Adaptive Formation Control and Target Tracking in a Class of Multi-Agent Systems

Sabahat Duran and Veysel Gazi

Abstract—In this article a direct adaptive fuzzy control methodology is developed to handle formation control and target tracking problems in a class of multi-agent systems with nonlinear and uncertain dynamics. The proposed adaptive method guarantees stability and achievement of the desired tracking and formation tasks. The effectiveness of the algorithm is verified with numerical simulations.

I. INTRODUCTION

The field of coordination and control of multi-agent dynamic systems has gained a lot of popularity in the past decade. It has witnessed a lot of developments some of which have been summarized in [1]. One of the problems considered is the problem of formation control or basically achieving and keeping a predefined geometrical shape by the multi-agent system. It has been addressed by many researchers using various methodologies. Potential functions method is one of the methods which is effectively used for formation control and/or tracking tasks [2], [3], [4], [5], [6], which can also be combined together with other conventional methods such as sliding mode control [2], [4], [6] to improve the performance of the control action or to provide robustness to uncertainties and disturbances. Behavior based control is another methodology which is discussed in [7] where various formation control strategies are presented for different applications. In [8] formation control and trajectory tracking problems are formulated as a nonlinear servomechanism problem and solved by using decentralized controllers. Also, control Lyapunov functions and Lyapunov based approaches are effectively used for formation control and coordinated behaviors of swarm systems [9], [10]. Adaptive control approaches have been also considered for control of coordinated behaviors in multi-agent systems [11], [12], [13], [14], [15], [16], [17]. When system uncertainties or unknown disturbances are present in the system dynamics one needs to use either a robust strategy to suppress the effect of the uncertainties or disturbances or an adaptive strategy to compensate these effects by changing the controller accordingly. Usually, the adaptive methods parameterize the uncertainties or the control action and adapt the parameters to achieve the control objectives such as coordinated motion and this is the approach taken in most of the above studies. In the cases in which a proper parametrization of the uncertainties or the controller is not known one can use neural networks or fuzzy systems to estimate and to counter affect the uncertainties. It is well known that neural networks and fuzzy systems are universal approximators which can arbitrarily closely approximate any smooth function on a compact set. Because of these valuable properties adaptive control using neural networks and fuzzy systems has been popular in the nonlinear control literature [18], [19], [20], [21]. There are two common adaptive control schemes which are referred to as direct and indirect adaptive control [22], [23]. An indirect adaptive controller estimates the unknown plant dynamics and controls the system using a certainty equivalence based controller. A direct adaptive controller directly tries to approximate an ideal controller for the system without estimating the plant dynamics. It is possible also to augment the adaptive controller with other nonlinear control techniques to improve performance or to guarantee stability and/or boundedness. In [21], indirect and direct adaptive controllers based on linearly parameterized fuzzy systems or neural networks and augmented with sliding mode and bounding control terms are developed for nonlinear systems with uncertain dynamics. Moreover, the two control methodologies are applied for car following problem by using a fuzzy system.

In this article we consider the problem of tracking a desired trajectory in a desired formation by a class of multi-agent systems. The agent dynamics are assumed to be transformable to the normal form and to include model uncertainties and/or disturbances. To solve this problem we utilize the results in [21]. In particular, we use the weighted sum of the formation and tracking errors as the main signal to be regulated to zero and use a direct adaptive fuzzy controller augmented with sliding mode and bounding terms.

II. PROBLEM DEFINITION

Consider a multi-agent system composed of $N$ agents whose states evolve based on

\[
\begin{align*}
\dot{x}_i &= f_i(x_i) + g_i(x_i)u_i \\
y_i &= h_i(x_i)
\end{align*}
\]

where $x_i \in \mathbb{R}^n$ refers the state vector, $u_i \in \mathbb{R}^m$ is the control input and $y_i \in \mathbb{R}^m$ is the output of agent $i$. The functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$, and $h_i : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are smooth functions for all $i = 1, \ldots, N$. We assume that in the region of interest the agent dynamics of all agents have a well defined vector relative degree \{r, r, ..., r\} and can be
transformed into the form
\[
\dot{z}_i = f_{oi}(z_i, \xi_i)
\]
\[
\dot{\xi}_{i,1} = \xi_{i,2}
\]
\[\vdots\]
\[
\dot{\xi}_{i,r-1} = \xi_{i,r}
\]
\[
\dot{\xi}_{i,r} = \left[\alpha_{ki}(t) + \alpha_i(x_i)\right] + \ldots \text{ and sliding mode control terms. For this purpose, we define the output error for each agent as the weighted}
\]
where \(\xi_{i,j} \in \mathbb{R}^m, j = 1, \ldots, r\). \(\xi_i^T = [\xi_{i,1}, \ldots, \xi_{i,r}] \in \mathbb{R}^m\), and \(z_i \in \mathbb{R}^{n-rm}\). In these dynamics it is assumed that the functions \(\alpha_{ki}(t)\) are known. In contrast, the functions \(\alpha_i(x_i)\) and \(\beta_i(x_i)\) are assumed to be unknown. Since the agents have well defined relative degree the \(m \times m\) matrices \([\beta_i(x_i)]\) are always non-singular and assumed to satisfy \(0 < \beta_i \leq \|\beta_i(x_i)\| < \infty\). Moreover, it is assumed that within the region of operation we have \(\|\dot{\xi}_i(x_i)\| = \|\frac{\partial \beta_i}{\partial x_i} \xi_i\| \leq B_i(x_i)\) for some known functions \(B_i(x_i) \geq 0\). Note that if \(\|\beta_i(x_i)\|\) is a constant matrix this assumption is trivially satisfied. Furthermore, we assume that for all \(i\) the agent dynamics are minimum phase and the zero dynamics of the agents are exponentially stable.

In this article we consider the problem of tracking a reference trajectory in a predefined formation by a swarm of agents. In other words, the agents have to acquire a predefined geometric shape and track a reference trajectory while maintaining the formation. The reference trajectory can be thought of as a virtual leader which is followed or as a virtual target which is pursued by the agents. To achieve this we set the problem such that the center of the agents tracks the virtual target (which also results in enclosing the target by the swarm) while maintaining the geometric formation. It is assumed that each individual/agent knows its position and the relative positions of the other individuals/agents and the virtual leader/target. Note that although we develop the strategy under this assumption it is easy to relax it. In particular, the strategy should work under other interconnection topologies which preserve rigidity of the formation but are not necessarily complete graphs. The outputs of the agent dynamics are assumed to represent their positions and are the variables with respect to which the formation will be defined. In other words, the formation will evolve in the output space of the agent dynamics. In order to satisfy the formation keeping and trajectory tracking objectives it is required that for all \(i\) and \(j\) the outputs/positions of the agents satisfy the constraints
\[
\lim_{t \to \infty} d_{ij} - (y_i - y_j) = 0 \quad (3)
\]
and
\[
\lim_{t \to \infty} d_{it} - (y_i - y_t) = 0 \quad (4)
\]
where \(y_i, y_j,\) and \(y_t\) are the outputs/positions of \(i^{th}\) agent, \(j^{th}\) agent, and the virtual target, respectively, \(d_{ij}\) are the desired relative position vectors between the agents, and \(d_{it}\) are the desired relative position vectors between the agents and the target. Note that in order for the formation to be feasible the constraints should not be conflicting. For example for all pairs \((i, j)\) one needs \(d_{ij} = -d_{ij}\).

Note that in the above setting the number of inputs and outputs of the agents must be the same, i.e., \(u_i, y_i \in \mathbb{R}^m\) for all \(i\). \(\mathbb{R}^m\) is the space in which the output trajectories of the agents (viewed as the positions of the agents) and therefore the geometric formation evolve. Similarly, it is assumed that the number of internal states and relative degrees of the agents are also the same, i.e., the same \(n\) and \(r\) for all agents. However, this assumption can easily be relaxed allowing inclusion of heterogeneous agents with different number of internal states and relative degrees.

In this article we assume that the output of the virtual leader and its derivatives are bounded. Moreover, we assume that every agent knows the output of the virtual leader and its derivatives up to \(r\), where \(r\) is its relative degree (of each component in its output vector). Note that this assumption can be relaxed with the use of a high gain observer.

The control inputs to the agents need to be designed in order to satisfy the conditions in equations (3) and (4), which result in satisfaction of the control objectives, i.e., simultaneously achieving the desired formation and tracking the virtual leader/target. In this paper, inspired by the work in [21], we develop a direct adaptive control strategy which can be used to solve this problem, despite the uncertainties in the agent dynamics. In other words, this paper constitutes application of the procedure developed in [21] to solve the simultaneous formation control and trajectory tracking problem for a class of multi-agent systems.

### III. Controller Design

It is well known that for the agent dynamics in (2), provided that the functions \(\alpha_{ki}(t), \alpha_i(x_i),\) and \(\beta_i(x_i)\) are known, the corresponding feedback linearizing controllers
\[
u_i^* = [\beta_i(x_i)]^{-1} \left[-(\alpha_{ki}(t) + \alpha_i(x_i)) + \nu_i(t)\right], \quad i = 1, \ldots, N
\]
convert the system in the form
\[
\dot{z}_i = f_{oi}(z_i, \xi_i)
\]
\[
\dot{\xi}_{i,1} = \xi_{i,2}
\]
\[\vdots\]
\[
\dot{\xi}_{i,r-1} = \xi_{i,r}
\]
\[
\dot{\xi}_{i,r} = \nu_i(t)
\]
and with appropriate choice of \(\nu_i(t)\) (as a potential functions based controller, for example) the objectives of formation control and trajectory tracking can easily be achieved. In other words, for every agent \(i\) we know that an ideal controller \(u_i^*\) as in (5), which leads to satisfaction of the objectives, exists. However, we do not know it since we do not know the values of \(\beta_i(x_i)\) and \(\alpha_i(x_i)\). In this paper we approximate the ideal controller \(u_i^*\) using an adaptive fuzzy system. Moreover, as in [21], in order to guarantee boundedness and stability we augment the controller with bounding and sliding mode control terms. For this purpose we define the output error for each agent as the weighted
sum of the error caused by the relative distances to all the other agents and the error caused due to the relative distance to the virtual leader/target. In other words, the output error vector for each agent is defined as

\[ e_{0i} = k_i (d_{it} - (y_i - y_t)) + k_f \sum_{j=1}^{N} (d_{ij} - (y_i - y_j)), \quad 1 \leq i \leq N, \]

where \( k_f \) and \( k_t \) are the coefficients which determine the weight/importance of the corresponding formation keeping and trajectory tracking tasks. In order to force \( e_{0i} \) to zero for all agents, the control inputs \( u_i \) of the agents are defined in the form

\[ u_i = \hat{u}_{ai} + u_{si} + u_{bi}, \quad 1 \leq i \leq N, \]

where \( \hat{u}_{ai} \) represent an adaptive control term, \( u_{si} \) represents a sliding mode control term, and \( u_{bi} \) represents a bounding control term. These control terms will be discussed in more detail below.

A. Adaptive Control Term

The adaptive control term tries to approximate the ideal controller \( u_i^* \) using a fuzzy system. Since the best parameters of the fuzzy system are not known a priori an adaptive strategy is used to determine its parameters. In this manner, it forces the tracking error to go to a small neighborhood of zero. We use a Takagi-Sugeno type fuzzy system [23], [22] in the approximator which is discussed below.

A fuzzy system is a universal approximator that adds heuristics and expert knowledge about the operation of the system to be controlled. It comprises linguistics instead of crisp values and an IF-THEN rule base. The general operation is in three steps: (i) fuzzification, in which the crisp values are fuzzified; (ii) inference, in which the rules which are constituted by expert knowledge are activated; (iii) defuzzification, in which the output of the fuzzy system is converted to an applicable crisp value. The membership functions specify the weights of the activated rules and have a wide variety according to the applications but the commonly used ones are Triangular and Gaussian membership functions. The rule \( j \) of the fuzzy system for agent \( i \) can be defined as

\[ R_{ij}: \quad \text{if } s_{i1} \text{ is } L_{i1}^k \text{ and } ... \text{ and } s_{ip} \text{ is } L_{ip}^k \text{ then } c_{ij} = a_{i0} + a_{i1} \theta_i(S_i) + ... + a_{iM-1} \theta_{M-1}(S_i) \]

where \( s_i = [s_{i1}, s_{i2}, ..., s_{ip}]^T \) is the input vector for the fuzzy system of agent \( i \) and \( c_{ij} \) is the output of its \( j \)th rule, \( j = 1, ..., N_i \). For every input \( s_{iq} \) there is a set of corresponding possible linguistic variables (fuzzy sets) \( L_{iq}^1, ..., L_{iq}^q \) which can be expressed in the form

\[ L_{iq}^k = (s_{iq}, \mu_{L_{iq}^k}(s_{iq})): s_{iq} \in \mathbb{R}, \quad 1 \leq i \leq N, \]

where \( k = 1, ..., q \) denotes the label of the linguistic value or basically denotes the \( k \)th fuzzy set of the \( q \)th input of agent \( i \) and \( \mu_{L_{iq}^k}(s_{iq}) \) is the corresponding membership function. Similarly, \( \theta_i(S_i), ..., \theta_{M-1}^i(S_i) \) are input functions which are Lipschitz continuous and \( S_i \) represents the variables of interest which include the state of the agents \( x_i \), the output error \( e_{0i} \) and its derivatives, and the trajectory (the output and its derivatives) of the virtual leader. Note that the above description is a description of a multi-input-single-output fuzzy system. In cases in which multi-input-multi-output fuzzy systems are needed several multi-input-single-output fuzzy systems can be used in parallel.

There are various methods for calculating the defuzzified output of a fuzzy system one of which is the center-average method

\[ \hat{y}_i = \frac{\sum_{j=1}^{N_i} c_{ij} \mu_{ij}}{\sum_{j=1}^{N_i} \mu_{ij}}, \quad 1 \leq i \leq N \]

where \( \mu_{ij} \) is the output membership function for rule \( R_{ij} \) which can be calculated using the product operator as

\[ \mu_{ij} = \mu_{L_{iq}^k}(s_{iq}) \times \cdots \times \mu_{L_{iq}^p}(s_{ip}) \]

Defining \( c_i \) and \( \zeta_i \) as \( c_i = [c_{i1}, ..., c_{iN_i}]^T \) and \( \zeta_i^T = [\mu_{i1}, ..., \mu_{iN_i}] / [\sum_{j=1}^{N_i} \mu_{ij}] \), the output of the fuzzy system for agent \( i \) can be written in the form

\[ \hat{y}_i = c_i^T \zeta_i \]

Also, note that the vectors \( c_i \) can be expressed as

\[ c_i^T = z_i^T A_i \]

where \( z_i \) consists of the Lipschitz continuous functions

\[ z_i = [1, \theta_1(S_i), ..., \theta_{M-1}(S_i)]^T \in \mathbb{R}^{M_i} \]

and \( A_i \in \mathbb{R}^{N_i} \times \mathbb{R}^{M_i} \) is a parameter matrix of the form

\[ \begin{bmatrix} a_{i0} & a_{i1} & \cdots & a_{i,M_i-1} \\ a_{i2} & a_{i1} & \cdots & a_{i,M_i-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{iN_i,0} & a_{i1} & \cdots & a_{i,M_i-1} \end{bmatrix} \]

Then, the output of the fuzzy system is given by

\[ \hat{y}_i = z_i^T A_i \zeta_i \]

In the strategy here the fuzzy systems are used as the adaptive part \( \hat{u}_{ai} \) of the agent controllers. Therefore, for agent \( i \) the output of the fuzzy system is the control input

\[ \hat{u}_{ai} = z_i^T A_i \zeta_i \]

However, since the parameters corresponding to the ideal controllers are not known, as in [21], we adapt the corresponding parameter matrices \( A_i \) using

\[ \dot{A}_i = Q_i^{-1} c_i e_{si}^T \]

where \( Q_i \in \mathbb{R}^{M_i} \times \mathbb{R}^{M_i} \) are positive definite diagonal matrices, \( e_{si} \) is the tracking error such that \( e_{si} = k_i^T e_i \) and \( e_i = [e_{0i}, e_{0i}, ..., e_{0i(q-1)}]^T \) and \( k_i = [k_{i0}, ..., k_{ir-2}, 1]^T \) are gain vectors which are chosen such that the roots of the corresponding polynomials \( T_i(s) = s^{q-1} + k_{i,r-2}s^{r-2} + ... + k_{i1}s + k_{i0} \) are in the open left half plane to ensure the output to be asymptotically stable.

This update law does not guarantee the boundedness of the parameter matrix \( A_i \). Therefore, in order to guarantee that
$A_i \in \Omega_i$ for some $\Omega_i = [A^{\text{min}}_i, A^{\text{max}}_i]^{N_i \times M_i}$ or basically the components of $A_i$ are within $[A^{\text{min}}_i, A^{\text{max}}_i]$ one can saturate or project the values of the elements of $A_i$ within that set. One possible strategy for that, which was also used in [21], is to use an update law of the form

$$\dot{A}_i = Q_i^{-1} \hat{A}_i$$

(11)

where $\hat{A}_i$ is a matrix the elements of which are determined as

$$\hat{a}^i_{p,m} = \begin{cases} 0, & \text{if } a^i_{p,m} \notin (A^{\text{min}}_i, A^{\text{max}}_i) \\
\hat{a}^i_{p,m} (a^i_{p,m} - a^{ic}_{p,m}) > 0, & \text{otherwise} \end{cases}$$

(12)

where $\hat{a}^i_{p,m}$ is the $(p,m)$'th element of $\bar{z}^i e_{si}$ in (10) and $a^{ic}_{p,m}$ is an element of a fixed matrix $A^c_i \in (A^{\text{min}}_i, A^{\text{max}}_i)$, and $p = 1, \ldots, N_i$ and $m = 1, \ldots, M_i$. Note that the update law in (11)-(12) sets the value of $\hat{a}^i_{p,m} = 0$ if its corresponding parameter $a^i_{p,m}$ is outside of the allowed region $(A^{\text{min}}_i, A^{\text{max}}_i)$ and the calculated value of $\hat{a}^i_{p,m}$ is such that the update will be in the direction of further divergence from $(A^{\text{min}}_i, A^{\text{max}}_i)$. In the case in which the $a^i_{p,m}$ is inside of the allowed region $(A^{\text{min}}_i, A^{\text{max}}_i)$ or the the calculated value of $\hat{a}^i_{p,m}$ is such that the update will be in the direction towards $(A^{\text{min}}_i, A^{\text{max}}_i)$ the update is performed with $\hat{a}^i_{p,m} = a^i_{p,m}$. With this update rule the parameters of the fuzzy systems are projected within the sets $\Omega_i$ and are guaranteed to be bounded.

B. Bounding Control Term

The bounding control term is necessary for bounding the states and the output of the system. The boundedness is quite important for asymptotic stability of the tracking error $e_{si}$. Let $M_e$ be a constant bound for tracking error $e_{si}$ such that when $|e_{si}| \geq M_e$ the bounding controller is activated. Then, following the procedure in [21], the bounding controller can be defined as

$$u_{bi} = \Pi_i(e_{si}) k_{bi}(t) \text{sign}(e_{si})$$

(13)

where

$$k_{bi}(t) = |\tilde{a}_{si}| + |u_{si}| + \tilde{a}_i(x_i) + |\nu_i|$$

(14)

and $\tilde{a}_i(x_i) \geq |a_i(x_i)|$ and $\nu_i(t)$ are specified as

$$\nu_i(t) = y_t^{(r)} + \eta_i e_{si} + \bar{e}_{si} + \alpha_i(t)$$

where $y_t^{(r)}$ is the $r$'th derivative of the output of the virtual leader, $\eta_i$ is a constant, and $\bar{e}_{si} = [\bar{e}_{si0}, \ldots, \bar{e}_{sir-1}]^T$. Note also that the operations sign and $\cdot$ in the above equations are performed elementwise on their input vectors. The $j$'th element $\Pi_j(t)$ of the function $\Pi_j(e_{si})$ can be expressed as

$$\Pi_j(t) = \begin{cases} 1, & \text{if } M_e \leq |e_{sj}| \\
\frac{|e_{sj}| + \epsilon_M - M_e}{\epsilon_M}, & \text{if } M_e - \epsilon_M \leq |e_{sj}| < M_e \\
0, & \text{otherwise} \end{cases}$$

where $\epsilon_M$ is a constant such that $0 < \epsilon_M \leq M_e$. The bounding control term guarantees that the fuzzy system performs function approximation in a compact set, which is needed for guaranteeing its universal approximation property. Otherwise, in order to achieve approximation of the controller function by a fuzzy system on an unbounded set one would probably need a larger parameter space and infinite number of rules.

C. Sliding Mode Control Term

The sliding mode control term is effective on decreasing the approximation error of the ideal controller for the system. This term is not directly like conventional sliding mode controllers and it does not force the error to slide on a sliding surface. To guarantee Lyapunov stability, the sliding mode control term is selected as

$$u_{si} = k_{si}(t) \text{sign}(e_{si})$$

(15)

where

$$k_{si}(t) = \frac{B_i(x_i)|e_{si}|}{2\beta_i^2} + D_{ui}(x_i)$$

(16)

and $D_{ui}(x_i)$ is a known upper bound for the approximation error of the ideal controller in equation (5). In other words, defining $d_{ui}(x_i) = u_i^\star - \hat{u}_i$ it satisfies $|d_{ui}(x_i)| \leq D_{ui}(x_i)$. Then the robustness properties of the sliding mode controller guarantee that the approximation errors will be suppressed.

With the controller in (8) composed of adaptive, bounding, and sliding mode control terms one can show that, provided that the formation and tracking constraints are not conflicting, the output errors $e_0$ will converge to zero resulting in achieving the desired formation and simultaneously tracking the virtual leader.

IV. Simulation Results

In this section the direct adaptive control method discussed in the previous sections is applied on a multi-agent system consisting of $N = 6$ point-mass agents moving in a 2-dimensional Euclidian space. Also we assume that there is model uncertainty/disturbance acting on the agent dynamics. The motion dynamics of the agents $i = 1, \ldots, 6$ are assumed to be in the form

$$\begin{align*}
\dot{x}_i &= v_i \\
\dot{v}_i &= \alpha_i + \beta_i u_i \\
y_i &= x_i
\end{align*}$$

(17)

where the state of agent $i$ is $[x_i, v_i]^T \in \mathbb{R}^4$, $y_i = x_i \in \mathbb{R}^2$ refers to its position, $v_i \in \mathbb{R}^2$ is its velocity, and $u_i \in \mathbb{R}^2$ is its control input. The values/expressions of $\alpha_i$ and $\beta_i$ are assumed to be unknown. In the simulations we utilize periodic $\alpha_i$ and constant $\beta_i$ for all agents. In particular, we use $\alpha_i = [(i - 0.5) \cos(2t), (i + 0.5) \sin(2t)]^T$ and $\beta_i = i$, which are agent dependent. As can be seen the agent dynamics are fully linearizable (i.e., no zero dynamics) and the relative degree of the agent dynamics is $\{r, r\} = \{2, 2\}$ for all agents. The six agents are required to form an equilateral triangle formation with sides equal to 2 which, depending on the relative desired positions, results in inter-agent distances equal to $2$, $\sqrt{3}$, or 1. Simultaneously, the center of this
triangle is required to track a virtual leader/target. For the tracking task, the dynamics of the virtual leader is specified as
\[
\begin{align*}
\dot{x}_{1t} &= 0.2 \cos(2t) \\
\dot{x}_{2t} &= 0.95 \cos(0.5t)
\end{align*}
\]
which start its motion form the initial state \(x_{11}, \dot{x}_{11}, x_{21}, \dot{x}_{21} = [5, 5, 0.1, 0] \). The coefficients in (7), which determine the weights of the formation and the tracking tasks, are selected as \(k_f = 5 \) and \(k_h = 50 \). We treat the \(x \) and \(y \) components of the 2-dimensional desired relative-position vectors \(d_{ij} \) separately. With that purpose we define the \(d_x \) and \(d_y \) matrices as
\[
d_x := 
\begin{bmatrix}
0 & 0.5 & 1 & 0 & -1 & -0.5 \\
-0.5 & 0 & 0.5 & -0.5 & 1.5 & -1 \\
-1 & -0.5 & 0 & -1 & -2 & -1.5 \\
0 & 0.5 & 1 & 0 & -1 & -0.5 \\
1 & 1.5 & 2 & 1 & 0 & 0.5 \\
0.5 & 1 & 1.5 & 0.5 & -0.5 & 0 \\
\end{bmatrix}
\]
\[
d_y := 
\begin{bmatrix}
0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\
-\frac{\sqrt{3}}{2} & 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 \\
-\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 & 0 \\
-\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 \\
-\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\
-\frac{\sqrt{3}}{2} & 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 \\
\end{bmatrix}
\]
where the \((i,j)^{th}\) element of the matrices represents the desired relative distance between the \(i^{th}\) and the \(j^{th}\) agents in the respective dimension. In this manner we decouple the problem for controller design for the \(x \) and \(y \) dimensions and design separate controller for each of them. For all agents we use fuzzy systems consisting of \(R = 9 \) input membership functions which are selected as gaussian membership functions which are uniformly distributed within \([-10, 10]\) such that the rightmost and the leftmost membership functions are centered at 10 and \(-10\), respectively and are saturated and extended. The spreads of the membership functions are chosen equal and are given by \(20/(R-1) = 2.5 \). The input of the fuzzy system is \(e_{0i}\) for each agent and \(M_i = 2 \), the output of the fuzzy system is determined by \(z_i = [1, \nu_i(t)]^T \). The rule base of the fuzzy systems for the agents are the same and are in the form
\[
R_{ij}: \quad \text{If } e_{0i} = L_j \quad \text{Then } e_{ij} = a_{ij,0}^i + a_{ij,1}^i \nu_i(t) \]
where \(j = 1, ..., N_i = 9 \) represents the rule number (there is only one input and each input membership function corresponds to a rule) and \(i = 1, ..., N \) represents the agent number. The positive definite diagonal matrix \(Q_i \in \mathbb{R}^{2 \times 2} \) used for parameter update in (11) is chosen as \(Q_i = I \), where \(I \) is the identity matrix. \(e_{si} \) is tracking error and \(e_{si} = k_i^T e_i, \) \(e_i = [e_{0i}, e_{0i}]^T, \) \(k_i = [k_{0i}, 1]^T \) and \(k_{0i} = 10 \) for all \(i = 1, ..., N \). For the simulations below we do not employ the parameter projection in (11)-(12). The bounds for \(e_{si} \) are defined as \(M_e = 10 \) for all agents \(i \) and \(\epsilon_{si} = 1 \). For the given system, as the bounds in (14) we use \(\alpha_i = 8.5 \) and \(\beta_i = 0.8 \) for all agents \(i \). \(\bar{e}_{si} = [\bar{e}_{0i}]^T [k_{0i}] \) and \(\eta_i = 1 \). \(\nu_i(t) \) is defined in the form of \(\nu_i(t) = \dot{x}_i + e_{si} + [\bar{e}_{0i}]^T [k_{0i}] \).

In the example, in equation (17) \(\beta_i \) is a constant implying that its derivative is zero and \(B_i(x_i) = 0 \). The \(k_{si}(t) \) gain in (16) can be expressed as
\[k_{si}(t) = D_{ui}(x)\]
\(D_{ui}(x) \) is chosen as 0.5. The initial positions for the agents are selected randomly from within \([0, 10]^2 \) and the initial velocities for the agents are set to zero. The initial position of the target is \(x_t(0) = [5, 5]^T \) and the initial values for the fuzzy system parameters are \(e_{p, m}(0) = 0 \) for \(i = 1, ..., 6 \), \(p = 1, ..., 5 \) and \(m = 1, 2 \). In the consequence of these choices, the movement of the center of the triangle and position of the virtual leader/desired path are shown in Figure 1. The simulation runs for 30 seconds. The triangle formation and the positions of the agents and the center at various time instants are also shown in the figure. As can be seen from the figure the agents have formed the desired formation and track the virtual leader in the formation. In Figure 2 the formation at the last instant of the simulation run is shown where the triangle formation can be realized more clearly. The little circles show the final positions of the agents, whereas the square at the center shows the position of the virtual leader.

Fig. 1. The position of the agents and motion of the center.

Fig. 2. The final positions of the swarm members.

Figure 3 shows the plots of the inter-agent distances with
respect to time. As can be seen from the figure the inter-agent distances converge to the values 1, $\sqrt{3}$, and 2, which are the desired values in the desired formation. In this setting three agents locate in corners and the other three locate in the middle of the edges of the triangle.

![Fig. 3. The distance values between the agents.](image)

The tracking error which is the distance between the center of the triangle and the virtual target/desired trajectory quickly converges to a small neighborhood of zero as shown in Figure 4. In other words, the formation acquisition/maintenance and the trajectory/virtual leader tracking objectives are simultaneously achieved as can be observed from the plots.

![Fig. 4. The tracking error.](image)

V. CONCLUDING REMARKS

In this paper, inspired by the work in [21], a direct adaptive fuzzy controller was developed for solving the problem of formation control and target tracking in a class of multi-agent systems with uncertain agent dynamics. We use a Takagi-Sugeno fuzzy system in the adaptive part of the control action. The controller is also augmented with bounding and sliding mode terms. The bounding term serves to bound the states and the agents, whereas the sliding mode term helps to suppress the approximation uncertainties and decrease the error. The method guarantees boundedness of the states and the outputs of the agents as well as the stability of the system. The method is applied to a multi-agent system consisting of 6 agents to track a desired trajectory in an equilateral triangle formation and successful results are obtained.

REFERENCES