A Preliminary Study of Fusion ARTs with Adaptively Information Intensity Attenuation Controlling

Zhu Wenxuan[1], Hou Yaqing*[1], Zhang Qiang[1], Ge Hongwei[1], Yang Xin[1], Feng Liang[2], Qu Xinghua[3]
[1]School of Computer Science and Technology, Dalian University of Technology(DLUT), China
[2]College of Computer Science, Chongqing University(CQU), China
[3]School of Computer Engineering, Nanyang Technological University(NTU), Singapore
E-mail: [1]zhuwenxuan@mail.dlut.edu.cn, [1]{houyq, zhangq, hwge, xinyang}@dlut.edu.cn, [2]liangf@cqu.edu.cn, [3]xinghua001@ntu.edu.sg

***Abstract*—Fusion ART is an enhanced version of Adaptive Resonance Theory (ART) which is derived from a biologically-plausible theory of human cognitive information processing. Due to its well-established ability of learning associative mappings across multimodal pattern channels in an online and incremental manner, fusion ART has been widely applied in many real world learning problems. In this paper, we take a Fusion Architecture for Learning, Cognition, and Navigation (FALCON) as the specification and essential backbone of fusion ART and introduce an intensity attenuation controller δ for adaptively adjusting the intensity of information captured from the environment, by taking inspiration from Broadbent-Treisman Filter-Attenuation's perceptual model of environmental attention. Particularly, we propose both an adaptive δ detection algorithm as well as a δ-based pruning algorithm to enhance the learning performance of FALCON while reduce the redundant memory storage incurred by the "detrimental δ". To verify the effectiveness and efficiency of our proposed method, comprehensive experimental studies are carried out on a classical minefield navigation task.**

***Keywords*—Fusion ART, FALCON, Information Intensity Attenuation, Node Pruning, Minefield Navigation Task**

I. Introduction

Adaptive Resonance Theory (ART) [1], [2] is a number of self-organizing neural network models which is derived from a biologically-plausible theory of human cognitive information processing. In the literature, ART principles have been widely applied for pattern recognition, analysis and prediction, as well as for behavioral and neurobiological prediction. Particularly, fusion ART models are direct extensions of single-channel ART which can learn associative mappings across multimodal pattern channels in an online and incremental manner [3]. In past decades, fusion ART has been used in a variety of learning scenarios, i.e., reinforcement learning [4], supervised learning [5], and multimodal learning [6], and enjoyed excellent success across many real world applications, i.e., text categorization [7], non-player character modeling [8], [9] and personal profiling [10].

A Fusion Architecture for Learning, Cognition, and Navigation (FALCON) [3], [4] is a specific instantiation of three-channel fusion ART. Compared with deep neural networks, FALCON has the advantage of achieving the fast learning process in an online manner which indicates FALCON can quickly adapt to inputs that may occur very rarely. Due to its well-established fast and online self-learning capabilities, FALCON has shown great significance for solving problems of low-dimensional input patterns with few data samples, i.e., the online path planning task in minefield navigation domain [11], [12], game AI design in real-time shooting games [13] and homeland defense games [14], virtual world [15], and so on.

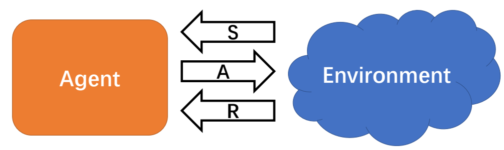

Fig. 1. The learning struture of FALCON-based agent by interacting with the environment.

Learning and memory are the two main foundational functions of human brains, which can be realized by a FALCON-based agent (as shown in Fig. 1) through the following two steps, 1) the agent captures sensory inputs from the environment and makes decisions based on the knowledge in the memory, 2) upon receiving environmental returns based on the behaviors performed, the agent will learn from the feedback and update its memory accordingly [3], [16]. During this process, the FALCON-based agent may obtain environmental information from multiple sensors as sensory inputs. This is similar to the human brain receiving information from multiple senses at the same time. It is noteworthy that we humans usually work at different tasks with different levels of sense involvement. For example, when a person is absorbed in reading, its central neural system mainly receives information from vision and may fail to be able to pay much attention on the loud noises nearby in the meantime. This means that information from the ear is suppressed. Another example is that a person may not notice a friend coming over while enjoying a piece of music, resulting in attenuated visual information .

The sense organs of humans, including eyes, ears and so on, are capable of capturing the external stimuli from the environment. All the senses of the human are being buffeted by numerous internal and external stimuli at the same time. Due to the limited capability of human information processing system, it is impossible for humans to perfect the processing of all

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

sensory stimuli. Hence, people always choose the important ones while ignore the others. Therefore, the core issue of attention is the selective analysis of information. Broadbent-Treisman Filter-Attenuation Model is the perceptual selection model of attention, which assumes that there is a filter between primary analysis and advanced meaning analysis [17], [18]. In the Filter Model, Broadbent believes that the filter allows only one channel's information to reach the level of advanced meaning analysis. But in the Attenuation Model, Treisman believes that the filter not only allows the information of one channel to pass through, but also provides permission for information from the other channels while its intensity in the other channels has been attenuated, weakened [18].

We take our cue from this psychological model and introduce an intensity attenuation controller δ among the information received by the FALCON-based agent from multiple sensors. According to the value of information from different sources to the current task, the intensity of information can be adjusted adaptively with the intensity attenuation controller δ. Generally, a higher intensity of information may lead to the generation of clearer memorial knowledge inside agents' mind universe.

The contributions of this work include the adaptive detection of δ and the introduction of a δ-based pruning algorithm to remove the redundant nodes generated by the "detrimental δ". We modify the code activation process by leveraging the idea from the Broadbent-Treisman Filter-Attenuation Model and propose the new FALCON network architectures denoted by IA-FALCON and P-FALCON for the adaptive algorithm and pruning algorithm respectively. The effectiveness of our idea is investigated on a minefield navigation task wherein an autonomous vehicle needs to reach the target within a specified number of steps while avoid hitting the obstacles (i.e., mines) in the map field. The results demonstrate that the adaptive algorithm with δ-based pruning achieves a higher success rate while incurs a lower memory storage by generating much less neural nodes in agents' mind universe than the benchmark without intensity attenuation controlling.

The rest of the paper is organized as follows. Section 2 introduces the FALCON network architecture. Section 3 proposes the algorithm detecting the intensity attenuation controller δ adaptively and the algorithm pruning nodes based on δ. Section 4 presents the experiment results for verifying the effectiveness of our proposed method. Section 5 summarizes and introduces our future work.

II. Background

A. FALCON Network Architecture

FALCON employs a three-channel architecture based on fuzzy ART operations consisting of a category field F_2 and three input fields, namely F_1^{c1} for the sensory input field, F_1^{c2} for the motor input field and F_1^{c3} for the feedback input field. The dynamics of FALCON are as follows:

- *Input Vectors:* Let $\mathbf{IV} = (\mathbf{S}, \mathbf{A}, \mathbf{R})$ be the input vector where $\mathbf{S} = (s_1, s_2, \ldots, s_n)$ denotes the state vector, and s_i indicates the value of the sensory input i; $\mathbf{A} = (a_1, a_2, \ldots, a_m)$ denotes the action vector, and a_i indicates a possible action i; $\mathbf{R} = (r, 1-r)$ denotes the reward vector, and $r \in [0,1]$.
- *Activity Vectors:* The F_1^{ck} activity vector is denoted by $\mathbf{x}^{ck}$ for $k = 1,2,3$. The F_2 activity vector is denoted by $\boldsymbol{y}$.
- *Weight Vectors:* $\mathbf{w}_j^{ck}$ denotes the weight vector associated with the j^{th} neuron in F_2 layer. The weight vector $\mathbf{w}_j^{ck}$ is updated by the input activity vector $\mathbf{x}^{ck}$ for $k = 1,2,3$.

While the FALCON-based agent performs a sense-move-learn loop, the network operates alternately in two modes, namely, predicting and learning. The detailed algorithm is as follows:

1) Predicting: Given the current state $\boldsymbol{S}$ of the agent, the FALCON network searches the node J matching $\boldsymbol{S}$ through code competition. Then the network reads out the value of the motor input field of node J, so as to select an action. First, initialize the activity vectors of the three input fields to be $\boldsymbol{x}^{c1} = \boldsymbol{S}$, $\boldsymbol{x}^{c2} = \boldsymbol{N}$, where $N_i = 1$ for all i, and $\boldsymbol{x}^{c3} = (1{,}0)$. Then, the algorithm selects the action by code activation, code competition and activity readout.

a) Code activation: Given the activity vectors $\boldsymbol{x}^{c1}$, $\boldsymbol{x}^{c2}$, $\boldsymbol{x}^{c3}$, the choice function T_j calculates the matching degree between the input vector and the weight vector corresponding to the chosen node j. The formula for the choice function T_j is defined as follows:

$$T_j = \sum_{k=1}^{3} \gamma^{ck} \frac{\left|\mathbf{x}^{ck} \wedge \mathbf{w}_j^{ck}\right|}{\alpha^{ck} + \left|\mathbf{w}_j^{ck}\right|}, \tag{1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$ for vectors $\mathbf{p}$ and $\mathbf{q}$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vector $\mathbf{p}$.

b) Code competition: The F_2 node with the highest value of the choice function will be selected. The winner is indexed at J where

$$T_J = \max\{T_j\colon \text{for all } F_2 \text{ node } j\}. \tag{2}$$

When the F_2 node J is selected, $y_J^c = 1$ and $y_j^c = 0$ for all $j \neq J$. This is a winner-takes-all strategy.

c) Activity readout: The node J in F2 field reads the value of its weight vector to the motor input field F_1^{c2} such that

$$\mathbf{x}^{c2} = \mathbf{w}_J^{c2}. \tag{3}$$

The action corresponding to the largest element of the activity vector $\mathbf{x}^{c2}$ is selected.

$$x_I^{c2} = \max\{x_i^{c2} : \text{ for all } F_1^{c2} \text{ node } i\} \tag{4}$$

2) Learning: When the agent receives a positive feedback, the agent will perform code activation, code competition, template matching and template learning to learn the association between the state vector S, the action vector A and the reward vector R. When the agent receives a negative feedback, the agent will learn the association between the state vector S, the complement code of the action vector $\bar{A}$ where $\bar{a_i} = 1 - a_i$ for all i, and the complement code of the reward vector $\bar{R}$ where $\bar{r_i} = 1 - r_i$ for all i. After code activation and code competition, the algorithm will perform template matching and template learning to update the neurons. Initially, there is only one uncommitted neuron in F_2 layer. The neuron weight is initialized to all 1's. When this uncommitted neuron is selected to learn the association, it is committed, and another uncommitted neuron is initialized.

a) Template matching: The template matching process checks whether the weight template for node J is sufficiently similar to the corresponding activity vector $\boldsymbol{x}^{ck}$ for $k = 1,2,3$ in F_1 layer before node J can be used for learning. When the weight vectors in node J are sufficiently similar to the F_1 activity vectors, resonance occurs. To be specific, resonance occurs if the match function m_J^{ck} of the chosen node J meets its vigilance criterion for each channel k:

$$m_J^{ck} = \frac{\left|\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck}\right|}{\left|\mathbf{x}^{ck}\right|} \geq \rho^{ck}. \tag{5}$$

If the match function m_J^{ck} does not meet its vigilance in a certain channel, T_J is set to 0, and the code competition repeatedly selects a new node J until the match function m_J^{ck} meets its vigilance in each channel k.

b) Template learning: Once node J is selected for learning, the weight vector $\boldsymbol{w}_J^{ck}$ will be updated according to the following rules:

$$\mathbf{w}_J^{ck(\mathrm{new})} = (1 - \beta^{ck})\mathbf{w}_J^{ck(\mathrm{old})} + \beta^{ck}(\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck(\mathrm{old})}). \tag{6}$$

III. The Proposed Method

A. Code Activation based on Broadbent-Treisman Filter-Attenuation Model

Based on the Broadbent-Treisman Filter-Attenuation Model, we introduce an attention "filter" before the information from the sensors is transferred to the sensory input fields. We adjust the intensity of signals in the "filter" as the intensity of differing signals from multiple sources is of great difference.

Consider that the intensity of the signals from different types of sensors in sensory input do affect their contributions to the learning task undertaken. We introduce an intensity attenuation parameter δ to adjust the intensity of signals from different sensors, and synchronously adjust the intensity of corresponding parts in the weight vector. Based on the above ideas, we define the choice function as follows:

$$T_j = \frac{\sum_{i=1}^{l} \delta_i \left|\mathbf{x}_i^{c1} \wedge \mathbf{w}_{i,j}^{c1}\right|}{\alpha^{c1} + \sum_{i=1}^{l} \delta_i \left|\mathbf{w}_{i,j}^{c1}\right|}, \tag{7}$$

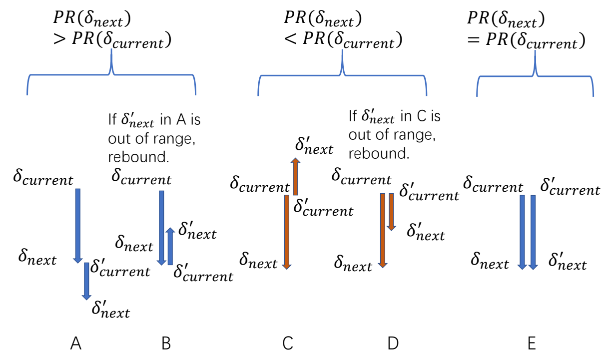

Fig. 2. The adaptive detection strategy of δ_1.

where $\mathbf{x}_i^{c1}$ denotes the i^{th} component of the sensory input from the ith sensor/information source, $\mathbf{w}_{i,j}^{c1}$ denotes the i^{th} component of the weight vector $\mathbf{w}_j^{c1}$ corresponding to the $\mathbf{x}_i^{c1}$, δ_i denotes the intensity attenuation parameter δ for the signals from the ith sensor and l denotes the number of sensor types/information sources.

We call this variation of FALCON as IA-FALCON. Except for the code activation process, everything else remains the same for fair comparison.

B. Adaptive Detection of δ and Node Pruning

Another contribution of our work is the use of Hill Climbing method to adaptively adjust the value of δ, and pruning nodes generated by "detrimental δ" during detection of the intensity attenuation controller δ at the same time.

The adaptive detection strategy in Fig. 2 is to handle the adaptive adjustment of two intensity attenuation parameters assuming that the agent has two types of sensor. And we assume that $\delta_2 = 1 - \delta_1$ because it is the ratio of δ_1 and δ_2 that affects the performance of the agent. So, δ_1 is the only independent variable to be adaptively adjusted in Alg. 2. We adjust δ_1 by comparing the performance of the agent with two different values of δ_1. Two values of δ_1 are denoted by $\delta_{current}$ and δ_{next} respectively.

The performance of $\delta_{current}$ and δ_{next} denoted by PR($\delta_{current}$) and PR(δ_{next}) is compared per time in a given regular interval during the learning process. When $\delta_1 = \delta_{current}$, the number of times the FALCON-based agent receives positive feedback is denoted by PF($\delta_{current}$), the number of times the FALCON-based agent receives negative feedback is denoted by NF($\delta_{current}$) and the positive feedback ratio PR($\delta_{current}$) is as follows:

$$PR(\delta_{current}) = \frac{PF(\delta_{current})}{PF(\delta_{current}) + NF(\delta_{current})}. \tag{8}$$

When $\delta_1 = \delta_{next}$, the number of times the FALCON-based agent receives positive feedback is denoted by PF(δ_{next}), the number of times the FALCON-based agent receives negative feedback is denoted by NF(δ_{next}) and the positive feedback ratio PR(δ_{next}) is as follows:

Algorithm 1 Adaptive detection of δ_1

Initialize IA-FALCON network.
While (The times of detection do not exceed the specified times **and** the value of $\delta_{current}$ stays the same no more than the specified times t_{stay})
  Run the regular intervals. In each step, the probability of $\delta_1 = \delta_{current}$ and $\delta_1 = \delta_{next}$ is 50% respectively.
  Count the times of positive feedback and negative feedback when $\delta_1 = \delta_{current}$ and $\delta_1 = \delta_{next}$ respectively.
  If $(PR(\delta_{next}) > PR(\delta_{current}))$
    $\delta'_{current} = \delta_{next}$
    $\delta'_{next} = \delta'_{current} + \frac{\delta_{next} - \delta_{current}}{2}$
    If (δ'_{next} exceeds the range of δ_1^{c1})
      $\delta'_{next} = \delta'_{current} + \frac{\delta_{current} - \delta_{next}}{2}$
    End If
  Else If $(PR(\delta_{next}) < PR(\delta_{current}))$
    $\delta'_{current} = \delta_{current}$
    $\delta'_{next} = \delta'_{current} + \frac{\delta_{current} - \delta_{next}}{2}$
    If (δ'_{next} exceeds the range of δ_1^{c1})
      $\delta'_{next} = \delta'_{current} + \frac{\delta_{next} - \delta_{current}}{2}$
    End If
  Else
    $\delta'_{current} = \delta_{current}$
    $\delta'_{next} = \delta_{next}$
  End If
  $\delta_{current} = \delta'_{current}$
  $\delta_{next} = \delta'_{next}$
End While

$$PR(\delta_{next}) = \frac{PF(\delta_{next})}{PF(\delta_{next}) + NF(\delta_{next})}. \tag{9}$$

$\delta_{current}$ and δ_{next} in the next time of detection are denoted by $\delta'_{current}$ and δ'_{next}.

The specific detection process is shown in the Alg. 1 and Fig. 2. Note, that the agent receives positive feedback as it approaches the target. We determine which value of δ_1 is superior by comparing the percentage of positive feedback that the agent receives when δ_1 takes two different values. We will choose the value with a higher percentage of positive feedback to be $\delta'_{current}$, as is shown in case A, B, C and D in Fig. 2. And then we will choose a point starting from $\delta'_{current}$ and away from another value of δ_1 with worse performance. If this point is not out of range, we will choose it as δ'_{next}, as is shown in case A and C in Fig. 2. Otherwise, we will choose a point starting from $\delta'_{current}$ in another direction as δ'_{next}, as is shown in case B and D in Fig. 2. Specifically, if the agent with each value of δ_1 receives the same percentage of positive feedback, we will test these two values of δ_1 again, as is shown in case E in Fig. 2. As the algorithm prefers random exploration at the beginning, we initialize two values of δ_1 far away from each other to maintain the performance diversity. Specifically, considering $\delta \in (0,1)$, we initialize $\delta_{current} = 0.99$ and $\delta_{next} = 0.01$. And the distance between $\delta_{current}$ and δ_{next} will be halved in the next time of detection.

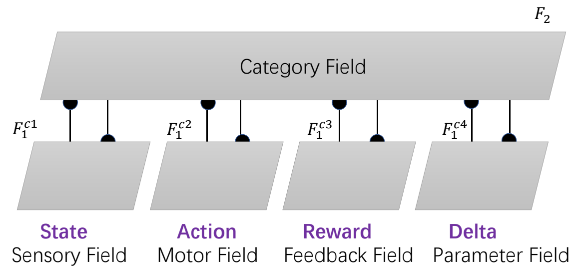

Fig. 3. The P-FALCON network architecture.

Algorithm 2 δ-based pruning algorithm

Initialize the P-FALCON network.
While ($trials \leq$ n)
  Run the regular intervals.
  For each node j in P-FALCON network
    If (rand $< (\frac{trials}{n})^2$ **and** $\frac{|\mathbf{x}^{c4} \wedge \mathbf{w}_j^{c4}|}{|\mathbf{x}^{c4}|} < \theta$)
      Delete node j
    End If
  End For
End While

Then, we introduce a δ-based pruning algorithm to remove the redundant nodes generated by the intensity attenuation parameter δ away from $\delta_{current}$. We do this based on the hypothesis that $\delta_{current}$ is getting closer to the optimal δ. Specifically, we modify the IA-FALCON network architecture as shown in Fig. 3. Each node holds the range of δ values that is used to select that node. This variation of IA-FALCON network architecture that stores parameter information in nodes is called P-FALCON. We focus on the differences between P-FALCON and IA-FALCON in the following.

P-FALCON uses a four-channel architecture consisting of a category field and four input fields. F_1^{c4} is the parameter input field. Modify the input vector of IA-FALCON to be $\mathbf{IV} = (\mathbf{S}, \mathbf{A}, \mathbf{R}, \mathbf{D})$ where $\mathbf{D} = (\delta_1, 1 - \delta_1)$ denotes the parameter vector, and δ_1 indicates the value of the intensity attenuation controller being used in this step. The parameter field works like any other input fields in code activation, code competition, template matching and template learning process.

We prune the P-FALCON network in regular intervals in the first n trials. After that, we delete all nodes that do not satisfy the requirement in the parameter field. The specific algorithm is shown in Alg. 2.

For example, the agent has sonar sensors in 5 directions, and a target bearing sensor. According to our hypothesis, the agent should be able to adjust the intensity attenuation controllers δ of different sensors adaptively to improve the performance of the agent. We can regard each cognitive code in the FALCON network as a rule for reaching a target while avoiding obstacles.

And we have observed in the experiments that some bad rules are generated, when the intensity attenuation controller δ is being adaptively adjusted and a better δ is not yet available. To remove these bad rules, we introduce a δ-based pruning algorithm.

The same sensory inputs might activate and generate two different rules due to different intensity attenuation controller δ. For example, the target is 1 unit directly in front of the person. And Rule 1 is to move 1 unit forward when the target is right ahead and the obstacle is more than 1 unit away from the person. In this rule, the information of the distance of the obstacle is vague, while the information of the direction of the target is clear. Rule 2 is to move 1 unit to the right when the obstacle right ahead, in the front-right, in the front-left, and in the left all is 1-3 units away from the person and the obstacle in the right is 3-4 units away from the person, no matter which direction the target is in. In this rule, the information of the distance of the obstacle is clear, while the information of the direction of the target is vague. It is clear that Rule 1 is reasonable and Rule 2 is misleading. If the agent does not even know the target bearing, it is impossible to get the right decision. This over-generalization of information from key information sources is caused by improper δ values. Cognitive codes that may interfere with the decisions of the agent need to be removed in time.

IV. Experiment

To verify the performance of the proposed method, we conducted experiments in a 16 by 16 Minefield Navigation Task (MNT) experimental platform. During the experiment, the FALCON-based autonomous vehicle (AV) repeats sense-move-learn steps until it reaches the target. We consider a total of 1000 trials. In each trial, the AV must reach the randomly initialized position of the target while avoiding 10 mines in a maximum of 30 steps, otherwise the task will fail. The AV owns the sonar sensor to detect the distance from the AV to the mines or the boundaries and the target bearing sensor to detect the bearing of the target. This work uses the FALCON network architecture with the immediate reward. We update the value of $\delta_{current}$ and δ_{next} in the adaptive algorithm and prune the nodes every 500 steps.

The parameters in the experiment for IA-FALCON are set as follows: the number of sensor types/information sources $l = 2$, the choice parameter $\alpha^{c1} = 0.1$, the learning rate parameter $\beta^{ck} = 1(k = 1, 2, 3)$ and the vigilance parameter $\rho^{c1} = 0.2, \rho^{c2} = 0.2, \rho^{c3} = 0.5$. The parameters of P-FALCON are set as follows: the number of sensor types/information sources $l = 2$, the choice parameter $\alpha^{c1} = 0.1$, the learning rate parameter $\beta^{ck} = 1(k = 1, 2, 3, 4)$, the vigilance parameter $\rho^{c1} = 0.2, \rho^{c2} = 0.2, \rho^{c3} = 0.5, \rho^{c4} = 0$, the number of trials for pruning $n = 500$, the threshold parameter $\theta = 0.8$ and *the intervals between twice node pruning* are 600 steps. The parameters of the Alg. 1 are set as follows: initialize two values of δ_1 in which $\delta_{current} = 0.99$ and $\delta_{next} = 0.01$, *the times of detection* of δ_1 are 5, the maximum times that the value of $\delta_{current}$ stays the same $t_{stay} = 3$ and *the intervals of once detection of* δ_1 are 600 steps. Notably, the parameters of basic FALCON dynamics are configured to be consistent with previous studies for fair comparison. Readers are referred to [19] for more details.

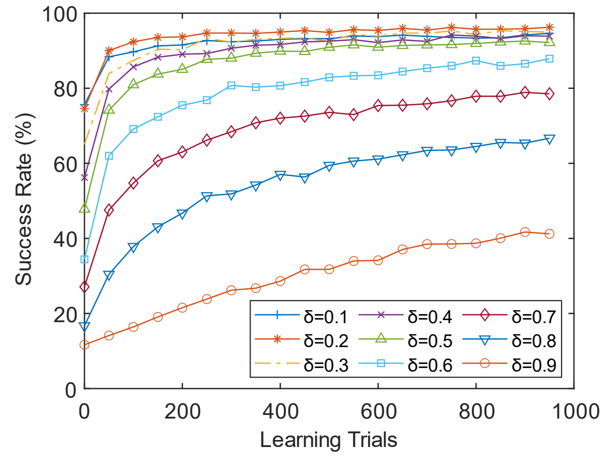

Fig. 4. Average success rate of the FALCON-based agent when differing δ values are involved.

We assume that $0 \le \delta_1 \le 1$, $0 \le \delta_2 \le 1$ for the complement code of the intensity attenuation controllers in P-FALCON. According to the Treisman Attenuation Model, sensory input from different sources is suppressed rather than eliminated. In other words, $\delta_1 \ne 0$ and $\delta_2 \ne 0$. Considering that it is the ratio of two intensity attenuation controllers that affects the performance of the algorithm, we set $\delta_2 = 1 - \delta_1$. And since $\delta_1 = 1 - \delta_2$, $\delta_1 \ne 1$. So, we assume that δ_1 goes from 0.01 to 0.99.

As is shown in Fig. 4, we compare the performance curves of the FALCON-based agent at different values of δ_1. Each curve is drawn with the data obtained by the FALCON-based agent performing 100 sets of independent experiments at the corresponding value of δ_1. When $\delta_1^{c1} \le 0.4$, the FALCON-based agent performs better than the FALCON-based agent without consideration of the intensity of different components in sensory input. In particular, when $\delta_1 = \delta_2 = 0.5$, it is equivalent to the FALCON-based agent without considering the intensity attenuation parameter δ.

As can be seen from Fig. 5, the lower the value of δ_1, the less nodes will be generated by the FALCON-based agent. The FALCON-based agent performs the best when the value of δ_1 locates between 0 and 0.3. Specifically, the success rate of the agent with $\delta_1 = 0.2$ is the highest (i.e., around 96% in Fig. 4). It exceeds the performance of agent with $\delta_1 = 0.1$ before the starting 50 learning trials. Moreover, the number of nodes when $\delta_1 = 0.1$ (i.e., 240) and $\delta_1 = 0.2$ (i.e., 260) is less than half of the number of nodes when $\delta_1 = 0.5$ (i.e., 560) after the learning process. This result indicates that the information from the target bearing sensor is more valuable than the information from the sonar sensor, hence have a higher contribution for instructing the learning process more effectively.

Obviously, the success rate of the adaptive algorithm is higher than that of the algorithm not considering the intensity attenuation controller δ when the number of nodes is approximately equal to the latter algorithm (see Fig. 6 and Fig. 7). Although the adaptive algorithm generates more nodes at the beginning of detection with a “detrimental δ”, its code numbers

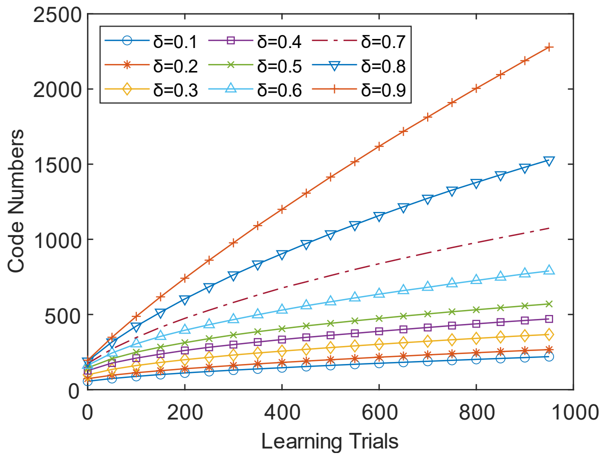

Fig. 5. Average code numbers of the FALCON-based agent when the value of δ differs.

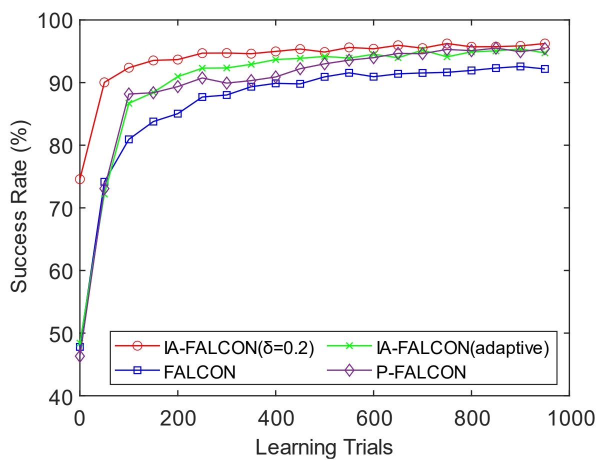

Fig. 6. Average success rate of the adaptive detection algorithm with and without δ-based pruning.

get closer to that of the algorithm without considering the intensity attenuation controlling (Fig. 7).

After applying the δ-based adaptive detection and pruning algorithms simultaneously, the number of nodes is significantly reduced than the adaptive algorithm while the success rate is still competitive by reporting around 95% after the learning process (see Fig. 6 and Fig. 7). Note, in our case the pruning algorithm stops after around 500 learning trials when the adaptive detection of δ_1 finishes, which means the algorithm will not generate and prune redundant nodes with a "detrimental δ". The success rate of the adaptive algorithm with pruning is almost equal to the success rate of the algorithm without pruning after the 500 trials. This result thus highlights the efficacy of the proposed node pruning method for reducing the number of redundant memorial nodes while remaining competitive in performance by achieving a high success rate.

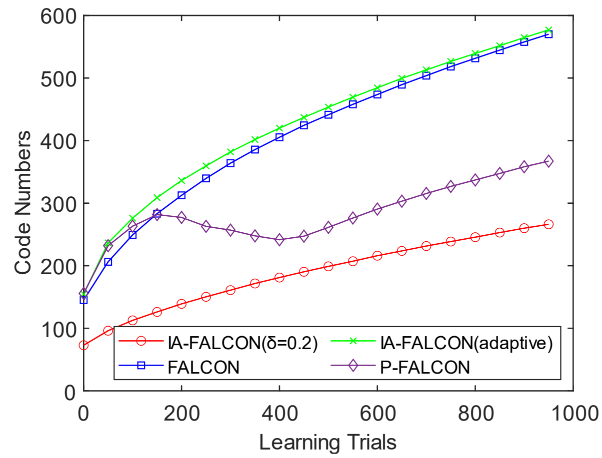

Fig. 7. Average code numbers of the adaptive detection algorithm with and without δ-based pruning.

Further, the agent using the δ-based adaptive detection and pruning algorithms simultaneously reported a success rate of around 95% at a small size of 300 nodes, which is significantly better than the agent not considering the intensity attenuation controller δ which obtained a success rate of around 92% at 560 nodes. Notably, the success rates of the adaptive algorithm are superior to that of the algorithm without consideration of δ after 50 trials when the first time δ detection finishes (refers to Fig. 6). The code numbers of the adaptive algorithm with pruning are less than the algorithm without consideration of δ after 150 trials (see Fig. 7). As a result, the effectiveness of δ-based adaptive detection can be verified as well.

V. Conclusion

In this paper, we introduce an intensity attenuation controller δ for fusion ART models by taking inspiration from the Broadbent-Treisman Filter-Attenuation Model. Specifically, taking a Fusion Architecture for Learning, Cognition, and Navigation (FALCON) as the specific and essential backbone of fusion ARTs, we propose an IA-FALCON with a newly proposed code activation process for adaptive information intensity detection as well as a P-FALCON for pruning the redundant nodes generated by the "detrimental" attenuation controller. Comprehensive experimental studies are conducted on a classical minefield navigation problem. The results demonstrate that our proposed method obtains significantly better learning performance in terms of coverage speed while successfully reduces redundant memory storage incurred by the "detrimental δ" , thus verifies its effectiveness and efficiency.

The immediate future work may consider to investigate the adaptive detection of the intensity attenuation controller with various different types of information sources/sensors. And we would like to consider the adaptive algorithm in more complicated experimental circumstances, for example the multi-agent problems, hence improve its generality for real-world problem solving.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0910500, in part by the National Natural Science Foundation of China under Grant 61906032 and Grant 61876025, in part by the Liaoning Key Research and Development Program under Grant 2019JH2/10100030, in part by the Liaoning United Foundation under Grant U1908214 and in part by the Venture & Innovation Support Program for Chongqing Overseas Returnees under Grant No. cx2018044 and cx2019020.

REFERENCES

[1] G. A. Carpenter and S. Grossberg, "Adaptive resonance theory," in *The Handbook of Brain Theory and Neural Networks*, Michael A. Arbib, Ed. Cambridge, MA: MIT Press, 2003, pp. 87–90.

[2] G. A. Carpenter and S. Grossberg, *Pattern Recognition by Self-Organizing Neural Networks*, 1st ed. Cambridge, MA, USA: MIT Press, 1991.

[3] A.-H. Tan, B. Subagdja, D. Wang, and L. Meng, "Self-organizing neural networks for universal learning and multimodal memory encoding," *Neural Netw.*, vol. 120, pp. 58–73, Dec. 2019, doi: 10.1016/j.neunet.2019.08.020.

[4] A.-H. Tan, "FALCON: A fusion architecture for learning, cognition, and navigation," in *Proceedings of the International Joint Conference on Neural Networks*, 2004, vol. 4, pp. 3297–3302.

[5] L. Meng, A.-H. Tan, and D. C. Wunsch II, "Adaptive resonance theory (ART) for social media analytics," in *Adaptive Resonance Theory in Social Media Data Clustering: Roles, Methodologies, and Applications*, L. Meng, A.-H. Tan, and D. C. Wunsch II, Eds. Cham: Springer International Publishing, 2019, pp. 45–89.

[6] L. Meng, A.-H. Tan, C. Leung, L. Nie, T.-S. Chua, and C. Miao, "Online multimodal co-indexing and retrieval of weakly labeled web image collections," presented at the Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015, pp. 219–226.

[7] J. He, A.-H. Tan, and C.-L. Tan, "On machine learning methods for Chinese document categorization," *Appl. Intell.*, vol. 18, no. 3, pp. 311–322, 2003, doi: 10.1023/A:1023202221875.

[8] Yaqing Hou, L. Feng, and Y. Ong, "Creating human-like non-player game characters using a Memetic Multi-Agent System," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 177–184, doi: 10.1109/IJCNN.2016.7727196.

[9] S. Feng, "Computational techniques for modeling non-player characters in games," Nanyang Technological University, 2014.

[10] A.-H. Tan and H.-S. V. Soon, "Predictive adaptive resonance theory and knowledge discovery in databases," presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000, pp. 173–176.

[11] Y. Hou, Y. Ong, J. Tang, and Y. Zeng, "Evolutionary multiagent transfer learning with model-based opponent behavior prediction," *IEEE Trans. Syst. Man Cybern. Syst.*, pp. 1–15, 2019, doi: 10.1109/TSMC.2019.2958846.

[12] G. Zhang, L. Feng, Y. Xie, Z. Wu, and L. Chen, "Improving reinforcement FALCON learning in complex environment with much delayed evaluation via memetic automaton," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand, Jun. 2019, pp. 166–173, doi: 10.1109/CEC.2019.8789988.

[13] Y. Hou, Y.-S. Ong, L. Feng, and J. M. Zurada, "An evolutionary transfer reinforcement learning framework for multiagent systems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 601–615, Aug. 2017, doi: 10.1109/TEVC.2017.2664665.

[14] Y. Zeng, X. Chen, Y.-S. Ong, J. Tang, and Y. Xiang, "Structured memetic automation for online human-like social behavior learning," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 102–115, 2016.

[15] Y. Kang and A.-H. Tan, "Self-organizing agents for reinforcement learning in virtual worlds," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, Barcelona, Spain, Jul. 2010, pp. 1–8, doi: 10.1109/IJCNN.2010.5596363.

[16] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, Second edition. Cambridge, MA London: The MIT Press, 2018.

[17] D. E. Broadbent, *Perception and Communication*. Pergamon, 1958.

[18] Anne. M. Treisman, "Selective attention in man," *Br. Med. Bull.*, vol. 20, no. 1, pp. 12–16, 1964, doi: 10.1093/oxfordjournals.bmb.a070274.

[19] A.-H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Trans. Neural Netw.*, vol. 19, no. 2, pp. 230–244, 2008.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE