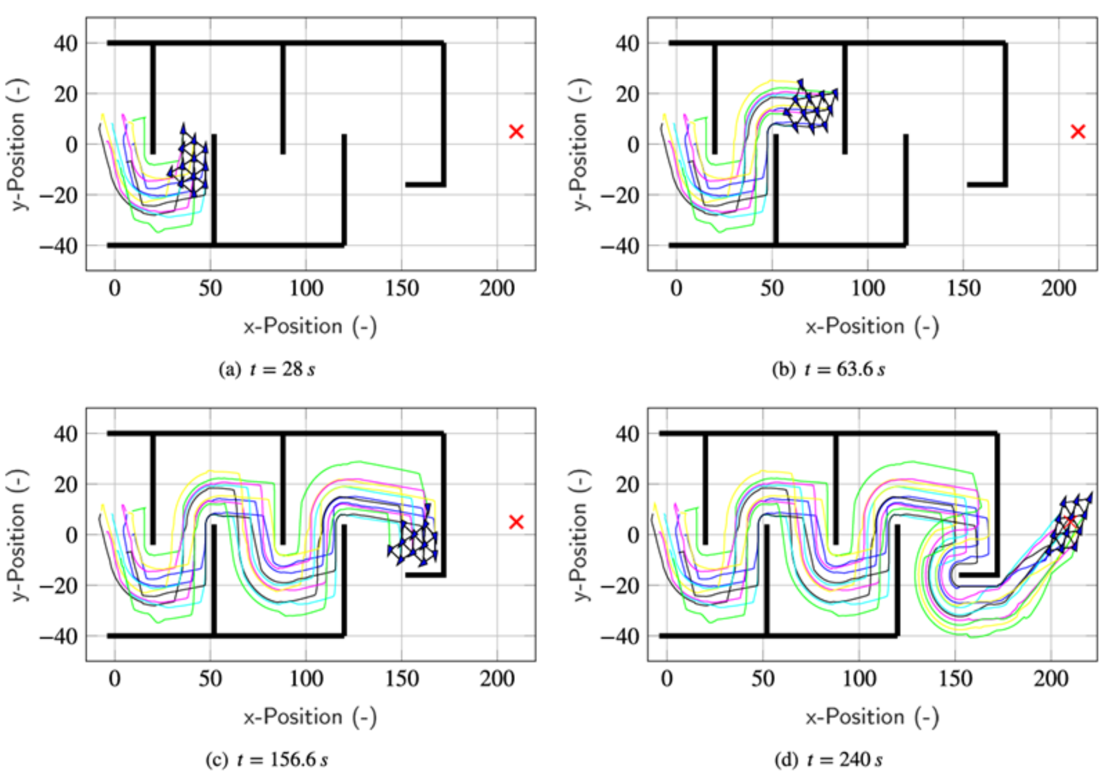


Fig. 9. Sequential snapshots of 12 robots collectively navigating through a corridor.

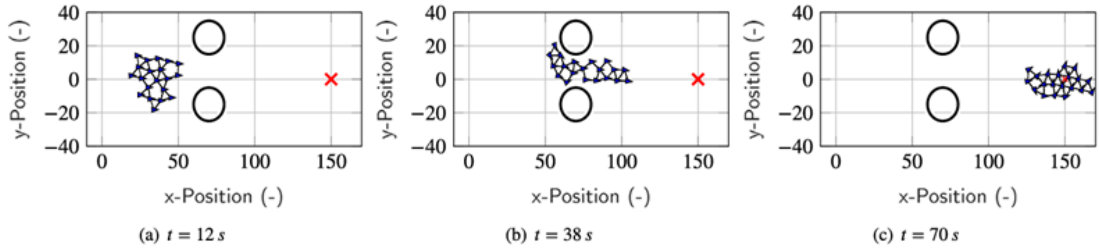


Fig. 10. Sequential snapshots of 20 robots collectively navigating around two circular obstacles.

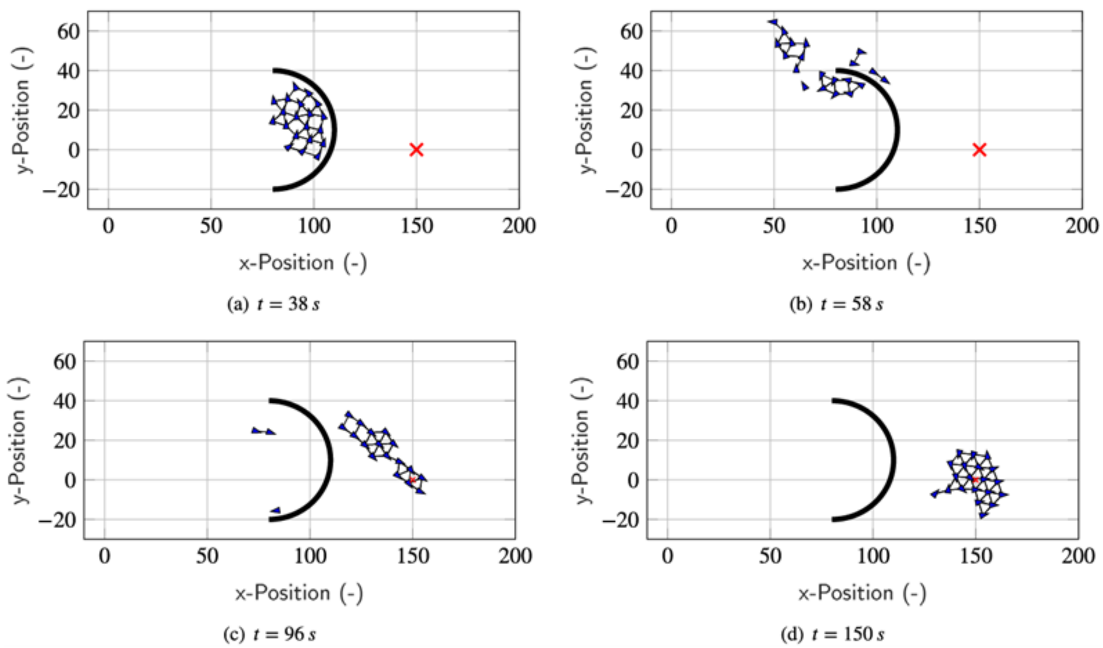


Fig. 11. Sequential snapshots of 20 robots collectively navigating around a semi circular obstacle.

E. Olcay, F. Schuhmann and B. Lohmann / Robotics and Autonomous Systems 132 (2020) 103604

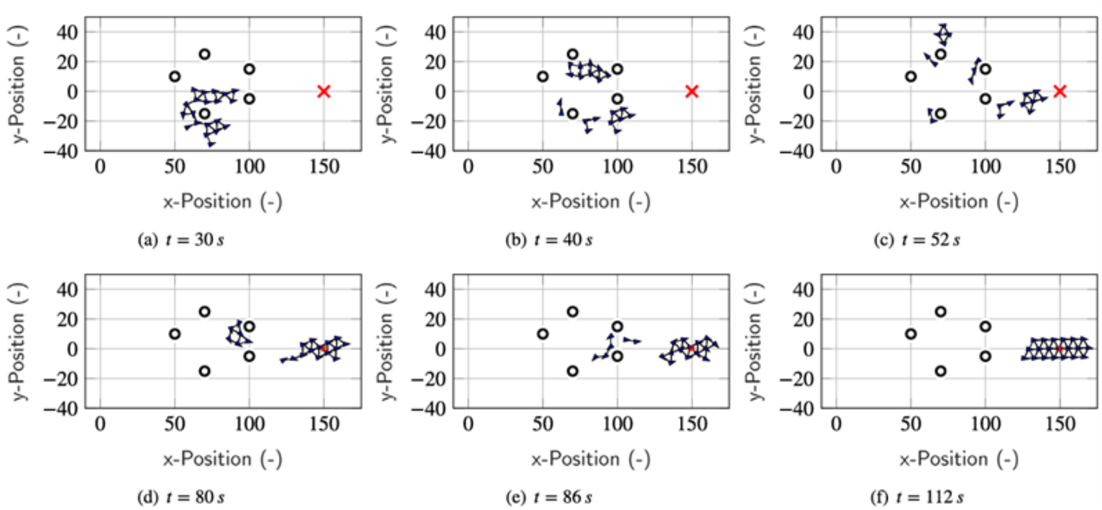


Fig. 12. Sequential snapshots of 20 robots collectively navigating around small circular obstacles.

REFERENCES

[1] E. Olcay, F. Schuhmann, and B. Lohmann, "Collective navigation of a multi-robot system in an unknown environment," *Robotics and Autonomous Systems*, vol. 132, p. 103604, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889020304449

[2] D. P. Stormont, "Autonomous rescue robot swarms for first responders," in *CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, 2005*. IEEE, null, pp. 151–157.

[3] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1987, pp. 25–34.

[4] A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "An analytical approach to avoid obstacles in mobile robot navigation," *International Journal of Advanced Robotic Systems*, vol. 10, no. 6, 2013.