A Dynamic Optimization Framework for Robust Multi-Target Trapping with Self-Organized Swarm Robots

Ziqing Zhou, Linqiang Hu, Yuning Chen, Yan Li, Zhongmeng Xiong, Hongda Zhang, Yi Liu, Chunlei Meng, Bowen Liu, Chun Ouyang*, Zhongxue Gan*, Dunzhao Wu, Zhihua Nie

This PDF file includes:

Z. Zhou, Y. Chen, L. Hu, H. Zhang, Y. Liu, C. Meng, B. Liu, C. Ouyang, Z. Gan, and Z. Nie are with the School of Intelligent Robotics and Advanced Manufacturing, Fudan University, Shanghai, China. Y. Li is with the School of Information Engineering, Zhejiang Ocean University, China. Z. Xiong is with the Children's Hospital of Fudan University, Shanghai, China. D. Wu is with Jiangling Motors Corporation, China.

This study was partially supported by Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103). This study was also supported by (1) Shanghai Engineering Research Center of AI & Robotics, Fudan University, China, and (2) Engineering Research Center of AI & Robotics, Ministry of Education, China.

*Dr. C. Ouyang, oy_c@fudan.edu.cn; Dr. Z. Gan, ganzhongxue@fudan.edu.cn

I. SUPPLEMENTARY TABLE

Table S1. Working ranges of the DODM model's optimized parameters. In the comparative experiments, the baseline algorithm DM uses the same model as described in the original paper.

Parameter	Description	Parameter range
k_{iic}	The strength of the inter-robot collision avoidance	[40,150]
k_{ioc}	The strength of the obstacle collision avoidance	[80,300]
k_{itc}	The strength of the robot-target collision avoidance	[40,200]
k_{rob}	The strength of the density-based inter-robot interaction	[40,150]
k_{tar}	The strength of the density-based robot-target interaction	[500,1500]
ρ_0	The reference density parameter	[0.05,0.15]

Table S2. Parameter settings of swarm robots and targets. The parameter values used for the DODM swarm in each experiment are consistent with those of the DM swarm.

Parameter	Description	Value	Unit
Δt	Time step	0.05	s
N	Number of agents	12-60	-
M	Number of targets	2-10	-
v_{max}	Maximum robot speed	5	m/s
u_{max}	Maximum robot acceleration	20	m/s^2
v_{max}^{tar}	Maximum target speed	3	m/s
u_{max}^{tar}	Maximum target acceleration	20	m/s^2
R_{sense}	Perception range	10	m
α	Acceleration coefficient for robots	5	-
β	Damping factor for robots	10	-
α_{tar}	Acceleration coefficient for targets	0	-
β_{tar}	Damping factor for targets	10	-
θ_{conv}	Threshold for trapping completion	0.07/0.1	-
k_{esp}	The strength of the target escape control	20	-
k_{rep}	The strength of the inter-target collision avoidance	20	-
k_{obs}	The strength of the target-obstacle collision avoidance	100	-
d_{desire}^{flo}	Preferred distance between agents	2	m
d_{danger}^{flo}	Dangerous distance between agents	1	m
d_{desire}^{obs}	Preferred interval between agent and obstacles	3	m
d_{danger}^{obs}	Dangerous interval between agent and obstacles	1	m
d_{desire}^{bou}	Preferred interval between agent and boundary	2	m
d_{danger}^{bou}	Dangerous interval between agent and boundary	1	m

II. SUPPLEMENTARY NOTE

Note S1. Dynamics of targets.

We introduced the improved density-based swarm control algorithm in the main text. Here, we present the dynamics of the target. The control input of the target can be divided into 4 components.

(1) Self-driven control (u_i^{self}): The self-driven control force for each target is proportional to its velocity and is influenced by a constant factor, α_{tar}, and a damping factor, β_{tar}. This control force encourages the target to continue moving in its current direction while considering the magnitude of its velocity. The formula for the self-driven control is given by:

$$u_i^{self}(t) = (\alpha_{tar} - \beta_{tar} \cdot \|\mathbf{v}_i^{tar}\|^2) \cdot \mathbf{v}_i^{tar} \tag{1}$$

In this equation, $\mathbf{v}_i^{tar}$ represents the velocity vector of the i-th target. The term α_{tar} is a constant that scales the target's velocity, while β_{tar} is a damping factor that reduces the force as the target's velocity increases. This allows the target to maintain a consistent motion without accelerating indefinitely, effectively preventing unrealistic speeds in the system. The self-driven control force thus helps the target follow its current velocity while moderating its speed through the damping effect, allowing for smooth and stable motion.

(2) Escape control (u_i^{esp}): The escape force models the evasive behavior of a target when encountering a robot, resembling a predator-prey dynamic. In this scenario, the target avoids being surrounded by the robot. The escape force is calculated based on the relative positions and distances to the robot, with a smoothing factor to prevent abrupt changes in movement.

$$u_i^{self}(t) = \sum_{j=1}^{N} \frac{k_{esp} \cdot \|\mathbf{r}_i^{tar} - \mathbf{r}_j\|}{(\|\mathbf{r}_i^{tar} - \mathbf{r}_j\| + d_0)^2} \cdot (\mathbf{r}_i^{tar} - \mathbf{r}_j) \tag{2}$$

Here, $\mathbf{r}_i^{tar}$ represents the position of the i-th target, while $\mathbf{r}_j$ denotes the position of the j-th robot. The term k_{esp} is the escape force constant, which controls the strength of the escape behavior, and d_0 is a small smoothing parameter used to avoid singularities when the distance between the target and the robot is very small. The distance $\|\mathbf{r}_i^{tar} - \mathbf{r}_j\|$ is the Euclidean distance between the target and the robot. This formula calculates the escape force for the i-th target by summing the forces exerted by all robots in its vicinity, with the magnitude of each force being influenced by the relative distance between the target and the robot. The smoothing factor ensures that the escape force remains finite even when the distance becomes very small, preventing abrupt or unrealistic behavior.

(3) Inter-target collision avoidance (u_i^{rep}): Targets are repelled from each other when they come too close. This repulsive force helps prevent collisions between the targets by ensuring that they maintain a minimum distance from each other. The repulsive force is calculated based on the inverse square of the distance between the targets, which means the force becomes stronger as the targets approach one another. The formula for the inter-target collision avoidance is given by:

$$u_i^{rep}(t) = \sum_{j=1}^{M} \frac{k_{rep}}{\|\mathbf{r}_i^{tar} - \mathbf{r}_j^{tar}\|^2} \cdot (\mathbf{r}_i^{tar} - \mathbf{r}_j^{tar}) \tag{3}$$

In this equation, k_{rep} is a constant that determines the strength of the repulsive force. The repulsive force for the i-th target is calculated as the sum of the forces exerted by all other targets in the system. This ensures that the targets avoid one another when their distance becomes too small, helping to maintain a safe separation and avoid collisions.

(4) Target-obstacle collision avoidance (u_i^{obs}): This force governs the movement of the targets to avoid collisions with obstacles in their environment. The repulsive force is calculated based on the distance between the target and the obstacle, with the strength of the repulsion decaying as the distance increases. This ensures that the target will be repelled more strongly when it is closer to an obstacle, helping to prevent collision. The formula for target-obstacle collision avoidance is given by:

$$u_i^{obs}(t) = \sum_{j=1}^{M} \frac{k_{obs}}{\|d_{\text{to\_gap}}\|^2} \cdot v_{tar\_obs\_unit} \tag{4}$$

In this equation, $\|d_{\text{to\_gap}}\|$ represents the distance between the i-th target and the obstacle. The term k_{obs} is a constant that determines the strength of the repulsive force, and $v_{\text{tar\_obs\_unit}}$ is the unit vector that points from the target toward the obstacle. The repulsive force for the i-th target is calculated as the sum of forces exerted by all obstacles in the vicinity, with each force decreasing in magnitude as the distance to the obstacle increases. This formula ensures that the target will avoid obstacles by generating a repulsive force that guides the target away from the obstacles as it moves.

(5) Position and Velocity Update: After the forces are calculated, the target's velocity and position are updated as follows:

$$v_i^{tar} = v_i^{tar} + (u_i^{self} + u_i^{esp} + u_i^{rep} + u_i^{obs}) \cdot dt \tag{5}$$

The target's position is updated based on the velocity:

$$r_i^{tar} = r_i^{tar} + v_i^{tar} \cdot dt \tag{6}$$

Note S2. The equations of fitness functions.

(1) The positional uniformity metric $C_{pos}(t)$ is calculated as:

$$C_{pos}(t) = \frac{1}{M}\sum_{k=1}^{M}(1 - \|\frac{1}{\|N_{tar}^{k}\|}\sum_{i\in N_{tar}^{k}}\frac{\mathbf{r}_i - \mathbf{r}_k^{tar}}{\|\mathbf{r}_i - \mathbf{r}_k^{tar}\|}\|) \tag{7}$$

where N_{tar}^{k} represents the set of robots interacting with target k, and $\|N_{tar}^{k}\|$ is the number of such robots. This metric reaches its maximum value of 1 when robots are perfectly distributed around targets.

(2) The numerical distribution metric $C_{num}(t)$ evaluates how closely the actual robot allocation matches the ideal distribution:

$$C_{num}(t) = \frac{1}{M}\sum_{k=1}^{M}(1 - \min(1, \frac{|n_{act} - n_{exp}|}{n_{exp}})) \tag{8}$$

where $n_{exp} = N/M$ is the expected number of robots per target, and n_{act} is the actual number of robots around target k.

(3) The connectivity metric $C_{con}(t)$ measures the proportion of robots engaged in target interactions:

$$C_{con}(t) = \frac{n_{inter}}{N} \tag{9}$$

where n_{inter} is the number of robots interacting with at least one target.

(4) The safety metrics ($C_{ii}(t)$, C_{it}(t), and $C_{io}(t)$) evaluate collision avoidance by measuring the minimum distances against desired safety thresholds:

The robot-robot collision avoidance metric $C_{ii}(t)$ evaluates the minimum distance $d_{ii}(t)$ between any two robots:

$$C_{ii}(t) = S(d_{ii}(t), d_{desire}^{flo}, d_{desire}^{flo} - d_{danger}^{flo}) \tag{10}$$

where $d_{desire}^{flo} = 2$ is the desired separation distance between robots, and $d_{danger}^{flo} = 1$ is the minimum safety distance.

The robot-target collision avoidance metric $C_{it}(t)$ assesses the minimum distance $d_{it}(t)$ between any robot and target:

$$C_{it}(t) = S(d_{it}(t), d_{desire}^{it}, d_{desire}^{it} - d_{danger}^{it}) \tag{11}$$

where $d_{desire}^{it} = 2$ is the desired robot-target separation, and $d_{danger}^{it} = 1$ is the critical safety distance.

The obstacle avoidance metric $C_{io}(t)$ measures the minimum distance $d_{io}(t)$ between any robot and obstacle:

$$C_{io}(t) = S(d_{io}(t), d_{desire}^{obs}, d_{desire}^{obs} - d_{danger}^{obs}) \tag{12}$$

where $d_{desire}^{obs} = 3$ is the preferred robot-obstacle separation, and $d_{danger}^{obs} = 1$ is the minimum safe distance.

All safety metrics use the smoothed transition function S defined as:

$$S(x, x_0, d) = \begin{cases} 0, & \text{if } x < (x_0 - d) \\ 1 - 0.5(1 - \cos(\frac{\pi}{d}(x - x_0))), & \text{if } (x_0 - d) \le x < x_0 \\ 1, & \text{if } x \ge x_0 \end{cases} \tag{13}$$

Note S3. The equations of the three minimum distance metrics.

In our safety evaluation, $d_{ii}(t)$, $d_{it}(t)$, and $d_{io}(t)$ represent the minimum pairwise distances within the swarm at time t, defined as:

$$d_{ii}(t) = \min_{i,j\in\{1,2,\ldots,N\}, i\neq j} \|\mathbf{r}_i(t) - \mathbf{r}_j(t)\| \tag{14}$$

$$d_{it}(t) = \min_{i\in\{1,2,\ldots,N\}, k\in\{1,2,\ldots,M\}} \|\mathbf{r}_i(t) - \mathbf{r}_k^{tar}(t)\| \tag{15}$$

$$d_{io}(t) = \min_{i\in\{1,2,\ldots,N\}, o\in\{1,2,\ldots,P\}} (\|\mathbf{r}_i(t) - \mathbf{c}_o\| - R_o) \tag{16}$$

These metrics capture the worst-case proximity scenarios between robots, targets, and obstacles, serving as critical indicators for collision risk assessment.

Note S4. Computational complexity analysis.

The computational complexity of our DODM framework can be analyzed component-wise:

1) **Density-based interactions**: For each robot i, computing ρ_i and $\nabla_i \rho_i$ requires evaluating the kernel function for all neighbors within sensing range. If we denote the maximum number of neighbors as n_i, the complexity is $O(n_i)$ per robot.
2) **Collision avoidance terms**: The inter-robot ($\mathbf{u}_i^{iic}$), robot-target ($\mathbf{u}_i^{itc}$), and obstacle avoidance ($\mathbf{u}_i^{ioc}$) components each have complexity $O(n_i)$, $O(m_i)$, and $O(p_i)$ respectively, where m_i is the number of targets and p_i is the number of obstacles within sensing range.
3) **PSO-FOHE optimization**: For the parameter optimization process, the complexity depends on: N_p: the population size in the PSO algorithm; N_s: the number of robots in the swarm; N_{iter}: the maximum number of iterations for convergence; $d = 6$: the dimensionality of the parameter space $\boldsymbol{\theta} = \{k_{rob}, k_{tar}, k_{iic}, k_{itc}, k_{ioc}, \rho_0\}$

The fitness evaluation for each particle requires computing the six component metrics (C_{pos}, C_{num}, C_{con}, C_{ii}, C_{it}, C_{io}) across the entire swarm, yielding a complexity of $O(N_s \cdot (n_i + m_i + p_i))$ per evaluation. For N_p particles across N_{iter} iterations, the total optimization complexity becomes $O(N_p \cdot N_{iter} \cdot N_s \cdot (n_i + m_i + p_i))$.

The history-guided estimation mechanism in PSO-FOHE reduces this computational burden by: Warm-starting optimization with previously successful parameter sets when similar environmental configurations are detected; Requiring fewer iterations to reach convergence in recurring scenarios.

Our empirical evaluations show that with the history-guided mechanism, the effective number of iterations can be reduced by 40-60% in scenarios similar to previously encountered environments, resulting in an effective complexity of $O(N_p \cdot \alpha \cdot N_{iter} \cdot N_s \cdot (n_i + m_i + p_i))$, where $\alpha \in [0.4, 0.6]$ represents the reduction factor due to historical knowledge utilization.

For real-time implementation, we can further constrain N_{iter} to a fixed upper bound (typically 20-30 iterations), ensuring consistent execution time regardless of problem complexity. Our experimental results demonstrate that even with these constraints, the algorithm achieves near-optimal parameter configurations for effective multi-target trapping while maintaining computational feasibility for online deployment on robotic platforms.

Note S5. Convergence analysis.

We now provide a theoretical proof of convergence for our multi-target trapping approach. The proof addresses the stability of the system and guarantees that: (1) robots maintain safe distances from each other, targets, and obstacles; and (2) robots successfully form stable entrapment formations around targets.

Theorem 1 (Safety Guarantee)**.** *For a swarm system operating under the DODM framework with appropriately chosen parameters, the minimum distances between robots, between robots and targets, and between robots and obstacles remain above specified safety thresholds: $d_{ii}(t) > d_{safe}$, $d_{it}(t) > d_{safe}$, and $d_{io}(t) > d_{safe}$ for all $t > T_0$, where T_0 is a finite time.*

Proof: Consider the collision avoidance term between robots i and j:

$$\mathbf{u}_i^{iic} = \sum_{j \in N_{flo}^i} k_{iic} \frac{1}{\|\mathbf{r}_{ij}\|^2} \frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|} \tag{17}$$

As $\|\mathbf{r}_{ij}\| \to d_{safe}$, this term approaches infinity in magnitude and points directly away from robot j. Similarly, for robot-target and robot-obstacle avoidance terms, the repulsive forces increase quadratically as distances approach the safety threshold.

Let us define a Lyapunov function candidate:

$$V(t) = \sum_{i=1}^{N} \sum_{j \in N_{flo}^i, j \neq i} \max\left(0, \frac{1}{d_{safe} - \|\mathbf{r}_{ij}\|}\right) + \sum_{i=1}^{N} \sum_{k \in N_{tar}^i} \max\left(0, \frac{1}{d_{safe} - \|\mathbf{r}_{ik}\|}\right) + \sum_{i=1}^{N} \sum_{o \in N_{obs}^i} \max\left(0, \frac{1}{d_{safe} - \|\mathbf{d}_{io\_gap}\|}\right) \tag{18}$$

This function is positive definite when any distance falls below d_{safe} and zero otherwise. Taking the time derivative:

$$\dot{V}(t) = \sum_{i=1}^{N} \sum_{j \in N_{flo}^i, j \neq i} \frac{\partial V}{\partial \mathbf{r}_{ij}} \cdot \frac{d\mathbf{r}_{ij}}{dt} + \text{similar terms for target and obstacle distances} \tag{19}$$

Substituting the system dynamics and the control law, we can establish:

$$\dot{V}(t) \leq -\alpha V(t) + \beta \tag{20}$$

where $\alpha > 0$ is determined by k_{iic}, k_{itc}, and k_{ioc}, and β represents bounded disturbances from other control terms.

For sufficiently large values of k_{iic}, k_{itc}, and k_{ioc} optimized by PSO-FOHE, we ensure $\alpha V(t) > \beta$ whenever any distance approaches d_{safe}, thus guaranteeing $\dot{V}(t) < 0$ when safety is threatened. This ensures that after some finite time T_0, all distances remain above d_{safe}.

Theorem 2 (Convergence to Trapping Formation)**.** *For a swarm system operating under the DODM framework with appropriately chosen parameters, the system converges to a stable multi-target trapping formation where: (1) robots are distributed around targets with positional uniformity $C_{pos} \to 1$ as $t \to \infty$, and (2) robots achieve balanced numerical distribution around targets with $C_{num} \to 1$ as $t \to \infty$.*

Proof: The density-based interaction terms drive the system toward configurations where $\rho_i \approx \rho_0$ for all robots. Consider the potential function:

$$U(\mathbf{r}, \boldsymbol{\theta}) = \sum_{i=1}^{N} \left(\frac{\rho_i - \rho_0}{\rho_0}\right)^2 \tag{21}$$

This function is minimized when all robots achieve the reference density ρ_0. The gradient of U with respect to robot positions corresponds to the density-based interaction terms in our control law.

Due to the balance between target attraction and robot-robot interaction, the minimization of U subject to safety constraints leads to configurations where robots form symmetric arrangements around targets. When the number of robots is sufficient ($N \geq M \cdot n_{opt}$, where n_{opt} is the optimal number of robots per target), the system partitions into M subgroups with target-centered formations.

The combined density-based target attraction term:

$$\mathbf{u}_i^{tar} = -k_{tar} \frac{1}{\rho_i} \left[\left(\frac{\rho_i}{\rho_0}\right)^3 - 1\right] \nabla_i \sum_{k \in N_{tar}^i} W(\|\mathbf{r}_{ik}\|, R_{sense}) \tag{22}$$

ensures that robots are attracted to targets while maintaining the desired density distribution. The modulation factor $\frac{1}{\rho_i}\left[\left(\frac{\rho_i}{\rho_0}\right)^3 - 1\right]$ creates a natural load balancing mechanism that prevents overconcentration around any single target.

For a given robot distribution, define the imbalance measure:

$$I(t) = \sum_{k=1}^{M} \left| n_k(t) - \frac{N}{M} \right| \tag{23}$$

where $n_k(t)$ is the number of robots around target k at time t.

To prove convergence, we define another Lyapunov function:

$$W(t) = U(\mathbf{r}, \boldsymbol{\theta}) + \gamma I(t) \tag{24}$$

where $\gamma > 0$ is a weighting factor.

Taking the time derivative and analyzing the system dynamics under the optimized parameters, we can show that $\dot{W}(t) < 0$ whenever the system is not in a balanced state. This implies that both $U(\mathbf{r}, \boldsymbol{\theta}) \to 0$ and $I(t) \to 0$ as $t \to \infty$, which corresponds to $C_{pos} \to 1$ and $C_{num} \to 1$.

The convergence rate is determined by the optimization of parameters through PSO-FOHE, which continuously adjusts k_{rob}, k_{tar}, and ρ_0 to maximize the combined fitness function. This adaptive mechanism ensures robust convergence even in the presence of obstacles and dynamic targets.

Therefore, the DODM framework guarantees convergence to stable multi-target trapping formations while maintaining safety constraints.