Journal of Computers Vol. 36 No. 2, April 2025, pp. 233-248
https://doi.org/10.63367/199115992025043602016

Multi Agent System Control and Scheduling Optimization Method Under Adaptive Event Triggering

Jun-Xian Han[1,2,3], Xiao-Yan Jiang[1], Long-Xue Cheng[1],
Jian-Fang Xue[1], Kai-Xin Cheng[1], and Bo Liu[4*]

[1] Hebei Institute of Mechanical and Electrical Technology,
Xingtai City 054000, Hebei Province, China
junxian_han@163.com, xiaoyan56782024@126.com, clongxue@163.com
jianfangxue8@sina.com, ckx_19950528@sina.com

[2] Hebei Technology Innovation Centre for Intelligent Sensing and Advanced Control of Electromechanical Equipment, Xingtai City 054000, Hebei Province, China

[3] Xingtai Technology Innovation Centre for Intelligent Sensing and Control of Electromechanical Equipment, Xingtai City 054000, Hebei Province, China

[4] School of Automation Engineering, Tangshan Polytechnic University,
Tangshan City 063299, Hebei Province, China
liubo20230101@163.com

Received 29 March 2025; Revised 8 April 2025; Accepted 16 April 2025

Abstract. In the process of intelligent equipment manufacturing, AGV is an important link to realize material flow. With the increase of manufacturing scale and disturbance events, the main research direction of multi-agent systems is to realize the optimization of computing resources and the stability of control system in frequent information interaction and complex associated scenes. Therefore, in this paper, for the collaborative control of multi-agent AGV vehicle system, considering that AGV will be affected by signal interference and network packet loss during driving, as well as the dynamic time-varying characteristics of communication topology between vehicles, semi Markov process is used to describe the time-varying characteristics of communication topology, and adaptive event trigger mechanism is used to adjust the trigger threshold of the system to reduce the network burden, so as to achieve excellent control while saving network resources. Performance, In this paper, the agent model is first created. For the multi-agent object that this paper faces, the improved Q-learning algorithm is used to solve the variables in the process, and all variables are stored in the form of discrete tables. Finally, Matlab is used to simulate the product assembly process of a new energy vehicle electric drive system assembly workshop to verify the feasibility of this method.

Keywords: multi agent system, adaptive events, Q-learning, system stability

1 Introduction

With the rapid development and wide application of network technology, the trend of control system towards networking, distribution, intelligence and integration has become increasingly apparent. In addition, with the vigorous promotion of the national strategy of deep integration of computing, communication and control represented by “made in China 2025” and “German industry 4.0”, a new type of system, multi-agent system (MAS), has begun to appear widely in modern industry. As a typical representative of distributed control system (DCS), MAS has become a hot spot for scholars in various fields. Multi-agent system is a group system composed of multiple autonomous individuals, which can achieve system goals through the coordination among agents, such as multi robot cooperative control. Uav formation, intelligent driving vehicle formation, AGV group automatic operation control, etc. This kind of system is distributed in structure. The whole system is composed of multiple subsystems. There is information interaction and complex correlation between subsystems, and they may also be coupled with each other. The above characteristics lead to the complexity of the model, constraints and objec-

* Corresponding Author

tives of this kind of system, the pervasive limitations of computing resources and physical constraints, and the need to consider the optimization control of subsystems and the global optimization of the whole system. Based on the above background, the traditional PID control and other strategies have been difficult to meet the complex control tasks, and can not effectively solve the multivariable constraint optimization control problem in the actual system. Therefore, finding a more efficient control method has become a key problem to be solved. The system control based on trigger mechanism can better adapt to the current intelligent manufacturing process, so it is favored by more scholars [1].

The system accepts the information and then makes corresponding actions. The information is the input condition for triggering the system. The system based on time triggered mechanism (TTM) is usually implemented in the form of continuous time or cycle. This data transmission mechanism is usually based on an idealized assumption that the network bandwidth is unlimited. However, in the actual scenario, this assumption is not tenable, so the system control method with event as trigger mechanism is more suitable for multi-agent system control in complex dynamic environment. In the process of event trigger mechanism, the core link is the setting of trigger conditions. If the trigger conditions are static parameters, it is inflexible for some special application scenarios, especially in the multi-agent dynamic scenario control. In order to better match the motion characteristics of the research object, some researchers designed the event trigger threshold as a time-varying parameter, so that it can be adjusted adaptively according to the system running state or other external factors, that is, the so-called adaptive event trigger [2].

In the process of intelligent manufacturing, the degree of human participation gradually decreases, and material handling is an essential link in the manufacturing process, so the automated guided vehicle (AGV) came into being. AGV is a wheeled robot, equipped with automatic guidance devices such as electromagnetic or optical sensors, which can automatically travel along the pre planned guidance path according to the set algorithm program, and carry the specified goods from the initial position to the target position [3]. In addition, AGV also has safety protection and various load transfer functions. In recent years, the rapid development of AGV automation technology has brought new vitality to modern logistics manufacturing and promoted the sustainable development of the entire industry. AGV road network design technology, AGV task scheduling technology and AGV path planning technology are the three core aspects of AGV automation technology, which jointly promote the process of material handling automation. Through the application of these technologies, the production efficiency is improved, the production cost is reduced, and the competitiveness of enterprises is also enhanced.

In a multi-agent system, all agents can eventually converge to the same state. This state may involve various attributes of the agents such as their positions, velocities, and directions. This consistency is one of the key conditions for a multi-agent system to complete cooperative control tasks. In a multi-agent system, each agent has the ability to make autonomous decisions and take actions, and they can complete complex tasks through information exchange and coordination. In this process, consistency plays a crucial role. It requires each agent to not only consider its own state but also take into account the states of neighboring agents, and constantly update and adjust its own state to achieve consistency throughout the system. Achieving multi-agent consistency relies on effective communication and coordination mechanisms. Agents need to be able to accurately and timely exchange information to understand each other's states and goals. At the same time, they also need to be able to formulate cooperative action strategies based on this information to achieve coordination and consistency throughout the system. In distributed multi-agent systems, although traditional continuous signal transmission can make the system converge extremely quickly, it consumes a lot of energy. Therefore, event-triggering is introduced to change continuous transmission to discrete. That is, signals are sent only when the system meets certain conditions. This method can effectively save communication resources, so event-triggering is widely used in distributed multi-agent systems.

Therefore, this paper will take the multi AGV cars used in the production of a new energy vehicle as the multi-agent, build the multi-agent model, and then solve the control optimal solution. The work is as follows:

1) Firstly, the agent model is constructed, and the connection relationship between agents is quantified. The multi AGV system is mathematically modeled using the topological relationship, and the stability problem of the complex vehicle queue system is transformed into the problem of solving a set of linear matrix inequalities;

2) An adaptive event triggering mechanism framework is constructed to reduce the consumption of network computing resources;

3) The improved algorithm uses the framework of centralized training decentralized execution. The agents in the network can obtain the global state of all agents through the improved algorithm.

2 Related Work

There are relevant research results in the direction of event triggering and multi-agent cooperative control. Therefore, in the process of summarizing the relevant research results, this paper describes the research results in two directions. In terms of adaptive event triggering, Sun Ying designed a joint recursive filtering scheme to estimate the input and system state of a multi rate information physical system with unknown input. The information transmission between the joint recursive filter and the sensor is controlled by the adaptive event triggering strategy. Then, through some algebraic operations, the sufficient conditions to ensure the bounded covariance of filtering error are obtained. Finally, the fusion estimation scheme of local state estimation is used to improve the effectiveness of the algorithm [4]. This approach enhances the accuracy of state estimation since it is a critical concern in real-time systems where state randomly and continuously changes. This ability to dynamically trigger information update ensures effective utilization of system resources with ensured preservation of estimation accuracy.

Tao Wang, proposed a new constraint contraction scheme and established a new distributed optimization problem; Then, the adaptive event triggering mechanism is designed by using the idea of variable time domain, and the adaptive event triggering distributed model predictive control method with consistent tracking of multi-agent system is proposed, which effectively solves the problems of constraint dissatisfaction and computational resource consumption in the tracking problem of disturbed multi-agent system [5].

Jia Deng of Nankai University, aiming at the consistency problem of second-order multi-agent system under undirected communication topology, proposed a new event triggered control strategy based on adaptive dynamic clock. The multi-agent broadcast its own state information to its neighbors at the trigger time and adaptively reset the clock. The improved control strategy effectively avoided continuous communication. Finally, it was proved that the proposed control strategy can ensure the asymptotic stability of the system by using algebraic graph theory and Lyapunov stability analysis method [6]. The adaptive dynamic clock mechanism optimizes communication efficiency to the maximum, reducing redundant data exchange to a great extent, which is extremely useful in low-bandwidth situations. The algebraic graph theory and Lyapunov stability analysis guarantee that the system does not break down due to such limited communication, thereby making the control strategy robust.

In the aspect of multi-agent system control, Zhuqiqi et al. Applied the multi-agent depth deterministic strategy gradient (MADDPG) algorithm to build the multi-agent cooperative control system. Through the training model, the optimal matching relationship between the conveyor running speed and coal flow was found, and the optimal speed control strategy for energy saving was obtained [7].

In order to improve the robustness of multi-agent control system, Fangyu Li of Beijing University of technology designed a distributed event triggered optimal control method The method includes calculating the joint strategy of communication and control and determining the trigger conditions of communication between multi-agent, so that multi-agent does not need to communicate in real time or periodically, so as to effectively reduce the total amount of data transmission, optimize the communication mechanism and reduce the communication cost; At the same time, in order to enable the agent to avoid other agents and obstacles, a collision penalty term based on exponential function is designed, which makes the agent be punished exponentially in the process of approaching obstacles or other agents, so as to avoid collision [8].

Zhangruyun constructed a multilateral distributed cooperative control framework based on multi-agent system in multimodal network environment. Then, for multiple multi-agent systems with different functions after networking, a communication topology reconstruction method was proposed, and a distributed control protocol was designed to make it more consistent. Finally, a simulation example was used to verify that the proposed method can complete the communication topology reconstruction of multi-agent cooperative control in multimodal network [9].

Xin'ao Zhang from Beijing Institute of Technology compared the advantages and disadvantages of the existing multi-agent cooperative control methods based on temporal logic tasks. Starting from the commonly used control methods that combine temporal logic languages, he sorted out the development trends of three key technologies: human-machine fusion heterogeneous team control methods, robust control of the degree of task violation by the system, and coupling task allocation between human-machine collaborative tasks. He also reviewed the excellent performance of emerging temporal languages such as TSTL in human-machine fusion architectures. He analyzed the scientific and technological bottlenecks existing in current research on this type of cooperative control, including the difficulty in task description, the difficulty in decoupling and allocation, and the large amount of online computation. His analysis conclusions and process provided a framework idea for the construction of this article [10]. The use of temporal logic languages for human-machine collaboration is finding increased importance in enabling multi-agent systems to carry out tasks in a reliable and safe manner. Even though task allocation and computational complexity are hard, the potential that comes from bringing these technologies together for cooperative decision-making is great, particularly where human intervention is needed.

3 Construction of Multi Agent System Control Model

The research object of this paper is the multi AGV scenario in the process of intelligent equipment manufacturing. When task allocation is carried out in the process of AGV handling, the ultimate goal is generally to complete the order task with the highest efficiency. Therefore, the queue cooperative control of AGV vehicles in the process of intelligent manufacturing is particularly important. In the process of queue control, the problem of limited resource consumption caused by frequent communication between AGV vehicles also increases the difficulty of queue control. How to consider the random communication topology while taking into account the communication resources and bandwidth, and design a suitable control algorithm to balance the stability and control performance of the system is the main problem to be solved in this paper [11]. Based on this, this paper studies the vehicle queue system with random communication topology, external disturbances, time-varying spacing and channel attenuation, designs an adaptive event trigger mechanism, and obtains the design algorithm of the controller on the basis of ensuring the stability and robustness of the multi AGV agent system.

In the process of AGV operation, in order to realize the cooperative control between multiple AGV vehicles, the information transmission, processing and execution between vehicles usually need to be carried out according to the given topology, which involves the wireless communication network transmission of a large amount of data.

In the cooperative control of multi-agent AGV vehicle system, the information interaction between vehicles is very important. Because AGV will be affected by signal interference and network packet loss during driving, the communication topology between vehicles is dynamic and time-varying. This paper uses semi Markov process to describe the time-varying characteristics of communication topology. At the same time, in order to reduce the network burden, this paper uses an adaptive trigger mechanism [12], which no longer uses fixed threshold parameters, but can adaptively adjust the threshold parameters according to the operation of the system [13], so as to achieve excellent control performance while saving network resources. At the same time, the hybrid $H\infty$ and passive performance are considered to make the control system robust to disturbances. Based on the above analysis, firstly, the multi AGV system is mathematically modeled using the topological relationship [14], and the stability problem of the complex vehicle queue system is transformed into the problem of solving a set of linear matrix inequalities, and the sufficient conditions for the closed-loop system to be asymptotically stable and meet the given performance index are obtained. The overall multi-agent topology is shown in Fig. 1.

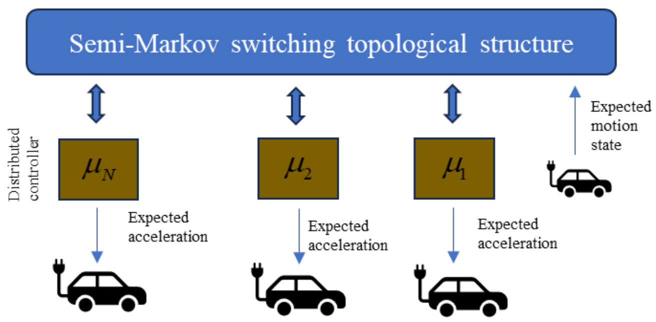

Fig. 1. Schematic diagram of multi-agent topology

3.1 Relationship Model Between Multi Agents

For the multi-agent object in this paper, the information exchange relationship between sub agents in the multi-agent system can be described through the topology structure [15]. Specifically, a multi-agent system containing

m sub agent can be described by directed network topology $Top = \{Nod, Sid, Adj, Wei\}$. in the above set, Nod represents the set of nodes in the structure, and the elements in the set of nodes are composed of various agents. The agents are numbered. Each agent is represented as $IG_{i,i\in[0,m]}$, and the set of nodes is represented as:

$$Nod = \{IG_1, IG_2, \cdots, IG_m\} \tag{1}$$

Agent Sid represents the edge set in the topology network structure. If agent IG_i can obtain the information of agent IG_j , agent IG_j is said to be the neighbor of agent IG_i . at this time, there is edge $\{IG_i, IG_j\} \in Sid$, otherwise, $\{IG_i, IG_j\} \notin Sid$. The edge set Sid collects the communication and communication between sub agents in the whole multi-agent system.

Matrix Adj represents the adjacency matrix, which is expressed as follows:

$$Adj = \left(x_{IG_{i,j}}\right) \tag{2}$$

The element of adjacency matrix is $x_{IG_{i,j}}$, and the values of $x_{IG_{i,j}} \in Nod$ and $x_{IG_{i,j}}$ are related to whether agent IG_i can obtain the information of agent IG_j , The value of $x_{IG_{i,j}}$ is shown as follows:

$$x_{IG_{i,j}} = \begin{cases} 1 & \{IG_i, IG_j\} \in Sid \\ 0 & \{IG_i, IG_j\} \notin Sid \end{cases} \tag{3}$$

Define Wei as the weight of directed edge $\{IG_i, IG_j\}$, and the weight is expressed as follows:

$$Wei = \left(wei_{IG_{i,j}}\right) = \begin{cases} !0 & \{IG_i, IG_j\} \in Sid \\ 0 & \{IG_i, IG_j\} \notin Sid \end{cases} \tag{4}$$

Based on the above description of the topological structure of multi-agent system, m multi-agent system with bounded disturbance is taken as the target object. At the same time, the agent in the target object has dynamic decoupling continuous time nonlinear characteristics. The sensors and actuators of each sub-agent are connected to the controller through the network channel, and exchange and transmit information between the sub-agent controllers through the communication network. Therefore, the dynamic model of information transmission between each agent is used for modeling [16]. The dynamic model of each agent in the system is defined as:

$$\begin{cases} \dot{s}_i(t) = \phi_i\left(s_i(t), in_i(t)\right) + out_i(t) \\ s_i(0) = s_i^0 \\ t \geq 0 \end{cases} \tag{5}$$

Where, $\dot{s}_i(t)$, $in_i(t)$ and $out_i(t)$ represent the state of agent IG_i, control input signal and external additive disturbance respectively. Firstly, in order to simplify the information transmission relationship between multi-agent, the following assumptions are made in this section:

1) The multi-agent system considered in this paper is the same kind of agent system with the same dimension, and there is a fixed communication network between the agents, which can be applied to formation control and cooperative tracking control. For sub agent IG_i, its neighbors are agents that can receive information from each other, and the index set is defined as $Index_{IG_i}$.

2) It is assumed that each agent has at least one neighbor and has the same sampling period. Although the linearized model has been widely used in the design of control system, most actual systems have inherent nonlinear characteristics. The prediction based on the linear model may lead to a large number of prediction errors, which will lead to the stability of the closed-loop system. Therefore, it is necessary to consider the nonlinear model to improve the accuracy of the future dynamics prediction of the system. Secondly, generally speaking, the nonlin-

ear model cannot be discretized accurately. Even if only the continuous differentiable nonlinear system is considered, there may be a large deviation between the discretized system model and the original system model due to the existence of nonlinearity.

3.2 Adaptive Event Trigger Mechanism Framework

In previous systems, agents engaged in continuous communication and information sampling. Due to the limited resources, such frequent interaction behavior was a significant waste of resources and could easily reduce the lifespan of the controller. To address this issue, the event-triggered control strategy was introduced. The event-triggered control strategy is designed for a control system, where the trigger conditions for the system controller are set in advance by humans. When the trigger conditions are met, the controller of the system will be updated, and the agents will update their current states. The controller only works at the moment of event triggering. For multi-agent systems, there are two types of event-triggering methods: centralized event triggering and distributed event triggering. Centralized event triggering refers to the situation where, for a multi-agent system, a unified state error threshold condition is set in advance for all agents. Once an agent meets the set trigger condition, all agents in the multi-agent system will perform actions uniformly, and the sampled data and control inputs will also be updated uniformly. Then, each agent that has updated its state will pass the latest state to other agents in the system. Finally, the set trigger threshold condition is reset to 0. The difference between distributed event triggering and centralized event triggering is that, in a multi-agent system, instead of setting a unified state threshold condition, a separate event-triggered state error threshold condition is set for each agent in the system. If an agent meets the event-triggered state error threshold condition, it will update its state information and then pass the updated state information to other agents for their potential state updates. Therefore, when the state information of one agent in the system is updated, it does not mean that the state information of other agents will also be updated. This is because each agent in the system has its own unique event-triggered condition, which is independent and does not affect each other. Each agent is only related to its own state and the states of its neighboring agents, and does not need to know the global state information.

Adaptive event triggering is used in the event-triggered consensus control of multi-agent systems to handle uncertainties such as unknown system dynamics, unknown parameters, and external disturbances within or outside the agents using adaptive theory. It is also used to estimate unknown global information (i.e., elements of the eigenvalues or eigenvectors of the Laplacian matrix). The adaptive idea is also used to design adaptive gains to ensure system stability or avoid the use of global information. In this section, the adaptive threshold parameters of AGVs are designed to improve the response capability to triggering events.

In the control process of multi-agent, each AGV is connected with each other through wireless network. Each AGV is equipped with sensors and receivers to collect the status information $s_i(t)$ of vehicle IG_i in real time and transmit and receive the information between them. In order to reduce the communication burden, this paper uses the adaptive event trigger mechanism to schedule the communication of data, and determines whether to release the collected state information to the network by using the threshold set by the event trigger as the trigger mechanism [17]. If the given trigger conditions are met, the sampled state information will be released, otherwise the sampled data will be discarded. Therefore, in the whole topology, the closed-loop system of a single AGV vehicle IG_i can be flouted. The schematic diagram of the system structure is shown in Fig. 2.

The k trigger time of the IG_i vehicle is recorded as $t_{IG_i}^k$, the start time is recorded as $t_{IG_i}^k = 0$, and the subsequent trigger time is represented as $t_{IG_i}^{k+1}$, then the adaptive trigger event is represented as:

$$t_{IG_i}^k = \inf\left\{t > t_{IG_i}^k \left| e_{IG_i}^T \psi_{\theta(t)} e_{IG_i}(t) \geq \lambda_{IG_i}(t) \beta_{IG_i}^T(t) \psi_{\theta(t)} \beta_{IG_i}(t)\right.\right\} \tag{6}$$

Where, $\dot{s}_i(t) = s_i\left(t_{IG_i}^k\right)$ represents the state of the IG_i AGV at the latest trigger time, $e_{IG_i}(t) = s_i(t) - \dot{s}_i(t)$ is the trigger error between the current time data and the latest trigger data.

$$\beta_{IG_i}(t) = \sum\left(\dot{s}_i(t) - \dot{s}_j(t) - d_{i,j}(t)\right) \tag{7}$$

$d_{i,j}(t)$ is the expected distance between AGV vehicle IG_i and AGV vehicle IG_j, $\psi_{\theta(t)}$ is the matrix to be solved depending on the mode, $\lambda_{IG_i}(t)$ is the adaptive threshold parameter of AGV Vehicle IG_i, and its time-varying characteristics meet the following relationship:

$$\lambda_{IG_i}(t) = \overline{\lambda}_{IG_i} - \frac{2\left(\overline{\lambda}_{IG_i} - \underline{\lambda}_{IG_i}\right)}{\pi} \arctan\left(\left\| e_{IG_i}(t) \right\|^2\right) \quad (8)$$

Threshold $\lambda_{IG_i}(t)$ can be adjusted adaptively within the range of $\left[\overline{\lambda}_{IG_i}, \underline{\lambda}_{IG_i}\right]$.

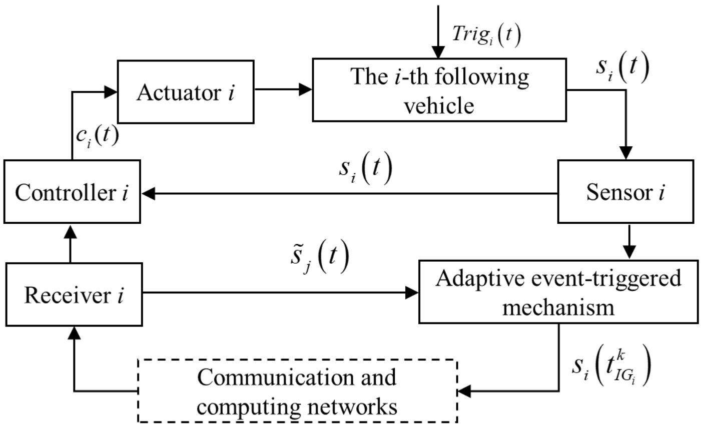

Fig. 2. Closed loop control structure of agent IG_i

4 Design of Coordinated Scheduling Algorithm for Multi Agent System

In most real working scenarios, agents are limited by the scope of perception and communication, and can only use local state information to perceive and adapt to the environment. However, there are certain scenarios, even if the global state information is available, the joint action space will grow exponentially with the increase of the number of agents, that is, the dimension disaster of the joint action space, which makes the reinforcement learning of single agent relatively difficult. In order to improve the coordinated operation method of multiple AGV vehicles in the dynamic equipment manufacturing scene, this paper designs an improved algorithm based on the Q-learning algorithm framework [18]. The overall structure of the improved algorithm adopts the idea of centralized training decentralized execution [19]. In the centralized training phase, the improved algorithm uses the value decomposition method to alleviate the disaster of the dimension of the joint action space.

Centralised Training and Decentralised Execution (CTDE) constitutes the third paradigm of Multi-Agent Reinforcement Learning (MARL). Such algorithms optimize agent policies through centralised training, and these policies are designed to support decentralised execution. Specifically, during the training phase, the algorithm may use globally shared information among all agents to update policies; while in the execution phase, each agent selects actions based solely on its local observations, thus achieving a fully decentralised deployment. This approach aims to integrate the advantages of both centralised training and decentralised execution.

The CTDE algorithm is particularly prevalent in the field of MARL because it can adjust the approximate value function in a computationally feasible way by leveraging privileged information. For instance, in a multi-agent actor-critic framework, a policy can be trained that relies on a centralized critic. This critic is adjusted based on the joint observation history, providing more accurate value estimates compared to a decentralized critic that only receives the observation history of a single agent. During the execution phase, since action selection is directly handled by the policy, the value function is no longer involved in the decision-making process. To ensure the feasibility of decentralized execution, the policies of the agents are adjusted based only on their local observation histories. The subsequent sections will delve into various deep MARL algorithms based on the CTDE paradigm, including multi-agent policy gradient methods, value decomposition techniques, agent modeling, and experience sharing mechanisms.

In the execution phase, each agent only needs to select the action according to the Q-value function related to its own action. In addition, each agent sends its perceived local state and the received local state of its neighbors to all its neighbors, so that the agent in the network can obtain the global state of all the agents. The time difference information required by the agent is obtained by the consistency algorithm, and the agent only needs to send the component information of the time difference information to the neighbor, which alleviates the problem of the limited communication ability of the agent. For ease of description, the improved algorithm is called *i−Qlearning* algorithm. The algorithm framework is shown in Fig. 3.

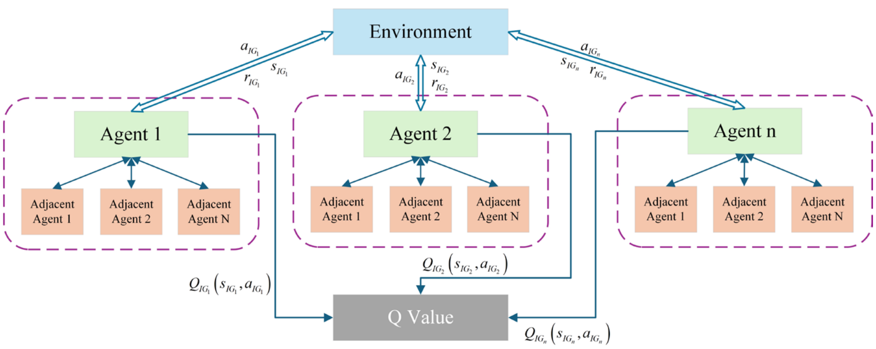

Fig. 3. Structure of improved algorithm framework

The parameters and values involved in the above algorithm structure diagram are shown in Table 1.

Table 1. Symbols and symbol description in structural drawings

Parameter symbol	Symbol description
$\mu[s_{IG_i}]$	Parameter array related to agent state s_{IG_i} in weight parameter
$v[s_{IG_i}]$	Parameter array related to agent state s_{IG_i} in offset parameter
K	Constants greater than 0
δ_{IG_i}	Exploration rate of an action selection of agent
l	step

4.1 Algorithm Design Process

The Q-value function $Q(s_{IG_i}, a_{IG_i})$ of the agent's global state s_{IG_i} and the agent's joint action a_{IG_i} is decomposed into the Q-value function $Q_{IG_i}(s_{IG_i}, a_{IG_i})$ of each agent's own action. Each AGV vehicle obtains the time difference information used by the training model by communicating with the neighbor AGV and using the consistency method. r_{IG_i} is the immediate return obtained by agent IG_i in state s_{IG_i}, and the algorithm uses discrete form to store all variables.

The constraint relationship between the independent Q value of the agent and the joint action Q value is the core of the Q value relationship decomposition. In the training phase, the algorithm in this paper does not simply sum the independent Q values of all agents to fit the joint action value as vdn, but calculates the independent Q value of each agent by training weight parameters and bias weight parameters. Therefore, the joint action Q value update formula is as follows:

$$\begin{cases} Q\left(s_{IG_i}, a_{IG_i}\right) = \sum_{i=1}^{m} \eta_{IG_i}\left(s_{IG_i}\right) \cdot Q_{IG_i}\left(s_{IG_i}, a_{IG_i}\right) + \sum_{i=1}^{m} \omega_{IG_i}\left(s_{IG_i}\right) \\ \eta_{IG_i}\left(s_{IG_i}\right) = K \dfrac{1}{1+e^{-\mu\left[s_{IG_i}\right]}} \\ \omega_{IG_i}\left(s_{IG_i}\right) = K \dfrac{1}{1+e^{-v\left[s_{IG_i}\right]}} \end{cases} \quad (9)$$

Where K is a constant greater than zero, and $\mu[s_{IG_i}]$ and $v[s_{IG_i}]$ are parameter arrays related to s_{IG_i}. Each agent IG_i selects actions in state s_{IG_i} as follows:

$$a_{IG_i} = \begin{cases} \delta_{IG_i} & selectaction \\ 1-\delta_{IG_i} & other \end{cases} \quad (10)$$

Where, δ_{IG_i} represents the probability that the agent selects the action, and its value range is $\delta_{IG_i} \in (0,1)$. after executing action a_{IG_i}, agent IG_i records the immediate return obtained and observes the next state s'_{IG_i} transferred to. Traverse all actions for all agents to obtain values, and select the action $a_{IG_i.\max}$ value with the largest Q_{IG_i} value for each agent, which is expressed as:

$$a_{IG_i.\max} = \arg\max Q_{IG_i}\left(s'_{IG_i}, a'_{IG_i}\right) \quad (11)$$

The communication between agents can be represented by the strong connectivity balance method [20]. Two agents that can communicate with each other are neighbors. ΔT and ΔT_i are used to represent the total time difference information and the component information of each agent's time difference information. The communication between agents is limited, $i - QLearning$ makes each agent get ΔT through consistency method, that is, each agent sends $\Delta T_i(t)$ to each neighbor at time t, and updates $\Delta T_i(t + \Delta t)$ at time $t + \Delta t$, which is expressed as follows:

$$\Delta T_i\left(t+\Delta t\right) = \Delta T_i\left(t\right) + l\sum\left[\Delta T_{IG_{i,j}}\left(t\right) - \Delta T_i\left(t\right)\right] \quad (12)$$

Where, l is the step size, $\Delta T_{IG_{i,j}}(t)$ is the component information between agents IG_i and IG_j. when the system is consistent, t is subject to limit operation:

$$\lim_{t\to\infty} \Delta T_i\left(t\right) = \frac{1}{m}\sum_{i=1}^{m} \Delta T_I\left(0\right) \quad (13)$$

In order to cancel the constant coefficient during gradient descent training, the global loss function is defined as follows:

$$Los\left(t\right) = \frac{\left(\sum_{i=1}^{m} \Delta T_i\left(t\right)\right)^2}{2} \quad (14)$$

At the same time, the gradient descent method is used to update the Q value of each agent, which is expressed as follows:

$$Q_{IG_i}\left(s_{IG_i}, a_{IG_i}\right) = Q_{IG_i}\left(s_{IG_i}, a_{IG_i}\right) - \rho \cdot \frac{\partial Los\left(t\right)}{\partial Q_{IG_i}\left(s_{IG_i}, a_{IG_i}\right)} \quad (15)$$

Where, ρ is the learning rate, after the above process, the pseudo code of the improved algorithm described in this paper is as follows:

```
#Initialize
for each agent i:
Initialize q-valued function Q_i(s_i, a_i) \each agent has its own q-valued func-
tion
Initialize the target Q-value function Q_i_target (s_i, a_i)
Initialize experience playback buffer D_i
Initialize the global state value function v (s) \\Initialize the global target
state value function V_target(s)
#Superparameter
alpha=learning rate
gamma=discount factor
epsilon=exploration rate
batch_size=batch size

update_freq=target network update frequency
#Intensive training phase
for episode = 1 to M:
Initialize the environment and get the initial state s
for t = 1 to T:
#Decentralized execution phase
for each agent i:
Select random action with probability epsilon a_i
Otherwise, select action a_i=argmax_{a_i} q_i (s_i, a_i)
        Execute the joint action a=(a_1, a_2,..., a_n), observe the reward R and
the next state s’
        #Storage experience
for each agent i:
Store (s_i, a_i, r_i, s’\u I) into the experience playback buffer d\u I
        #Update global state value function v (s)
V(s) = max_{a} [sum_{i} Q_i(s_i, a_i)]
        #Update q-valued function
for each agent i:
Randomly sample a batch from d_i (s_j, a_j, r_j, s’\u J)
target = r_j + gamma * V_target(s’_j)
Q_i(s_j, a_j) = Q_i(s_j, a_j) + alpha * (target - Q_i(s_j, a_j)
        #Update target network
if t % update_freq == 0:
for each agent i:
Q_i_target = Q_i
V_target = V
        #Update status
s = s’
        #Attenuation exploration rate
epsilon = epsilon * decay_rate
#Execution phase
for each agent i:
according to Q_i(s_i, a_i), Select action a_i
# Centralized Training Phase
    if len(replay_buffer) > batch_size:
        batch = replay_buffer.sample(batch_size)
        # Calculate target Q-values
        target_q_total = []
        for experience in batch:
            s, a, r, s_next, done = experience
                        if done:
                target = r
            else:
                # Calculate decentralized Q-values for next state
                next_qs = [agent.target_net.predict(get_local_observation(s_next,
```

```
i))
                          for i, agent in enumerate(agents)]
                max_next_actions = [np.argmax(q) for q in next_qs]
                next_q_total = sum([q[a] for q, a in zip(next_qs, max_next_ac-
tions)])
                target = r + gamma * next_q_total
                            target_q_total.append(target)
        # Calculate current Q-values sum (value decomposition)
        current_q_total = []
        for experience in batch:
            s, a, _, _, _ = experience
            q_values = [agents[i].q_net.predict(get_local_observation(s, i))[a[i]]
                      for i in range(num_agents)]
            current_q_total.append(sum(q_values))
        # Update networks using TD error
        loss = mean_squared_error(current_q_total, target_q_total)
        perform_gradient_descent(loss, alpha)

        # Soft update target networks
        for i in range(num_agents):
            agents[i].target_net = tau*agents[i].q_net + (1-tau)*agents[i].target_
net
   # Decay exploration rate
    epsilon *= epsilon_decay
```

5 Simulation Experiment and Result Analysis

Firstly, the impact of adaptive event triggering mechanism and other triggering mechanisms on multi-agent system is analyzed; Finally, a comparative experiment is carried out to illustrate the improvement of the efficiency of the improved algorithm proposed in this section compared with other algorithms in multi-agent communication.

The simulation example is carried out by using MATLAB. According to the intelligent equipment manufacturing production environment and the product assembly process of a new energy vehicle electric drive system assembly workshop, six AGVs, namely six agents, are set. The communication topology between agents is shown in Fig. 4.

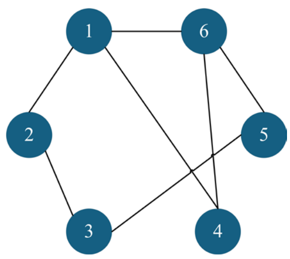

Fig. 4. Schematic diagram of communication structure between agents

The system matrix and input matrix of the selected system are represented as follows:

$$S = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{16}$$

$$I = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \textbf{(17)}$$

Assuming that the communication topology is directionless and connected in real time during the working process of the six agents, the algebraic Riccati equation can be obtained by the unique solution and asking for help:

$$P = \begin{bmatrix} 3.483 & 0.627 & 1.702 \\ 0.626 & 2.391 & -0.879 \\ 1.628 & -0.879 & 1.879 \end{bmatrix} \quad \textbf{(18)}$$

Set the initial value of the agent:

$s_{IG_1} = \begin{bmatrix} 0.1 & 0.2 & 0.3 \end{bmatrix}^T$

$s_{IG_2} = \begin{bmatrix} 0.2 & 0.4 & 0.6 \end{bmatrix}^T$

$s_{IG_3} = \begin{bmatrix} 0.02 & 0.04 & 0.06 \end{bmatrix}^T$

$s_{IG_4} = \begin{bmatrix} 0.017 & 0.034 & 0.065 \end{bmatrix}^T$

$s_{IG_5} = \begin{bmatrix} 0.3 & 0.6 & 0.9 \end{bmatrix}^T$

$s_{IG_6} = \begin{bmatrix} 0.025 & 0.050 & 0.075 \end{bmatrix}^T$

During the simulation experiment, the sampling period is set to 0.002s, and the total simulation time is set to 10s. At the same time, the external interference of each AGV is considered as $out_i(t) = 2\sin(t)$. In the trigger condition, the lower bound of the threshold is $\underline{\lambda}_{IG_i} = 0.02$ and the upper bound of the threshold is $\overline{\lambda}_{IG_i} = 0.3$, by using the LMI toolbox in MATLAB, the feedback gain matrix and adaptive event trigger matrix under each communication topology mode are calculated. The two-dimensional state trajectories of six AGVs are obtained through simulation, as shown in Fig. 5.

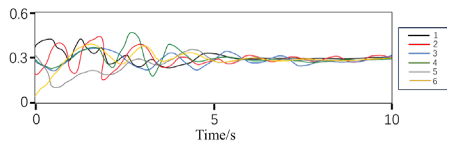

Fig. 5. Trajectory of six AGVs in 2D state

The dimensional consistency error between agents is shown in Fig. 6. It can be seen from the figure that the consistency error of the system converges to 0 in about 8s. This quick convergence indicates that the system's multi-agent coordination is effective, and the agents converge to a shared state quite rapidly. The smooth reduction of error signifies that the control schemes used and the communication protocols between the agents are effective in diminishing dimensional differences. The small error in such a limited time interval confirms the reliability and robustness of the proposed method to maintain agents synchronized, which is highly crucial for multi-agent system stability, particularly in dynamic environments.

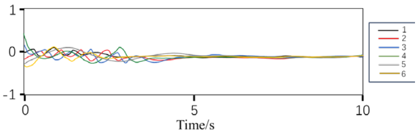

Fig. 6. Dimensional consistency error between agents

The dynamic threshold curve of each agent is shown in Fig. 7.

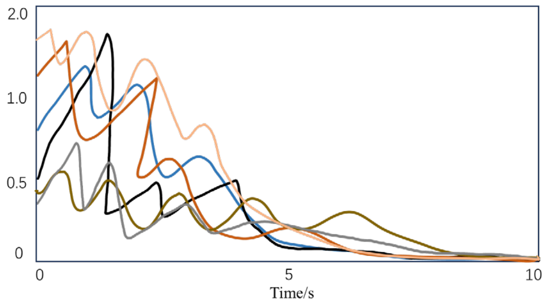

Fig. 7. Dynamic threshold curve of each agent

As can be seen from the figure, the dynamic threshold is adaptively adjusted according to the change of error, which improves the efficiency of event triggering. Since the system is adaptive, the dynamic threshold of each agent is different. With changing error, the threshold also changes accordingly, thereby causing more sensitive event triggering. The threshold is initially high, but as the error decreases, the threshold is reduced in order to avoid redundant communication. This reduces the rate of triggering events, thus improving utilization of resources and reducing computational expense. Dynamic threshold modification between agents on the fly permits the system to respond in real time with no loss of effectiveness.

In order to fully verify the effectiveness of a algorithm, a certain random dynamic disturbance is added in the actual AGV handling process, such as the dynamic disturbance of pedestrians in the manufacturing process and the addition of temporary work orders. The return rate of the improved algorithm and the improved algorithm in the process of information transmission is shown in Fig. 8.

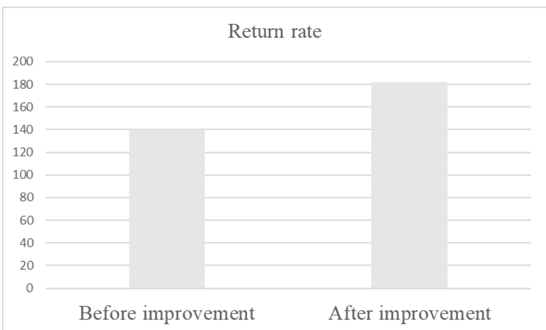

Fig. 8. Comparison of results between maximum expected returns

As can be seen from the figure, the average maximum expected return of the improved algorithm is 42, and the return rate has been improved. At the same time, in order to compare the advantages of the adaptive trigger mechanism and other trigger mechanisms in saving network resources, the static event trigger mechanism is selected for comparison.

The static trigger mechanism sets the thresholds to 0.02 and 0.3, respectively. Scenario 1 and scenario 2 are $\underline{\lambda}_{IG_i} = 0.02$ and $\overline{\lambda}_{IG_i} = 0.3$, respectively. Scenario 3 is a dynamic event trigger mechanism, and the thresholds are adaptively adjusted between 0.02 and 0.3.

In order to compare the control performance under various conditions, the thresholds are 0.02 and 0.2 respectively, and the gain matrix and event trigger matrix under each communication topology mode. The trigger times are shown in Table 2.

Table 2. Symbols and symbol description in structural drawings

	Static event triggering	Static event triggering	Adaptive event triggering
	$\lambda_{IG_6} = 0.02$	$\lambda_{IG_6} = 0.3$	$\underline{\lambda}_{IG_6} = 0.02$, $\overline{\lambda}_{IG_6} = 0.3$
Trigger rate	86.23%	48.36%	43.97%
Number of the agent is triggered	6987	3982	3996

According to Table 2, scenario 1 has the highest number of triggers, but the control effect is the best. Scenario 2 has the lowest number of triggers, but the worst performance. The trigger times of case 3 are much lower than that of case 1 and slightly higher than that of case 2, but the control effect is almost the same as that of case 1. To sum up, the adaptive event trigger mechanism used in this paper can achieve excellent control performance while saving network resources. According to the numerical calculation in the above figure, the maximum triggering rate is 86.23%, so it can be calculated that the saving rate of network resources after using the adaptive triggering mechanism in this paper is 42.26%.

This section analyzes the feasibility of the method described in this paper and achieves optimization in terms of computing resources.

6 Conclusion

Aiming at the stability of multi-agent control system and the problem of AGV cooperative scheduling in the intelligent manufacturing process, this paper first uses the topological relationship to model the multi-agent AGV system, converts the stability problem of complex vehicle queue system into the problem of solving a set of linear matrix inequalities, and obtains the sufficient conditions for the closed-loop system to be asymptotically stable and meet the given performance index. Then according to the control and scheduling problem of multi-agent AGV system in the production process, an intelligent scheduling method based on improved Q-learning is designed. Finally, taking a multi-agent AGV carrier car in the production process of a new energy vehicle as the control object, a simulation environment is constructed by MATLAB, which verifies the feasibility of this method.

This paper has conducted preliminary research on the cooperative control of multiple AGV vehicles in a communication network topology and achieved certain results. However, the depth and breadth of the research still need to be enhanced. Future studies can be conducted in the following directions:

(1) The controller design in this paper is mainly based on idealized conditions of the intelligent manufacturing environment. However, the actual intelligent manufacturing environment where AGVs operate is complex and variable. The current nonlinear model has not fully considered many practical factors. Therefore, future research can focus on how to design a controller suitable for nonlinear vehicle queue systems under fault-tolerant mechanisms and optimization objectives. This is a highly challenging and practically significant task.

(2) This paper models the entire control process as a semi-Markov process (SMP). Although SMP extends the flexibility of traditional Markov processes by allowing the state residence time to follow any distribution, it still has several limitations: First, the model complexity significantly increases, especially when the state space is large or the residence time distribution is complex, making parameter estimation and computation difficult;

second, the selection of residence time distribution in practical applications lacks universality, and incorrect assumptions may lead to model bias; third, most theoretical analyses rely on independence or specific distribution assumptions, limiting their applicability in non-stationary or strongly correlated scenarios; fourth, compared to hidden semi-Markov processes (HSMM), SMP has difficulty handling unobserved states, restricting its application scope. These deficiencies are particularly prominent in data-scarce or high-dimensional environments.

(3) Currently, the control algorithm proposed in this paper is still at the theoretical and simulation verification stage. Transferring the algorithm from the simulation environment to the actual AGV vehicle system is a key step for its application. This requires real vehicle tests of the algorithm in the real intelligent manufacturing environment, and further verification of the algorithm's effectiveness and reliability through the collection and analysis of actual driving data. At the same time, real vehicle tests can help identify problems not considered in the simulation, providing practical basis for algorithm improvement. Therefore, how to deploy the algorithm proposed in this paper in the actual environment to achieve real environment control of multiple AGV vehicles will be the focus of future research.

7 Acknowledgement

Funded by Project: "Multi-Robot Task Allocation and Motion Planning Technologies in Large-Scale Equipment (2023ZC017)".

References

[1] Z.-Y. Xiao, Z.-S. Xia, W.-J. Hong, J. Shi, Decoupling control method based on multi-agent deep reinforcement learning, Journal of Xiamen University (Natural Science) 63(3)(2024) 570-582.

[2] W.-B. Xiong, L. Guo, T.-Y. Jiao, A multi-agent path planning algorithm based on game theory and reinforcement learning, Journal of Shenzhen University Science and Engineering 41(3)(2024) 274-282.

[3] B.-Y. Zhao, L. Wang, K.-C. Huang, Q. Zhou, K. Zhang, Research on Intelligent Unmanned Warehouse Layout Design and AGV Scheduling Optimization Based on Multi-agent Algorithm, Logistics Engineering and Management 46(5) (2024) 18-20+52.

[4] Y. Sun, M.-M. Fu, J.-Y. Mao, G.-L. Wei, Recursive filtering of multi-rate cyber-physical systems with unknown inputs under adaptive event-triggered mechanisms, Frontiers of Information Technology & Electronic Engineering 25(2) (2024) 250-260.

[5] T. Wang, Y. Kang, P.-F. Li, Adaptive event-triggered distributed model predictive control for tracking consensus of multiagent systems, Scientia Sinica (Technologica) 53(11)(2023) 1885-1894.

[6] J. Deng, F.-Y. Wang, Z.-X. Liu, Z.-Q. Chen, Fully distributed consensus control for second-order multi-agent systems based on adaptive dynamic clock communication, Scientia Sinica (Informationis) 53(1)(2023) 97-110.

[7] Q.-Q. Zhu, J.-Z. Li, Q. Shi, J.-C. Liu, D. Hu, Bulk Material Conveying Multi-agent Collaborative Control Based on MADDPG Algorithm, Techniques of Automation and Applications 43(3)(2024) 10-13+34.

[8] F.-Y. Li, J.-Y. Liu, Y.-T. Huang, H.-G. Han, Multi-agent distributed event-triggered optimization control based on deep reinforcement learning, Scientia Sinica (Technologica) 54(10)(2024) 1991-2002.

[9] R.-Y. Zhang, G.-Y. Xiao, Q.-H. Shan, T. Zou, D. Li, F. Teng, Communication topology reconstruction method for multiagent cooperative control in polymorphic networks, Journal on Communications 43(4)(2022) 50-59.

[10] X.-A. Zhang, H. Fang, X.-Y. Zhao, Z.-Y. Chen, W.-L. Ke, Research progress on the collaborative control of the human-machine fusion heterogeneous multi-agent based on temporal logic tasks, Science & Technology Review 42(12) (2024) 167-177.

[11] F.-T. Xiang, J.-R. Luo, X.-Q. Gu, J.-M. Su, W.-P. Zhang, Survey on multi-agent reinforcement learning methods from the perspective of population, Chinese Journal of Intelligent Science and Technology 5(3)(2023) 313-329.

[12] R.-Y. Wang, H.-J. Li, D.-P. Wu, H.-X. Li, Semi-Markov Decision Process-based Resource Allocation Strategy for Virtual Sensor Network, Journal of Electronics & Information Technology 41(12) (2019) 3014-3021.

[13] H.-C. Li, N. Zhang, H.-M. Deng, Tracking Problem of Saturated Systems Based on Adaptive Event-triggered Strategy, Journal of University of Jinan (Science and Technology) 36(5)(2022) 609-615.

[14] B.-B. Zhao, Q. Wang, H.-K. Zhang, J.-X. Xie, L.-W. Wang, Topological relations between a directed line and a directed region with one hole, Science of Surveying and Mapping 48(1)(2023) 214-226.

[15] P.-L. Lu, W.-C. Luo, C.-K. Du, Dynamic event-triggered-based predictive control of multi-agent systems, Control and Decision 39(12)(2024) 3981-3988.

[16] P. Zhang, H.-F. Xue, S. Gao, X. Zuo, Distributed adaptive cooperative tracking control of multi-agent system with weak communication, Systems Engineering and Electronics 43(2)(2021) 487-498.

[17] J.-Q. Li, G.-Q. Zhang, X.-K. Zhang, W.-D. Zhang, Adaptive neural sliding mode control for USV with the hybrid threshold event-triggered mechanism, Control Theory & Applications 40(9)(2023) 1665-1671.
[18] H.-Q. Zheng, Y.-W. Jiang, J.-S. Dai, Optimization of Voltage Regulation and Loss Reduction of Offshore Wind Power Transmission Based on Multi-agent Q-learning in Uncertain Environment, Proceedings of the CSEE 44(20)(2024) 7995-8009.
[19] Q. Xin, Z.-X. Xin, L. Ma, S. Xin, T. Chen, Optimization Design of Interference Strategy and Interference Waveform Based on Two-layer Reinforcement Learning, Guidance & Fuze 44(4)(2023) 35-41.
[20] F.-Y. Yang, Z.-Y. Yu, H.-J. Jiang, D. Huang, Distributed fixed-time optimization for multi-agent systems via event-triggered intermittent communication, Control and Decision 38(5)(2023) 1412-1419.