

Research on CGF Formation Maintenance and Obstacle Avoidance

Siqing Wang ^{1,a} and Yongjun Qiao ²

¹ Naval Aviation University, Yantai, China

² Naval Aviation University, Yantai, China

Abstract. In order to solve the problem of CGF formation avoiding obstacles, avoiding collisions and reach the target point in the simulated battlefield environment, combined with the behaviour-based method, the following algorithm will be improved to control the CGF formation. In this paper, the motion of CGF formation is decomposed into moving to target, maintaining formation, avoiding obstacles, restoring formation, reaching target, and a linear feedback mechanism based on the relative position of leader and follower is introduced to prevent the occurrence of the team. The tangent obstacle avoidance algorithm evades obstacles. The simulation experiment applies the upper-layer behavioural reasoning to guide the structure of the underlying physical model, and the SOAR agent makes the choice of the state of the environment and the behaviour, and the underlying physical model implements the behaviour. The results show that the strategy can keep the CGF formation in the desired formation, avoid internal collisions while avoiding obstacles, and quickly restore the formation after reaching obstacles, reach the target location and achieve good results.

1 Introduction

The computer force generation technology can generate a certain autonomous and intelligent simulation entity in the virtual battlefield, and the program language controls its motion [1]. As the complexity of the battlefield environment continues to increase, the capabilities of combat personnel are increasingly demanding. The individual soldier's ability to perceive, make decisions, and learn is very limited. If a single force has the ability to cooperate, keep the formation in the team, and communicate with other CGF entities, it will help to complete complex tasks [2]. Therefore, it is necessary to study how the CGF entity team maintains the formation in the movement and avoids obstacles. Moreover, the Agent entity maintains the formation in motion with the following advantages: Compared with a single Agent, the Agent team has more ability to acquire more battlefield situations and environmental information; With a certain formation movement, it can enhance the defensive performance and have better ability to defend against external invasion to ensure its own security; It can speed up the completion of tasks, and multiple agents form different formations according to different environments, which greatly improves the operational efficiency.

There are many algorithms for maintaining the formation of mobile groups, such as Leader-Follower, behaviour-based and virtual structure methods [3-5]. The Leader-Follower algorithm selects an Agent entity in the CGF formation and uses it as the leader of the mobile

group. The remaining Agent entities act as followers. By keeping the distance and direction angle error between the follower and the leader at the desired level, the CGF formation maintains a certain tactical formation and completes the control of the formation [6-9]. Based on the behavioural method, the motion of each agent entity is decomposed into moving toward the target, maintaining the formation, avoiding obstacles, so as to control the movement. The artificial potential field method applies the principle of potential field in physics to the formation control algorithm [10-11], and defines the potential field of the target point, obstacles, and other Agent entities in the team. Each agent entity moves in a comprehensive artificial potential field, continuously moves to the weakest point of potential energy, and finally reaches the target point, and realizes the obstacle avoidance function in motion [12-14].

Based on the above analysis, the advantage of the Leader-Follower algorithm is that the design of the controller is relatively simple, and the communication cooperation between the Agent entities is simplified [15]. The disadvantage of this method is that the entire manoeuvring process is too dependent on the leader, and there is no clear motion error feedback between the leader and the followers. If the leader suddenly accelerates because of the change of the trajectory while avoiding the obstacle, the other followers may lose the tracking target because they cannot keep up with the leader, and eventually the formation cannot be formed. And the leader is the core of the algorithm [16]. If it fails or encounters other unpredictable emergencies, it will

cause the entire CGF formation to lose its original formation and the combat mission fails. In order to solve this problem, this paper combines the idea of behaviour-based method to improve the algorithm, and decomposes the whole motion process into the target motion, follow-up control, obstacle avoidance and so on, which reduces the dependence of the followers on the leader. Linear feedback is added to the Leader's controller to avoid the loss of Followers due to obstacle avoidance.

2 Decision mechanism based on SOAR reasoning

For the choice of entity behaviour, this paper introduces the logical reasoning engine SOAR to make the Agent respond in complex situations. The advantage of using this method is that the high-level reasoning can be separated from the underlying algorithm [17]. Only the algorithm design of the behaviour is needed, so that it has better expansion performance, and the efficient reasoning ability of SOAR ensures the real-time performance of the simulation.

SOAR is a logical reasoning tool consisting of three parts: the inference engine, the rule memory, and the work memory. As shown in Figure 1.

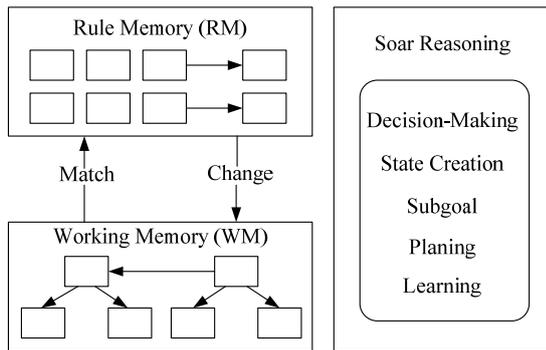


Figure 1. SOAR logical reasoning mechanism.

The inference engine is responsible for creating states, selecting operators, matching rules, behavioural output, etc. Knowledge in the form of IF-THEN is stored in the rule memory. The working memory is responsible for storing short-term knowledge, such as intermediate calculations, intermediate operators, and so on. Among them, goals, operators and states are three important concepts. The goal is the desired result for the current situation. An operator is a kind of logical relationship. When the condition is reached, the state change of the result can be achieved by a certain calculation. State refers to the state in which internal variables are located.

The decision process of SOAR is shown in Figure 2. The situation input is the entrance of the perceptual information. The SOAR selects and applies the operator according to the knowledge in the rule memory, changes the internal state and completes the behaviour output, thereby changing the external environment.

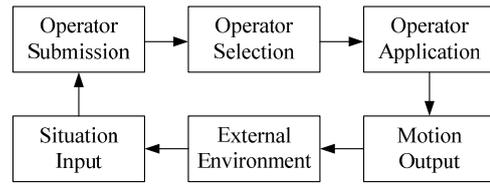


Figure 2. SOAR decision process.

Based on the marching process of the Agent team and the SOAR decision-making framework, the Agent behaviour is divided into the following parts: forward to the target, formation maintenance, obstacle avoidance and collision avoidance. When the Agent does not form any formation, the system first forms a formation, and the Agent advances in the current formation; if an obstacle occurs, the obstacle avoidance behaviour is triggered; if the obstacle is large or dense, the formation switching behaviour is triggered.

3 Formation and maintenance based on Leader-Follower

3.1. Agent dynamics representation

This paper establishes a quadratic dynamics model for the Agent, so the dynamic equation is:

$$\begin{cases} \dot{x}_1 = v_1 \\ \dot{x}_2 = v_2 \\ v_1 = u_1 / m \\ v_2 = u_2 / m \end{cases} \quad (1)$$

In order to more clearly describe the relationship between the agents, this paper presents the CGF formation in matrix form. The current location of the Agent is expressed as: $x = (x_1, x_2)^T$. The current speed is expressed as: $v = (v_1, v_2)^T$. The control input is:

$u = (u_1, u_2)^T$. The state space of $Agent_i$ is: $X_i = \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix}$. Then

the equation of motion of the Agent is obtained as follows:

$$\dot{X}_i = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u_i \quad (2)$$

In the formula, 0 is a zero matrix and I is an identity matrix.

Consider the Lyapunov function:

$$V = \frac{1}{2}(x - x_d)^T (x - x_d) + \frac{1}{2} \dot{x}^T \dot{x} \quad (3)$$

The control rate is:

$$u = -k_p (x - x_d) - k_d \dot{x} \quad (4)$$

Then we can get:

$$\begin{aligned} \dot{V} &= (x - x_d)^T \dot{x} + \dot{x}^T \ddot{x} = (x - x_d)^T \dot{x} + \dot{x}^T u \\ &= \dot{x}^T (x - x_d - k_p (x - x_d)) - k_d \dot{x}_i^T \dot{x}_i \end{aligned} \quad (5)$$

In the formula: x_d is a constant, let k_p be 1, then:

$$\dot{V} = -k_d \dot{x}_i^T \dot{x}_i \quad (6)$$

In the formula, because k_d is a positive number, V is positive and \dot{V} is negative, and the motion of the Agent is gradually stabilized.

3.2 CGF formation and maintenance

3.2.1 Leader moves toward the goal

The Leader is responsible for moving toward the goal, and its speed direction is always along the direction of the Leader's connection with the target point. Leader's speed deviation from the target:

$$\theta = \arcsin \frac{y_0 - y_l}{\sqrt{(x_0 - x_l)^2 + (y_0 - y_l)^2}} \quad (7)$$

In the formula: $(x_0, y_0)^T$ is the target point position and $(x_l, y_l)^T$ is the leader position. Leader speed is:

$$V = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} v_{value} \quad (8)$$

3.2.2 Formation and maintenance

When the SOAR reasoning mechanism is output to form a formation, followers move toward the leader according to the control algorithm and form a certain formation. The expected position difference between $Agent_i$ and leader is:

$$\Delta X_{i,d} = X_{i,d} - X_{l,d} = \begin{bmatrix} x_{i,d} \\ y_{i,d} \end{bmatrix} \quad (9)$$

At some point, the actual position difference between $Agent_i$ and leader is:

$$\Delta X_{i,T} = X_{i,T} - X_{l,T} = \begin{bmatrix} x_{i,T} \\ y_{i,T} \end{bmatrix} \quad (10)$$

Then the formation error of $Agent_i$ during the movement is:

$$e_i = \Delta X_{i,T} - \Delta X_{i,d} \quad (11)$$

By making the tracking error E_i tend to zero, the CGF formation can be trended toward the desired formation. Select the controller as follows:

$$U_i = -k_p \times (\Delta X_{i,T} - \Delta X_{i,d}) - k_d \times \dot{X}_i \quad (12)$$

3.3 Analysis of Agent Tangent Avoidance Algorithm

During the movement, if the distance between the Agent and the obstacle is less than the previously set threshold, obstacle avoidance measures will be taken. The principle of the tangent obstacle avoidance algorithm is to make the Agent move along the inscribed line between the obstacles and avoid them, as shown in Figure 3.

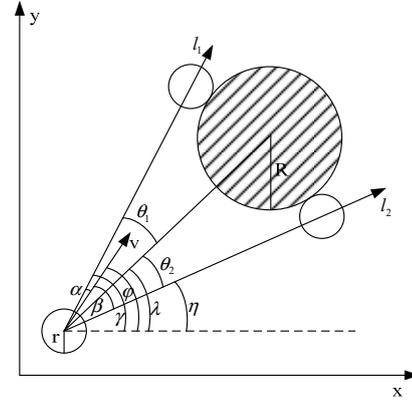


Figure 3. Schematic diagram of tangential obstacle avoidance. The physical meaning of each symbol and variable in the figure is shown in Table 1:

Table 1. The physical meaning of symbols and variables.

r	Radius of Agent
R	Radius of Agent obstacle
v	Speed of Agent
α	The angle between the Agent's speed direction and l_1
β	The angle between the Agent's speed direction and l_2
γ	The angle between l_1 and the positive direction of the x-axis
η	The angle between l_2 and the positive direction of the x-axis
φ	The angle between the direction of motion of the Agent and the positive direction of the x-axis
λ	The angle between the connection between the Agent and the obstacle and the positive direction of the x-axis
θ_1	The angle between Agent and obstacle's connection and l_1
θ_2	The angle between Agent and obstacle's connection and l_2

From Figure 3:

$$\theta_1 = \theta_2 = \arcsin \frac{r + R}{\sqrt{(x_o - x_a)^2 + (y_o - y_a)^2}} \quad (13)$$

In the formula: (x_a, y_a) is the coordinates of the agent center, and (x_o, y_o) is the coordinate of the center of the obstacle, and:

$$\begin{cases} \alpha = \gamma - \varphi \\ \beta = \varphi - \eta \end{cases} \quad (14)$$

In order to ensure that the Agent can completely avoid obstacles, a small angle needs to be added during the deflection process to ensure that the Agent does not collide with the edge of the obstacle. Then the speed direction of the Agent deflection is:

$$\begin{cases} \gamma + \varepsilon (\alpha < \beta) \\ \eta - \varepsilon (\alpha \geq \beta) \end{cases} \quad (15)$$

A linear feedback is added to the leader's controller due to the fact that the speed of the follower may change during the obstacle avoidance process. When follower slows down due to obstacle avoidance, the speed of the leader will change accordingly to prevent the phenomenon of falling behind. The control structure after adding linear feedback is shown in Figure 4:

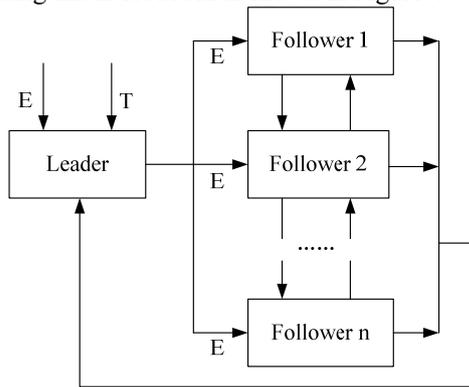


Figure 4. The control structure with linear feedback
 The linear feedback information is:

$$\dot{V}_{Leader} = -k_g \times \left(\max_{i=1}^n \{X_l - X_i - l_i\} \right) - k_i \times V_{Leader} \quad (16)$$

4 Simulation experiment and verification

Set the initial speed of the virtual pilot to 40, and the expected angle difference is $2\pi / 3$, $\Delta X_{i,d} = 40$, $k_p = 7$, $k_d = 1$. The change of the follower1's speed is shown in Figure 5. Figure 6 shows the difference in the angle of movement between the leader and the follower during the tracking process. It can be seen that the follower first speeds up to the leader to keep up with the leader, and then gradually decelerates after reaching the desired distance. The direction of motion turns to the target point, and the speed approaches the leader until it is consistent with the leader. Figure 7 is a trajectory change diagram of the leader and the follower1. It can be seen that the tracking control between the agents is

realized, and the formation and control of the formation are guaranteed.

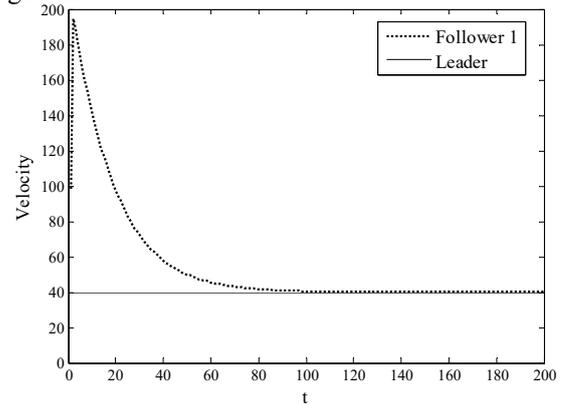


Figure 5. Change of velocity

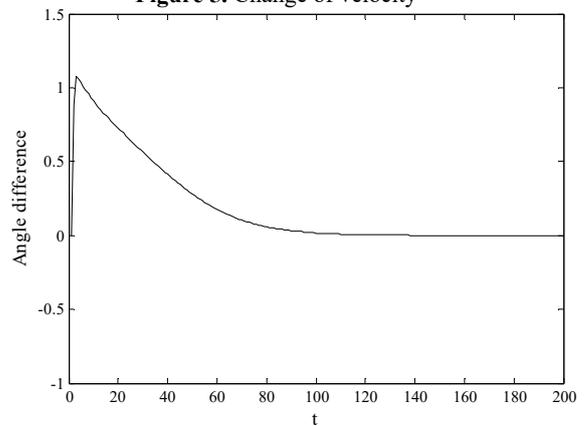


Figure 6. Relative angle

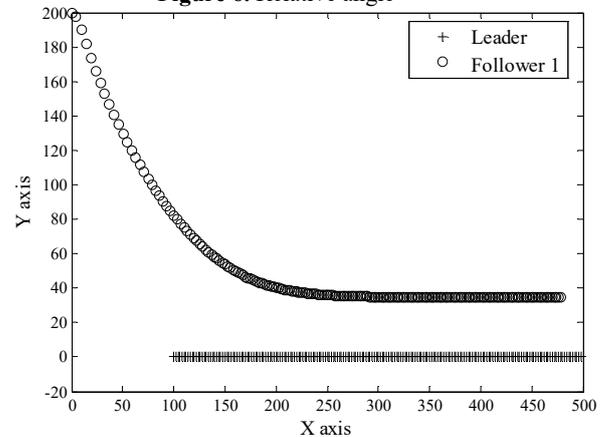


Figure 7. Track map when barrier-free

Taking the triangle formation as an example, four points are randomly generated in the coordinate range of 500×500 , and an obstacle is established as the centre of the circle, and the radius is randomly selected in the range of $[20, 150]$. According to the simulation results, the Agent can avoid obstacles in time and quickly return to the specified formation to ensure the controllability and security of the formation.

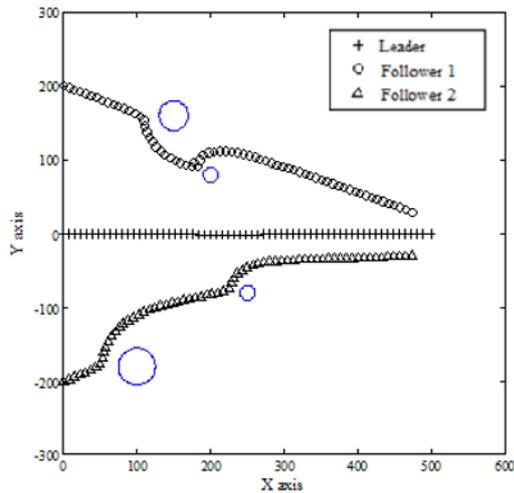


Figure 8. Trajectory map with obstacles

5 Conclusion

In this paper, the existing follower algorithm is combined with the behaviour based method to optimize, the SOAR reasoning mechanism is used for environment perception and behaviour selection, and then the linear feedback mechanism based on the relative position of leader and follower is introduced. Finally, the tangential obstacle avoidance method is used to avoid obstacles. In this paper, the complex formation control problem is transformed into the trailing path of the leader and the tracking of the leader to the leader, which greatly simplifies the complexity of the problem. The simulation results show that the method can effectively make the CGF formation avoid obstacles and expect the formation to reach the target point, which has achieved ideal results.

References

1. LAWTON J R, YOUNG B J, BEARD R W. A decentralized approach to elementary formation maneuvers [C] / / Proc of IEEE International Conference on Robotics & Automation. Piscataway, NJ: IEEE Press, 2000: 2728-2733.
2. LUO Xiao-yuan , HAN Na-ni , GUAN Xin-ping . Leader-following consensus protocols for formation control of multi-agent network [J] . Journal of Systems Engineering and Electronics, 2011, 22 (6):991-997.
3. Balch T, Arkin R C. Behavior-based Formation Control for Multi-robot Teams[J]. IEEE Transactions on Robotics & Automation, 1998, 14(6):926-939.
4. Wang G, Li D, Gan W, et al. Study on Formation Control of Multi-Robot Systems[M]. 2013.
5. Oh K K, Park M C, Ahn H S. A survey of multi-agent formation control[M]. Pergamon Press, Inc. 2015.
6. Zhang Y, Zeng L, Li Y, et al. Multi-robot formation control using leader-follower for MANET[C]// IEEE International Conference on Robotics and Biomimetics. IEEE, 2010:337-342.
7. Yu W, Chen G, Cao M. Distributed leader–follower flocking control for multi-agent dynamical systems with time-varying velocities[J]. Systems & Control Letters, 2010, 59(9):543-552.
8. Chen J, Sun D, Yang J, et al. Leader-Follower Formation Control of Multiple Non-holonomic Mobile Robots Incorporating a Receding-horizon Scheme[J]. International Journal of Robotics Research, 2010, 29(6):727-747.
9. Ghamry K A, Zhang Y. Formation control of multiple quadrotors based on leader-follower method[C]// International Conference on Unmanned Aircraft Systems. IEEE, 2015:1037-1042.
10. Zhen-Zhong Y U, Yan J H, Zhao J, et al. Mobile robot path planning based on improved artificial potential field method[J]. Journal of Harbin Institute of Technology, 2011, 43(1):349-354.
11. Chen Y B, Luo G C, Mei Y S, et al. UAV path planning using artificial potential field method updated by optimal control theory[J]. International Journal of Systems Science, 2016, 47(6):1407-1420.
12. Zhou L, Li W. Adaptive Artificial Potential Field Approach for Obstacle Avoidance Path Planning[C]// Seventh International Symposium on Computational Intelligence and Design. IEEE, 2015:429-432.
13. Wu L, Hui W. Multi-Robot Formation Control and Simulation[C]// Control and Decision Conference. IEEE, 2013:2830-2833.
14. Bishop A N, Deghat M, Anderson B D O, et al. Distributed formation control with relaxed motion requirements[J]. International Journal of Robust & Nonlinear Control, 2015, 25(17):3210-3230.
15. Lorenzen M, Belabbas M A. Distributed local stabilization in formation control[C]// Control Conference. IEEE, 2014:2914-2919.
16. Li F, Ding Y, Hao K. A Dynamic Leader-Follower Strategy for Multi-robot Systems[C]// IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2016:298-303.
17. Laird J E. Soar cognitive architecture[J]. 2012.