Leader-Following Formation Control with an Adaptive Linear and Terminal Sliding Mode Combined Controller Using Auto-Structuring Fuzzy Neural Network

Masanao Obayashi, Kohei Ishikawa,
Takashi Kuremoto, Shingo Mabu
Graduate School of Science and Engineering,
Yamaguchi University,Ube, Yamaguchi, Japan
email:{m.obayas, s003vk, wu, mabu}@yamaguchi-u.ac.jp

Kunikazu Kobayashi
School of Information Science and Technology,
Aichi Prefectural University, Nagakute, Aichi, Japan
email: kobayashi@ist.aichi-pu.ac.jp

***Abstract*—This paper proposes an intelligent formation control method of the leader and follower agents with nonlinear dynamics. In the proposed method, agents can exchange only information of positions of available agents and the follower agents follow the leader, taking a predefined formation. In the real world, the method that doesn't need a lot of information to make cooperative behaviors is very useful for the case of environments existing communication delay and weak communication. In addition, each agent will be controlled by the linear and terminal combined sliding mode control method. Furthermore, to adapt the change of the environment, the Auto-Structuring Fuzzy Neural Control System (ASFNCS) is introduced to provide appropriate control inputs while coping well with disturbance and nonlinear dynamics. In the simulation, it is verified that the proposed method is useful in the point of performance of the leader-following formation control.**

Keywords-formation control; linear sliding mode control; terminal sliding mode control; auto-structuring fuzzy neural network

I. INTRODUCTION

Recently, formation problems have attracted more researchers as their importance and necessity are known widely. For example, Hou et al. [1] proposed a robust adaptive control approach to solve a consensus problem of Multi Agent Systems (MAS), Song et al. [2] and Cheng et al. [3] dealt with leader-follower consensus problem. Defoort et al. [4], Morbidi et al. [5] proposed a formation following control using sliding mode control, a kind of representative robust control methods. Cui et al, [6] dealt with leader-follower formation control of underactuated autonomous underwater vehicles. Yu et al. [7] dealt with time-varying velocity cases of a distributed leader-follower flocking control for multi-agent dynamical systems. In the view point of controllers using soft computing tools, Chang et al. [8] used fuzzy, Lin [9] used fuzzy basis function network, Chen et al. [10] used neural networks, Lin et al. [11] used reinforcement Q learning method, for formation control. However, designed controller structures for these formation or consensus problems are fixed, therefore, their methods are no flexible for changes of environments and they have possibility of useless computation load. Many of the control methods mentioned above are using linear sliding mode control method. Recently, terminal sliding mode control method has been used for constructing the following controller because of its superiority on fast finite time convergence and less steady state errors to objective states of the agents, Yu et al. [12], Zou et al. [13], Chang et al. [14].

We have already proposed the leader-following formation control method, Obayashi et al. [15] that had the following three features; use of linear sliding mode control, and the controller had the optimal structure, named ASFNCS, using self-structuring algorithm adapting to change of environments making use of the concept of Cheng et al. [16] and then introducing of making only use of the positions of agents without velocities of them.

In this paper, we consider to apply the superior point of the terminal sliding mode control to our previous work, Obayashi et al. [15], that is, to switch the linear sliding mode control and terminal one appropriately.

The rest of the paper is as follows: in Section II preparation to be required to follow the paper. The structure of the proposed system are described in Section III. The controller design is described in Section IV. In Sections V, VI, and VII, learning algorithm, following control problem and computer simulation are described, respectively. We have a conclusion in Section VIII.

II. PREPARATION

The dynamics of each agent consisting of MAS is described as following the $n\,th$ order nonlinear differential equation,

$$\begin{aligned} x^{(n)} &= f(\mathbf{x}, u_{ASFNS}) + \Delta f(\mathbf{x}, u_{ASFNS}) + d \\ &= f(\mathbf{x}, u_{ASFNS}) - h \cdot u_{ASFNS} + h \cdot u_{ASFNS} + \delta_1 \ , \\ &= f_n(\mathbf{x}, u_{ASFNS}) + h \cdot u_{ASFNS} + \delta_1 \end{aligned} \qquad (1)$$

where $\mathbf{x} = [x \quad \dot{x} \quad \cdots \quad x^{(n-1)}]^T \in R^n$: observable states of the agent, $f(\mathbf{x})$: unknown continuous function, h : a predefined constant, u_{ASFNS} : control input, δ_1 : term including the indefinite term $\Delta f(\mathbf{x}, u_{ASFNS})$ of the agent and the disturbance. δ_1 is assumed that $|\delta_1| < D_1$. Here, D_1 is a predefined positive constant.

III. STRUCTURE OF THE PROPOSED SYSTEM

The proposed system in this paper is shown in Figure 1.

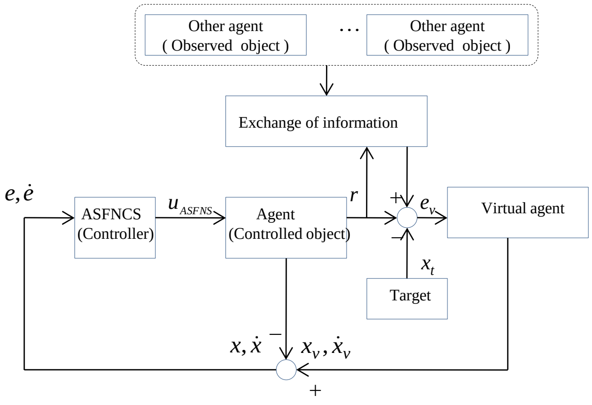


Figure 1. Structure of the proposed multi agent control system.

The objective of the control: The derivation of the control law which makes the orbit $\mathbf{x}$ of the states of the agent follow the reference orbit $\mathbf{x}_V$ accurately. We define the following error vector $\mathbf{e} = [e, \dot{e}, \cdots, e^{(n-1)}]$ as

$$\mathbf{e} = \mathbf{x}_V - \mathbf{x}\,, \qquad (2)$$

where $\mathbf{x}_V$: the reference vector, that is, their virtual agent state vector followed by its own agent. The virtual agent has its position and velocity decided by observing the position of the leader (see Section VI).

IV. CONTROLLER DESIGN

Figure 2 shows the ASFNCS. The ASFNCS consists of the fuzzy neural network controller with function of node adding/pruning (Auto-structuring FNNC) and the robust controller.

A Fuzzy neural network controller (FNNC)

Figure 3 shows the structure of the FNNC. Γ_k , output of k th node of the hidden layer, means the fitness of rule k .

$$\Gamma_k(\mathbf{e}) = \prod_{i=1}^{n} \exp\left\{ -\frac{\left(e^{(i-1)} - m_{ki}\right)^2}{\left(\sigma_{ki}\right)^2} \right\} \;, \qquad (3)$$

$$(k = 1, \cdots, R) \;,$$

$$u_{asfnn} = \sum_{k=1}^{R} \xi_k \cdot \Gamma_k \,, \qquad (4)$$

where $\mathbf{m}_k = [m_{ki}, m_{k2}, \cdots, m_{kn}]$, $\boldsymbol{\sigma}_k = [\sigma_{ki}, \sigma_{k2}, \cdots, \sigma_{kn}]$ are center and width vectors of the kth rule output function, respectively. R is the number of the rule nodes. u_{asfnn} is the output of the FNNC. ξ_k is the weight between the $k\,th$ rule node and the output of the FNNC.

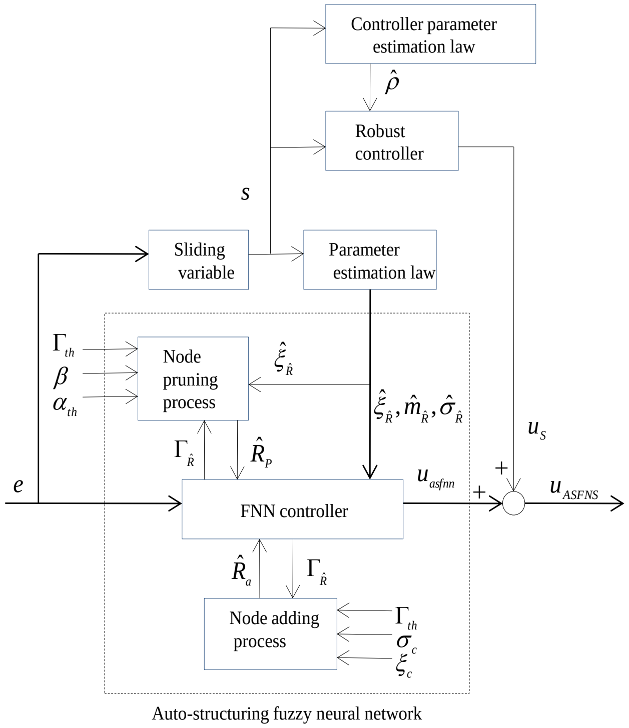


Figure 2. Structure of theASFNCS.

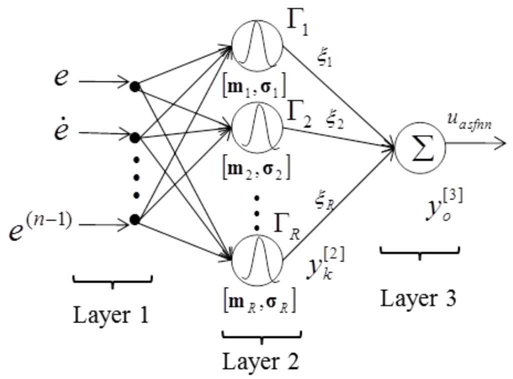


Figure 3. Structure of the FNNC.

The output u^* of ideal controller is like that,

$$u^* = u^*_{asfnn} + \varphi_1 = \sum_{k=1}^{R^*} \xi^*_k \Gamma_k + \varphi_1, \qquad (5)$$

φ_1: difference between ideal controller and aproximated optimal FNNC.

However, u^* would not be used for controller design actually (see Sectiuon IV *B*).

The estimated value $\hat{u}_{asfnn}$ of the optimal FNNC u^*_{asfnn} is as follows,

$$\hat{u}_{asfnn} = \sum_{k=1}^{\hat{R}} \hat{\xi}_k \cdot \Gamma_k \,, \qquad (6)$$

where $\hat{\xi}_k$ is the estimate value of the optimal ξ_k^*.

B. Auto structuring mechanism of FNNC

Node adding: Nodes adding process is as follows,

$$\Gamma_{\max} = \max(\Gamma_k), \quad k = 1,2,\cdots,R(t), \tag{7}$$

where $R(t)$ is the number of the node at time t. When the next relation exits,

$$\Gamma_{\max} \le \Gamma_{th}, \tag{8}$$

where $\Gamma_{th} \in [0, 1]$ is a pre-defined threshold and given by trial and error, a new node σ_c is added and the center and width of the node are like this,

$$m_{R_a i} = e^{(i-1)}, \sigma_{R_a i} = \sigma_c, \; (i = 1,\cdots,n), \tag{9}$$

where σ_c is a predefined positive constant and the weight ξ_{R_a} between the new node and the output is set as

$$\xi_{R_a} = \xi_c, \tag{10}$$

where R_a is number of the added node.

Node pruning: Nodes pruning process is as follows, the cost function for measuring the important index α_k of the node k is defined as,

$$E_1 = \frac{1}{2}(u^* - \sum_{k=1}^{\hat{R}} \hat{\xi}_k \Gamma_k)^2, \tag{11}$$

using the Taylor series expansion, where the parameter falls intou the local minimum, the sensitivity of E_1 for $\dot{\hat{\xi}}_k$ becomes next equation,

$$E_1(\dot{\hat{\xi}}_k) \approx \frac{1}{2}(\dot{\hat{\xi}}_k \Gamma_k)^2, \quad k = 1,2,\ldots,\hat{R}. \tag{12}$$

Here, we define the important index of the node as

$$\alpha_k(t+1) = \alpha_k(t) \times \exp\left\{-\beta \times \delta[E_{th} - E_1(\dot{\hat{\xi}}_k)]\right\}, \tag{13}$$

and $\delta[*]$ function to decide whether make α_k decrease is defined as

$$\delta[E_{th} - E_1(\dot{\hat{\xi}}_k)] = \begin{cases} 1 & \text{if } E_{th} - E_1(\dot{\hat{\xi}}_k) \ge 0 \\ 0 & \text{if } E_{th} - E_1(\dot{\hat{\xi}}_k) < 0 \end{cases}. \tag{14}$$

This means that if the rule k is not activated enough, that is, $E_{th} - E_1(\dot{\hat{\xi}}_k) \ge 0$, then, α_{k} decreases. If $\alpha_{\mathrm{k}} \le \alpha_{\mathrm{th}}$, the node k is considered as unnecessary node and it is deleted. Here α_{th} is a predefined positive constant and given by trial and error. The number of deleted nodes is presented as $\hat{R}_p$.The output u_{asfnn} of the FNNC introducing the auto-structuring mechanism is as follows,

$$u_{asfnn} = \underbrace{\sum_{k=1}^{\hat{R}} \hat{\xi}_k \Gamma_k}_{u_R} - \underbrace{\sum_{p=1}^{\hat{R}_p} \hat{\xi}_p \Gamma_p}_{u_p}, \tag{15}$$

where u_R is the output after adding new nodes and, u_p is the output after pruning nodes. With (15), we get

$$\tilde{u}_{asfnn} = u^* - \hat{u}_{asfnn} = \sum_{k=1}^{\hat{R}} \tilde{\xi}_k \Gamma_k + \varepsilon, \tag{16}$$

where $\tilde{\xi} = \xi^* - \hat{\xi}$.

C. Stability analysis

The structure of the auto-structuring fuzzy neural control system (ASFNCS) is shown in Figure 2. The output u_{ASFNS} of the ASFNCS is as follows, Cheng et al. [16],

$$u_{ASFNS} = u_{asfnn} + u_S, \tag{17}$$

u_{asfnn} is described by (15), and u_s is the output of the robust controller. To derive an adaptive controller, we define the sliding variable s as

$$\begin{aligned} s = {} & e^{(n-1)} + k_1 e^{(n-2)} + \cdots + k_{n-1} e \\ & + \gamma\, k_{n1} e^{p/q} + (1-\gamma) k_{n2} \int_0^t e dt \end{aligned}, \tag{18}$$

where each of all the cofficients k_* is pre-defined positive constant, $\gamma (0 \le \gamma \le 1)$ is a parameter. When $\gamma = 0.0, \; 1.0$, sliding variable (18) can be regarded as LSM, TSM, respectively. In this paper, γ changes nonlinearly from 1.0 to 0.0 during the controller working.
The p and q are positive odd integers, which satisfy the following condition:

$$p \;>\; q. \tag{19}$$

Then, a sliding mode controller can be designed as follows :

$$u_{SMC} = u_E + u_H, \tag{20}$$

An equivalent controller u_E is expressed as

$$\begin{aligned} u_E = h^{-1}(& -f_{n(x,u)} + x_v^{(n)} + k_1 e^{(n-1)} + \cdots + k_{n-1}\dot{e} \\ & + \gamma\, k_{n1} \frac{q}{p} e^{q/p} \cdot \dot{e} + (1-\gamma) k_{n2} \cdot e) \end{aligned} \tag{21}$$

$$u_H = D_1 \operatorname{sgn}(s), \tag{22}$$

Using (1), (2),(18) and (21), we get

$$\begin{aligned} \dot{s} = {} & e^{(n)} + + k_1 e^{(n-1)} + \cdots + k_{n-1}\dot{e} \\ & + \gamma\, k_{n1} \frac{q}{p} e^{q/p} \cdot \dot{e} + (1-\gamma) k_{n2} \cdot e \\ = {} & h(u^* - u_{ASFNS}). \end{aligned} \tag{23}$$

Using Eq. (16), Eq. (23) can be rewritten as

$$\dot{s} = h(u^* - u_{asfnn} - u_s) = h(\sum_{k=1}^{\hat{R}} \tilde{\xi}_k \Gamma_k + \varepsilon - u_s). \qquad (24)$$

In this paper, the robust controller is used to eliminate the effect of error ε. Consider the Lyapunov function candidate in the following form:

$$V = \frac{1}{2}s^2 + \frac{h}{2\eta_\xi}\sum_{k=1}^{\hat{R}}\tilde{\xi}_k^2 + \frac{h}{2\eta_\rho}\tilde{\rho}^2 . \qquad (25)$$

Take the derivative of Eq.(25) and using Eq.(24), it is concluded that

$$\begin{aligned} \dot{V} &= s\dot{s} + \frac{h}{\eta_\xi}\sum_{k=1}^{\hat{R}}\tilde{\xi}_k\dot{\tilde{\xi}}_k + \frac{h}{\eta_\rho}\tilde{\rho}\dot{\tilde{\rho}} \\ &= h\left\{\sum_{k=1}^{\hat{R}}\tilde{\xi}_k\left(s\Gamma_k + \frac{\dot{\tilde{\xi}}}{\eta_\xi}\right) + s(\varepsilon - u_s) + \frac{1}{\eta_\rho}\tilde{\rho}\dot{\tilde{\rho}}\right\} . \end{aligned} \qquad (26)$$

For achieving $\dot{V} \le 0$, the adaptation law and the robust controller are chosen as

$$\dot{\tilde{\xi}} = -\dot{\hat{\xi}}_k = -\eta_\xi s\Gamma_k , \quad k = 1, 2, \ldots, \hat{R}, \qquad (27)$$

$$u_s = \hat{\rho}\,\mathrm{sgn}(s), \qquad (28)$$

$$\dot{\tilde{\rho}} = -\dot{\hat{\rho}} = -\eta_\rho|\rho|, \qquad (29)$$

where η_ρ and η_ξ are positive constants. Then Eq. (26) can be rewritten as

$$\begin{aligned} \dot{V} &= h\{s\varepsilon - \hat{\rho}|s| - (\rho - \hat{\rho})|s|\} \le h\{|\varepsilon||s| - \rho|s|\} \\ &= -h(\rho - |\varepsilon|)|s| \le 0 \end{aligned} . \qquad (30)$$

If $|\varepsilon| \le \rho$, $\dot{V} \le 0$ holds. By Barbalat's Lemma, $s \to 0$ as $t \to \infty$, then the stability is guaranteed.

V. LEARNING ALGORITHM

We introduce the online learning algorithm to adjust the parameters of ASFNS, Cheng et al. [16]. We derive the algorithm the gradient-decent method. First, the adaptive law shown in (27) can be written as

$$\dot{\hat{\xi}}_k = \eta_\xi s\Gamma_k . \qquad (31)$$

According to the gradient-decent algorithm, the adaptive law of $\hat{\xi}$ also can be represented as

$$\dot{\hat{\xi}} = \eta_\xi \frac{\partial E_2}{\partial\hat{\xi}_k} = \eta_\xi \frac{\partial E_2}{\partial y_o^{[3]}}\frac{\partial y_o^{[3]}}{\partial net_o^{[3]}}\frac{\partial net_o^{[3]}}{\partial\hat{\xi}_k} = \eta_\xi \frac{\partial E_2}{\partial y_o^{[3]}} y_k^{[2]} , \qquad (32)$$

where $E_2 = 1/2(x_v - x)^2$ is defined as the cost function. From Figure 3, (3), and (4), $net_k^{[2]} = \sum_{i=1}^{n}\left\{-(e^{(i-1)} - \hat{m}_{ki})^2/(\hat{\sigma}_{ki})^2\right\}$, $y_k^{[2]} = \exp(net_k^{[2]})$, $y_o^{[3]} = \sum_k \hat{\xi}_k\, y_k^{[2]} = \sum_k \hat{\xi}_k\, \Gamma_k$, $*$ in the [*] means the number of layer in Figure 3. Thus the Jacobian term $\partial E_2 / \partial y_o^{[3]} = s$ is obtained through observation of (31) and (32). Thus, the adaptive law for the estimation terms of means $\hat{m}_{ki}$, and variance $\hat{\sigma}_{ki}$ can be derived as

$$\begin{aligned} \dot{\hat{m}}_{ki} &= \eta_m \frac{\partial E_2}{\partial\hat{m}_{ki}} = \eta_\xi \frac{\partial E_2}{\partial y_o^{[3]}}\frac{\partial y_o^{[3]}}{\partial y_k^{[2]}}\frac{\partial y_k^{[2]}}{\partial net_k^{[2]}}\frac{\partial net_k^{[2]}}{\partial\hat{\sigma}_{ki}} \\ &= \eta_m s\hat{\xi}_k\Gamma_k \frac{2(e^{(i-1)} - \hat{m}_{ki})^2}{(\hat{\sigma}_{ki})^2}, \end{aligned} \qquad (33)$$

$$\begin{aligned} \dot{\hat{\sigma}}_{ki} &= \eta_\sigma \frac{\partial E_2}{\partial\hat{m}_{ki}} = \eta_\xi \frac{\partial E_2}{\partial y_o^{[3]}}\frac{\partial y_o^{[3]}}{\partial y_k^{[2]}}\frac{\partial y_k^{[2]}}{\partial net_k^{[2]}}\frac{\partial net_k^{[2]}}{\partial\hat{\sigma}_{ki}} \\ &= \eta_m s\hat{\xi}_k\Gamma_k \frac{2(e^{(i-1)} - \hat{m}_{ki})^2}{(\sigma_{ki})^2}, \end{aligned} \qquad (34)$$

where η_m , η_σ are positive constants.

A. The algorithm of ASFNCS

Step 1 Initialize the parameters of ASFNCS.
Step 2 Calculate s using (18).
Step 3 Calculate u_{asfnn} using (15) and update
$\hat{\xi}, \hat{m}, \hat{\sigma}$ using (32)-(34).
Step 4 Calculate u_s using (28) and adjust $\hat{\rho}$
using (29).
Step 5 Calculate u_{ASFNS} using (17) and input
it to the agent.
Step 6 If the controlling time is over, the simulation
ends, else go to Step 2.

VI. LEADER-FOLLOWING FORMATION CONTROL PROBLEM

The dynamics (1) of the $i\,th$ agent in multi agents is rewritten as

$$\ddot{x}_i(t) = f(t, x_i(t), \dot{x}_i(t)) + d_i + u_{i,ASFNC}, \qquad (35)$$

where t : time, $x_i = (x_{1i}, x_{2i})^T$, $\dot{x}_i = (\dot{x}_{1i}, \dot{x}_{2i})^T$: the position, velocity of the agent, respectively. $f(t, x_i(t), \dot{x}_i(t)) \in R^2$: a term including the nonlinearity $u_{ASFNS} = (u_{1i}, u_{2i})^{\mathrm{T}} \in R^2$, $u_{i,ASFNC} = (u_{1i}, u_{2i})^T \in R^2$: control input to the $i\,th$ agent, $d_i \in R^2$: disturbance. In this paper, we consider the following control method that make the agents follow and catch the leader keeping the formation by the group, Cui et al. [6]. We assume that agents could exchange only positions information of the agents each other. Observing only the position of the leader, each agent constructs the virtual orbit to the leader. Realizing this virtual orbit by the virtual agent, and the agent follows its own agent. The following describes the method of following the leader by the virtual agent.

The following error of the $i\,th$ virtual agent is expressed as

$$e_{vi} = ((z_i + \omega_i) - x_t) + \sum_{j \in N_i} ((z_i + \omega_i) - z_j)\,, \qquad (36)$$

where e_{vi} :consensus error, $z_i = x_i$, $\omega_i = (x_{vi} - x_i)$, $x_t \in R^2$: the position of the leader in a 2-dimension space.

The following error x_{ei} of the $i\,th$ virtual agent considering the integral term to reduce the steady state error is expressed as

$$x_{ei} = e_{vi} + \phi_i + \int_0^t e_{vi} dt \ , \qquad (37)$$

where $\phi_i = [\phi_{1i}, \phi_{2i}]^T$, the derivative of ϕ_i with respect to time is defined as

$$\dot{\phi}_i = -\beta_{1i}(\phi_i) - Kx_{ei} \ , \qquad (38)$$

where $\beta_{1i}(\phi_i) = [\lambda_1 \tanh(\phi_i/\lambda_1), \lambda_2 \tanh(\phi_i/\lambda_2)]^T$, $K = diag\ [\ k_{v1}, k_{v2}]$, $\phi_i(0) = 0$.

The velocity $\dot{x}_{vi}$ of the $i\,th$ virtual agent is defined as

$$\dot{x}_{vi} = \beta_{1i}(\phi_i) + \beta_{2i}(\phi_i) \ , \qquad (39)$$

where $\beta_{2i}(\phi_i) = [k_{v1} \tanh(\phi_i/k_{v1}), k_{v2} \tanh(\phi_i/k_{v2})]^T$, k_*, λ_* are positive constants.

VII. COMPUTER SIMULATION

We demonstrate the effectiveness of our proposed method by trying to following the leader (target) by the multi agent constructing by four agents.

The horizontal and vertical direction of velocity of the leader are $\dot{x}_{t1} = 0.02\cos(0.05t)$, $\dot{x}_{t2} = 0.015\sin(0.03t)$, respectively, and the initial position of the leader is given as $x_t = [0{,}0]^T$. The initial horizontal and vertical positions of the agents are both given randomly in the range of $[-5{,}5]$, and the initial velocities of all the four agents are equal to 0. The parameters used in the simulation are shown in Tables I and II. The dynamics and disturbances of the each agent are given as

$$f(t, x_i(t), \dot{x}_i(t)) = \begin{pmatrix} \sin(x_{1i}) + \dot{x}_{1i}\cos(\dot{x}_{2i}) \\ \cos(x_{2i}) + \dot{x}_{2i}\sin(\dot{x}_{1i}) \end{pmatrix} \ , \qquad (40)$$

$$d_1 = \begin{pmatrix} -\sin(2t) \\ \sin(t^2) \end{pmatrix}, d_2 = \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}, d_3 = \begin{pmatrix} \cos(2t) \\ -\cos(t) \end{pmatrix}, d_4 = \begin{pmatrix} -\sin(t) \\ -\cos(2t) \end{pmatrix} . \ (41)$$

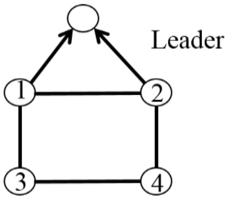


Figure 4. Network structure of multi agents.

The information exchanges among agents each other are carried out according to Figure 4. Each agent behaves dispersive and cooperatively through information exchanges and follows the leader, taking the predefined formation. In this simulation, from Figure 4, only agent 1 and 2 can observe the position of the leader and the agent 4 can observe the positions of the agent 1 and 3, and so on. Sampling time is $0.01[s]$ and total controlling time is 50 [sec].

Each of Figures 5-11 has 5 orbits; red is for target (leader), other 4 orbits are for follower agebts. The orbits of the target and follower agents in the case of the conventional method, LSM, are shown in Figure 5. Figure 6 is the enlarged figure of the transient (initial part of) positons of the leader and follower agents. The orbits of the leader and agents in the case of the conventional method, LSM, are shown in Figure 7. Figure 8 is the enlarged figure of the transient (initial part of) positons of the agents and the leader. Comparing these four figures, TSM method is superior to the LSM method in the point of consensus error of the positions of transient states of the leader and follower agents. However, the consensus error of them in the point of the steady state by LSM method is smaller than those of TSM method.

TABLE I. PARAMETERS OF THE VIRTUAL AGENT.

Parameter	Setting value
λ_1, λ_2	0.015
k_{v1}, k_{v2}	0.025

TABLE II. PARAMETERS OF THE ASFNCS.

Parameter	Setting value
k_1	2
k_2	3
Γ_{th}	0.3
σ_c	2
ξ_c	0
α_c	0.5
E_{th}	0.00001
β	0.002
α_{th}	0.05
η_ξ	1
η_ρ	0.02
η_m, η_σ	0.0015

ISBN: 978-1-61208-390-2

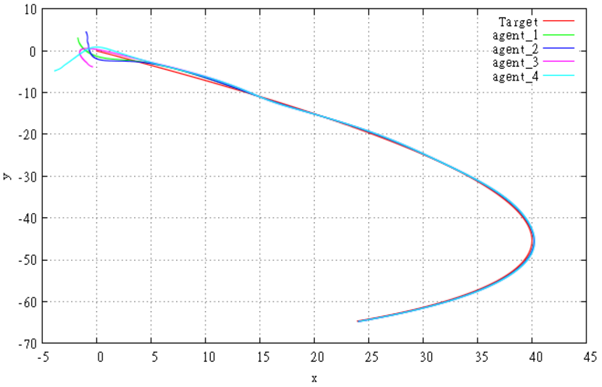


Figure 5. Trajectories of all the agents using LSM ($\gamma = 0$ in (18)) .

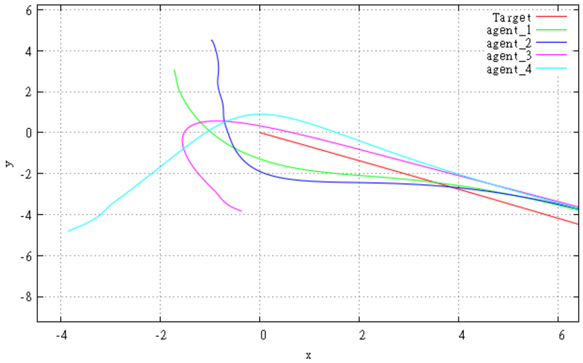


Figure 6. Transient trajectories of all the agents using LSM ($\gamma = 0$).

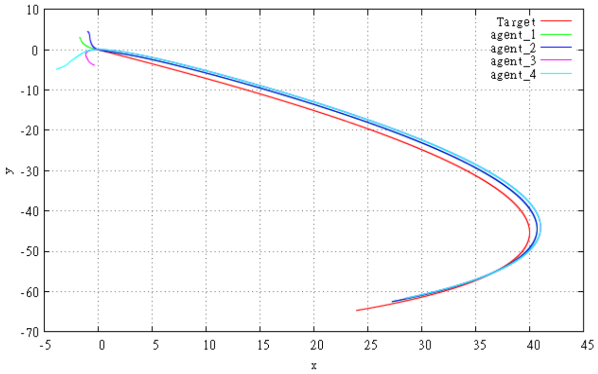
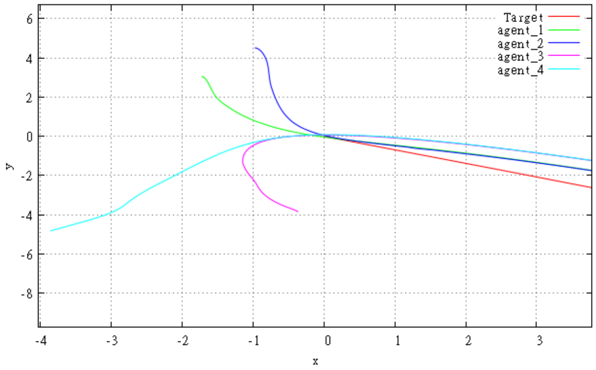


Figure 7. Trajectories of all the agents using TSM($\gamma = 1$ in (18)) .



\ Figure 8. Transient trajectories of all the agents using TSM ($\gamma = 1$).

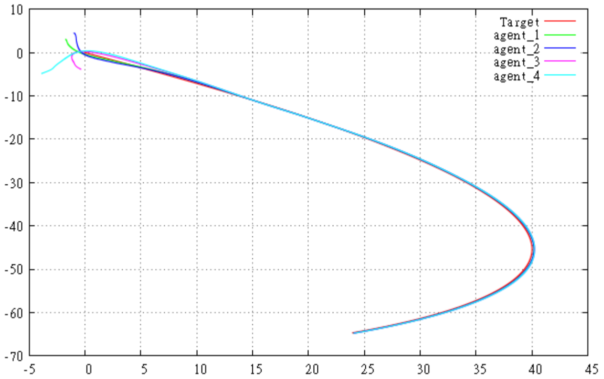


Figure 9. Trajectories of all the agents using TLSM.

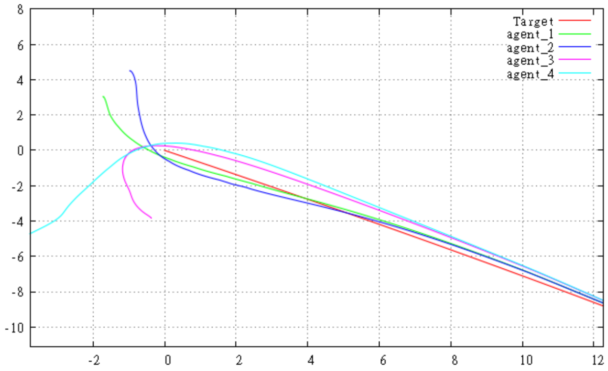


Figure 10. Transient trajectories of all the agents using T LSM.

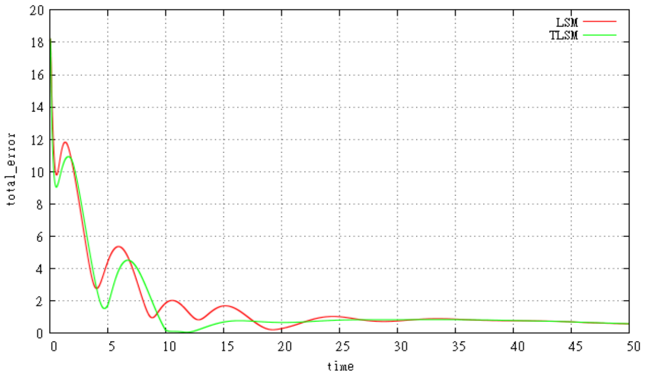


Figure 11. Trends in distances error between a leader and all follower agents.

Figures 9-10 show the results using our proposed LSM and TSM combined control, changing γ in (18) from 1.0 to 0.0 nonlinearly. Figure 11 shows trends in the consensus error between a leader and all follower agents for LSM and TLSM. From Figure 11, it can be found that consensus error by our proposed method is smaller than that of the conventional LSM method.

In order to confirm that our method is superior to the LSM method, we adopted the next performance index J, average error area;

$$J = \frac{1}{T}\sum_{t=0}^{T} e(t), \qquad e = \sqrt{\sum_{i=1}^{4}\left\{\|x_L - x_{Fi}\|^2 + \left(\|y_L - y_{Fi}\|\right)^2\right\}}, \qquad (42)$$

where T means controlling time, $x_L = [x_l, \dot{x}_l]$, $y_L = [y_l, \dot{y}_l]$ mean position and velocity vectors in x direction and y direction of the leader, respectively. x_{Fi} and $\dot{x}_{Fi}$ mean those of the *i*th follower agent. e, J mean sum of consensus errors of between the leader and all the follower agents at time t, average consensus error area, respectively. Table 3 shows comparison of the control performances, that is, transient consensus error and average error area. Table 3 shows our proposed method is superior to the conventional LSM method

VIII. CONCLUSION

In this paper, we proposed the simple and useful method, that is, an adaptive LSM and TSM combined formation controller design method with auto- structuring fuzzy neural network. Additionaly, feature of the proposed method is that all the agents can exchange only information of positions according to the network structure of multi agents like Figure 4, making velocities of their virtual agents, and it has the variable structure to adapt for changes of the environment.

TABLE III. PERFORMANCE COMPARISONS OF BOTH METHODS.

Transient consensus error

agent	a_1	a_2	a_3	a_4	Total
LSM	0.148	0.161	0.234	0.24	0.78
TLSM (Proposed)	0.14	0.145	0.226	0.237	0.748

Average consensus error area

agent	a_1	a_2	a_3	a_4	Total
LSM	0.356	0.368	0.535	0.541	1.8
TLSM (Proposed)	0.316	0.321	0.478	0.489	1.604

REFERENCES

[1] Z. G. Hou, L. Cheng, and M. Tan, “Decentralized robust adaptive control for the multiagent system consensus problem using neural networks”, IEEE Transactions on Systems, Man, Cybernetics-Part B: Cybernetics, vol. 39, no. 3, pp. 636-647.

[2] Q. Song, J. Cao, snd W. Yu, ”Second-order leader-following consensus of nonlinear multi-agent systems via pinning control”, Systems & Control Letters 59, 2009, pp. 553-562.

[3] L. Cheng, Z. G. Hou, M. Tan, Y. Lin, and W. Chang, “Neural-network-based adaptive leader-following control for multiagent systems with uncertainties”, IEEE Transactions on Neural Networks, vol. 21, no. 8, 2010, pp. 1351-1358.

[4] M. Defoort, T. Floquet, A. Kokosy, and W. Perruquetti, ”Sliding-mode formation control for cooperative autonomous mobile robots”, IEEE Transactions on Industrial Electronics, vol. 55, no. 11, 2008, pp. 3944-3953.

[5] F. Morbidi and D. Prattichizzo,”Sliding mode formation tacking control of a tractor and trailer –car system”, Robotics Science and Systems online proceedings, 2007.

[6] R. Cui, S. S. Ge, B. V. E. How, and Y. S. Choo, “Leader-follower formation control of underactuated autonomous underwater vehicles”, Ocean Engineering, vol. 37, no. 17-18, 2010, pp. 1491-1502.

[7] W. Yu, G. Chen, and M. Cao, “ Distributed leader-follower flocking control for multi-agent dynamical systems with time-varying velocities ” , System & Control Letters 5, 2010, pp. 543-552.

[8] Y. H. Chang, C. W. Chang, C. L. Chen, and C. W. Tao, “Fuzzy sliding-mode formation control for multirobot systems: design and implementation”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 42, no. 2, 2012, pp. 444-457.

[9] Chuan-Kai Lin,”Robust adaptive critic control of nonlinear systems using fuzzy basis function networks: An LMI approach”, Information Sciences, 177, 2007, pp. 4934-4946.

[10] C. L. Chen, G. X. Wen, Y. J. Liu, and F. Y. Wang, “Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks”, IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 6, 2014, pp. 1217-1226.

[11] J. L. Lin, K. S. Hwan, and Y. L. Wan,”A simple scheme for formation Control based on weighted behavior learning ”, IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 6, 2014, pp. 1033-1044.

[12] X. Yu, M. Zhihong, Y. Feng, and Z. Guan, ”Nonsingular Terminal sliding mode control of a class of nonlinear ”, IFAC, 15th Triennial World Congress, Barcelona, Spain, 2002.

[13] A. M. Zou, K. D. Kumar, Z. G. Hou, and X. Liu, ”Finite time attitude following control for spacecraft using terminal sliding mode and Chebyshev neural network”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 41, no. 4, 2011, pp. 950 - 963.

[14] Y. H. Chang, C. Y. Yang, W. S. Chan, C. W. Chang, and C. W. Taol, “Leader-following formation control of multi-robot systems with adaptive fuzzy terminal sliding-mode controller”, IEEE International Conference on System Science and Engineering, 2013.

[15] M. Obayashi, Y. Yokoji, S. Uchiyama, L. B. Feng, T. Kuremoto, and K. Kobayashi, “Intelligent Following control method of a leader by Groups of agents with nonlinear dynamics”, Proc. of 11th International Conference on Control, Automation and Systems, Korea, 2011.

[16] K. H. Cheng, “Auto-structuring fuzzy neural system for intelligent control”, Journal of Franklin Institute, 346, 2009, pp. 267-288.