Indirect Adaptive Formation Control with Nonlinear Dynamics and Parametric Uncertainty

Samet GÜLER[1], Nasrettin KÖKSAL[1], Barış FİDAN[1], and Veysel GAZİ[2]
[1]Mechanical and Mechatronics Engineering Department, University of Waterloo, Canada
{sguler, nkoksal, fidan}@uwaterloo.ca
[2]Department of Electrical and Electronics Engineering, Istanbul Kemerburgaz University, Turkey
veysel.gazi@kemerburgaz.edu.tr

***Abstract*—This paper focuses on coordination of multi-agent systems with agents having holonomic nonlinear dynamics and uncertain system parameters. An indirect adaptive control scheme composed of a parameter estimator and a feedback-linearization based control law is designed based on the certainty equivalence principle. (i) Simple linear and (ii) sliding mode control structures are used in the outer control loop. The controllers successfully satisfy the objectives of formation maintenance and trajectory tracking. Simulation results for a sample two-dimensional formation are presented, demonstrating the performance of the proposed control scheme.**

I. Introduction

Being of interest to many researchers, control of multi-agent systems has been a challenging area of study for years [1]–[10]. An analysis that contains parameters which are not known or cannot be measured most likely makes use of a robust [11], [12] or an adaptive control strategy [3], [13]. Direct and indirect adaptive control approaches [13] are two main approaches designed for estimation of the unknown parameters, each having different advantages.

Modeling of a single agent of a swarm may result in a nonlinear system of equations including unknown dynamics as in the case of [3]. Following the results in [14], the work in [3] proposes a direct adaptive fuzzy algorithm with feedback linearization to control a multi-agent system moving in an n-dimensional space maintaining a specific formation. The proposed algorithm is verified with simulations for a six-agent system moving in the plane. In this article, we consider the same problem formulation as in [3]. However, our work differs from the work in [3] in application of the adaptive controller and the controller design process. In particular, here we consider an indirect adaptive control approach and utilize least squares estimation instead of a fuzzy system. Moreover, we consider and compare two different control techniques in the outer control loop, namely (i) linear model reference and (ii) sliding mode control.

In multi-agent dynamic systems in general, and in formation control problems in particular, the agents usually have limited sensing capabilities, so information acquisition can be restricted to inter-agent measurements within a local neighborhood. In such a case, the developed control algorithms need to rely only on local information and the behavior of neighboring agents to satisfy the objectives such as formation keeping, obstacle avoidance, reference tracking [2]. On the other hand, some works assume that an agent is able to detect its environment in a perfect manner though this assumption brings the demand of using additional tools for sensing in applications. Virtual leader definition [7], [8], [15] inside the formation is one of the approaches that utilize this assumption. In this approach, agents are controlled so that they position themselves around the virtual leader as desired using the distance and/or bearing measurements.

In this work, we develop an adaptive controller for a multi-agent system applying virtual leader approach. The assumption that the agents know their positions relative to the other ones allows us to define a virtual target whose dynamics are omitted. Our objective is to maintain the formation established by the agents and at the same time satisfy reference trajectory tracking for the whole swarm. The need for using adaptive algorithm comes from the existence of unknown parameters in agent dynamics. We solve the control problem using indirect estimation algorithm and feedback linearization. We then design an outer loop controller for the linearized plant using (i) linear model reference and (ii) siding mode control techniques. Success of the linearization procedure is related to the stability of the estimation algorithm, which is also discussed in a proposition. At the end, we simulate a specific 6-agent formation instance on the $x-y$ plane comparing the performance of the linear model reference and sliding mode controllers.

II. Problem Definition

A. System Equations

We study multi-agent systems represented by

$$\begin{aligned} \dot{x}_i &= f_i(x_i) + g_i(x_i)u_i, \\ y_i &= h_i(x_i), \end{aligned} \tag{1}$$

where $i = 1, \ldots, N$ denotes the agent number, $x_i \in \mathbb{R}^n$ is the state vector, $u_i \in \mathbb{R}^m$ is the input vector, and $y_i \in \mathbb{R}^m$ is the output vector of agent i. The functions $f_i \in \mathbb{R}^n \to \mathbb{R}^n$, $g_i \in \mathbb{R}^n \to \mathbb{R}^{n\times m}$, and $h_i \in \mathbb{R}^n \to \mathbb{R}^m$ are unknown, but assumed to satisfy the following assumption:

978-1-4673-5769-2/13/$31.00 ©2013 IEEE

Assumption 1: For each agent i, the functions $f_i,\ g_i,\ h_i$ are differentiable up to the order r, and the system model (1) has relative degree $r = n$ with no zero dynamics.

The above assumption basically implies that the sum of the relative degrees r_j, $j = 1, ...m$ for the input-output channels is n (i.e., $\sum_{j=1}^{m} r_j = n$). In this paper we develop the results as if $r_j = r_k$ for all j and k. However, this is not really necessary and the procedure can easily be modified and applied to systems with agents which have different relative degree for different input-output channels.

For the sake of controller design, under Assumption 1, the agent dynamics (1) is transformed to the canonical form [13], [16]

$$\begin{aligned} \dot{\xi}_{i,1} &= \xi_{i,2} \\ &\cdots \\ \dot{\xi}_{i,r-1} &= \xi_{i,r} \\ \dot{\xi}_{i,r} &= \alpha_i(t, \xi_i) + \beta_i(t, \xi_i) u_i \\ y_i &= \xi_{i,1}, \end{aligned} \tag{2}$$

where $\xi_i = [\xi_{i,1}^T, \ldots, \xi_{i,r}^T]^T \in \mathbb{R}^n$, $\xi_{i,j} \in \mathbb{R}^m$, $j = 1, \ldots, r$. Here, since $f_i,\ g_i,\ h_i$ are unknown, $\alpha_i(t, \xi_i) \in \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$ and $\beta_i(t, \xi_i) \in \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^{m \times m}$ are unknown as well. Nevertheless, the following is assumed:

Assumption 2: All the entries of α_i and β_i are bounded, and β_i is invertible for all ξ_i and t.

Note that it is sufficient that the statement of Assumption 2 holds only in some regions of interest. It is not absolutely necessary for it to hold globally.

B. Control Problem

The main goal of the swarm is to track a reference trajectory and maintain a predefined formation in the m-dimensional output space. In other words, the formation constraints are defined with respect to the output variables of the agents, and the output space $\mathbb{R}^m$ is the space in which the formation will evolve. In many practical applications, this space might be $\mathbb{R}^2$ or $\mathbb{R}^3$, although the procedure holds for higher dimensional spaces as well. A representation of a system under consideration with five agents for the special case of $m = 2$ or basically the $x - y$ plane is depicted in Fig. 1. We define a virtual leader so that the agents satisfy the formation constraint and reference trajectory tracking. The idea behind the virtual leader definition is that, if the relative position vector between the agents and the virtual leader is maintained as required, then both the formation maintain and reference tracking objectives will be satisfied. In other words, denoting the output vector of the virtual leader as y_v, an agent should satisfy

$$y_i(t) - y_v(t) \to d_{i,v}(t) \text{ as } t \to \infty, \quad i = 1, \ldots, N, \tag{3}$$

where $d_{i,v}$ is the desired relative position vector between the i^{th} agent and the virtual target. This condition is sufficient for both the reference trajectory tracking and rigid formation objectives. The vectors $d_{i,v}(t)$ can be constant or time varying. If they are constant vectors, the shape, the size, and the orientation of the formation do not change during motion if the condition (3) is satisfied. By appropriately varying the values of $d_{i,v}(t)$, it is possible to achieve various formation maneuvers as in [17]. The reference trajectory tracked by the virtual leader is assumed to be given. We have one more assumption about distance measurements.

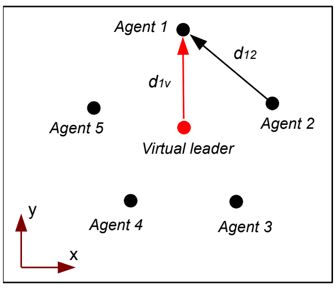

Fig. 1. A representation of a five-agent system with its virtual leader

Assumption 3: The output vector y_i of agent i, $i = 1, \ldots, N$, is the position vector in corresponding dimension and agent i knows the position of all agents $y_j(t)$ at each time instant.

Note that since the output vectors $y_i(t)\ \ \forall i, t$ are known according to Assumption 3, the centroid of the formation can be calculated all the time using $y_c(t) := \frac{1}{N} \sum_{i=1}^{N} y_i(t)$. It is possible to design the algorithm without the need of Assumption 3, by defining the reference relative positions of the agents with respect to the target as if the target is the centroid of the group. In that case the agents will need to know only their relative position to the target. However, in such a case, since the agents don't sense the position of the other agents, means for inter-agent collision avoidance need to be also considered. We now summarize the control problems.

Problem 1: [*Estimation Problem*] Consider the system (2) with the unknown functions $\alpha_i(t, \xi_i)$ and $\beta_i(t, \xi_i)$. Under the Assumptions 1 and 2, for each agent i, generate the estimates of $\alpha_i(t, \xi_i)$ and $\beta_i(t, \xi_i)$ $\forall t \geq t_0$ where t_0 is the initial time.

Problem 2: [*Formation Problem*] Consider the system (2) with the estimates of the unknown functions $\alpha_i(t, \xi_i)$ and $\beta_i(t, \xi_i)$. Under the Assumptions 1, 2, and 3, generate the control signal u_i so that the agents satisfy the condition (3).

An input-output channel of i^{th} agent is illustrated in Fig. 2. In this figure, the controller block controls the linearized plant, combination of the feedback linearizing controller and the nonlinear plant model (2). The path planning algorithm of an agent computes the agent's reference trajectory.

III. Path Planning Algorithm

We use a non-hierarchical, decentralized control architecture in planning the reference path for the agents. Each agent computes its reference "position" considering its desired relative position vector $d_{i,v}$ to the virtual target.

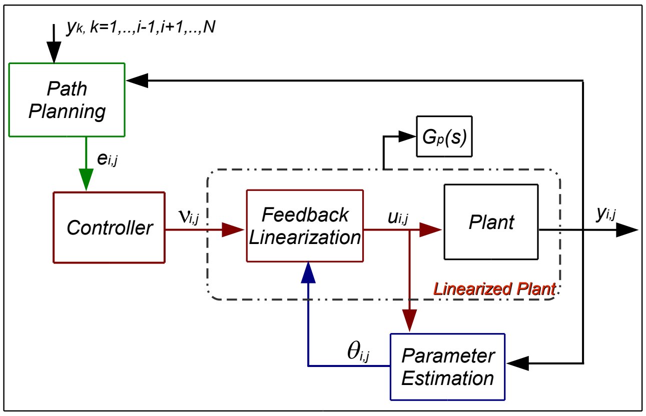

Fig. 2. Channel j of the adaptive controller of agent i

Even though we use a decentralized controller scheme, Assumption 3 states that an agent can measure the positions of the other agents. This may require use of multiple and/or sophisticated sensor structures on the agents. We calculate the position error vector e_i of the i^{th} agent by

$$e_i(t) = d_{i,v}(t) - (y_i(t) - y_v(t)), \quad i = 1, \ldots, N. \tag{4}$$

Therefore, for the j^{th} channel, the reference signal $r_{i,j}$ becomes

$$r_{i,j}(t) = y_{i,j}(t) + e_{i,j}(t), \tag{5}$$

where $y_{i,j}$, $e_{i,j}$ are the j^{th} entries of the vectors y_i and e_i.

IV. Controller Design

In the previous section, two different, specific, albeit related, control problems are presented to meet the reference tracking and formation maintain objectives. We apply least squares algorithm for estimation of the unknown functions α_i and β_i which appear in the agent dynamics in (2) to solve the Problem 1. We separate Problem 2 into two different parts. Employing feedback linearization, we first convert the nonlinear system to a linear one. A linear controller is then synthesized to achieve tracking and formation maintain objectives taking advantage of holonomic motion of the agents.

A. Estimation using Least Squares

Following the results of [13], we design a least squares estimation algorithm to find the estimates of the unknown functions α_i and β_i, treating them as constants. The last line of the dynamic equations (2) is written (with loose notation) in Laplace domain as

$$\dot{\xi}_{i,r} = s^r \xi_{i,1} = s^r y_i = \alpha_i + \beta_i u_i,$$

where $y_i = [y_{i,1} \ldots y_{i,m}]^T$, $i = 1, \ldots, N$. Focusing solely on one element of y_i, we write

$$s^r y_{i,j} = \alpha_{i,j} + \beta_{i,j}^T u_i, \quad j = 1, \ldots, m,$$

with $\alpha_{i,j}$ and $\beta_{i,j}$ being the corresponding unknown function and jth row of the unknown matrix of ith agent, respectively. Treating $\alpha_{i,j}$ and $\beta_{i,j}$ as piecewise constants in time (i.e., assuming time variations are small enough), filtering each side by the stable filter $\frac{1}{(s+\lambda)^r}$, $\lambda > 0$, we obtain

$$\frac{s^r}{(s+\lambda)^r} y_j = \alpha_{i,j} \frac{1}{(s+\lambda)^r}[\mathbf{1}] + \beta_{i,j}^T \frac{1}{(s+\lambda)^r}[u_i], \tag{6}$$

where $\mathbf{1}$ and u_i represent a unit-step signal and the input vector for agent i, respectively. Then, we write a parametric model for (6) as

$$z_{i,j} = \theta_{i,j}^{*T} \phi_i, \quad z_{i,j} = \frac{s^r}{(s+\lambda)^r} y_{i,j}, \tag{7}$$
$$\theta_{i,j}^* = [\alpha_{i,j} \ \beta_{i,j}^T]^T, \quad \phi_i = [\frac{1}{(s+\lambda)^r} 1 \ \frac{1}{(s+\lambda)^r} u_i^T]^T.$$

Note here that the vector $\theta_{i,j}^*$ represents the actual unknown system functions.

We now apply pure least squares algorithm [13] to the parametric model (7) as

$$\dot{\hat{\theta}}_{i,j}(t) = P_i(t)\epsilon_{i,j}(t)\phi_i(t), \quad \hat{\theta}_{i,j}(0) = \hat{\theta}_{i,j,0}, \tag{8}$$
$$\dot{P}_i(t) = -P_i(t)\frac{\phi_i(t)\phi_i(t)^T}{m_i^2(t)} P_i(t), \quad P_i(0) = P_{i,0},$$
$$\epsilon_{i,j}(t) = \frac{z_{i,j}(t) - \hat{z}_{i,j}(t)}{m_i^2(t)}, \quad m_i^2(t) = 1 + \phi_i^T(t)\phi_i(t),$$

where $P_i \in \mathbb{R}^{m+1 \times m+1}$ is the covariance matrix and $P_i = P_i^T > 0$, m_i is the normalizing signal, $\phi_i \in \mathbb{R}^{m+1}$ is the regressor signal for i^{th} agent as described above, and $\epsilon_{i,j}$ is the estimation error for j^{th} row of i^{th} agent. The estimation model is given by

$$\hat{z}_{i,j}(t) = \hat{\theta}_{i,j}(t)^T \phi_i(t), \tag{9}$$

with $\hat{\theta}_{i,j}(t)$ being the estimated parameter vector at time instant t. Stability features of this estimation algorithm are given in the following proposition.

Proposition 4.1: Consider Problem 1. According to Theorem 3.7.2 of [13], the estimation algorithm (8)-(9) applied to the plant model (2) has the following properties:

i) $\epsilon_{i,j}$, $\epsilon_{i,j} m_i$, $\dot{\hat{\theta}}_{i,j} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ and $\theta_{i,j}$, $P_i \in \mathcal{L}_2$,
ii) $\hat{\theta}_{i,j}$ converges to a constant vector,
iii) If $\frac{\phi_i}{m_i}$ is persistently exciting, then $\hat{\theta}_{i,j} \to \theta_{i,j}^*$ as time goes to infinity.

Note that in order for the least squares algorithm to guarantee convergence of the parameter values to their true values it is required that the input signal to the estimator $\phi_i(t)$, or basically the control input $u_i(t)$, be sufficiently rich. This may not be always the case and the least squares algorithm may not guarantee convergence of the parameter values to their true values. However, despite this, the control algorithm is able to achieve the formation control and trajectory tracking objectives. Note also that in addition to the least squares estimation used here there are alternative approaches for estimating the unknown functions α_i and β_i such as neural, fuzzy approaches and others. The basic property that the used approximator should possess is the universal approximation property and should have the ability to approximate arbitrarily closely any smooth function on a compact set.

B. Feedback Linearization

The functions $\hat{\alpha}(t)$ and $\hat{\beta}(t)$ are obtained from the parameter vector $\hat{\theta}(t)$ at each time instant. Using the estimated function values, we apply a controller with which the nonlinear system model (2) becomes linear because a linear model is more suitable in terms of controller design. Let us consider the feedback rule

$$\bar{u}_i(t) = \begin{cases} (\hat{\beta}_i(t,\xi))^{-1}(\nu_i(t) - \hat{\alpha}_i(t,\xi)), & \text{if } \det|\hat{\beta}_i(t,\xi)| > \bar{\beta}_i \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $\hat{\alpha}_i(t,\xi)$, $\hat{\beta}_i(t,\xi)$ are the estimates of the corresponding functions, $\nu(t)$ is the controller output to be synthesized for the resultant linear model, and $\bar{\beta}_i > 0$ is a bound used to avoid possible singularity issues due to the inverse of the parameter matrix $\hat{\beta}_i(t,\xi)$. Then, under the certainty equivalence principle assuming that $\hat{\alpha}_i(t,\xi)$ and $\hat{\beta}_i(t,\xi)$ are close enough estimates of $\alpha_i(t,\xi)$ and $\beta_i(t,\xi)$ the plant model (2) can be represented as

$$\begin{aligned} \dot{\xi}_{i,1} &= \xi_{i,2} \qquad (11)\\ &\cdots \\ \dot{\xi}_{i,r-1} &= \xi_{i,r} \\ \dot{\xi}_{i,r} &= \nu_i \\ y_i &= \xi_{i,1}. \end{aligned}$$

This model can be used for development of the (linear) controller ν_i. Related with the element-wise estimation algorithm, an observation on the feedback rule (10) is that finding the controller $\bar{u}_i(t)$ requires utilizing all the estimated elements $\alpha_{i,j}$ and $\beta_{i,j}$, while each element of the vector ν_i, namely $\nu_{i,j}$, can be designed separately.

After the linearization procedure, in one channel of an agent the linear model (11) is constituted. We now design two different controller structures to solve Problem 2.

C. Model Reference Controller

We now design a model reference controller for the resultant linear model (11) which, in Laplace domain, corresponds to

$$G_{i,j}(s) = \frac{y_{i,j}(s)}{\nu_{i,j}(s)} = \frac{1}{s^r},\ i = 1,\ldots,N;\ j = 1,\ldots,m,$$

where $y_{i,j}$ and $\nu_{i,j}$ are output and the corresponding linear controller of the j^{th} channel of the i^{th} agent.

Denoting the linear controller block by $C_{i,j}(s)$, from the reference signal $r_{i,j}$ in (5) to the output $y_{i,j}$, the closed-loop transfer function of the reference model $A_{i,j}(s)$ is found to be

$$A_{i,j}(s) = \frac{C_{i,j}(s)G_{i,j}(s)}{1 + C_{i,j}(s)G_{i,j}(s)} = \frac{C_{i,j}(s)}{s^r + C_{i,j}(s)}. \tag{12}$$

Let us now choose the controller $C_{i,j}$ as

$$C_{i,j}(s) = c_{r-1}s^{r-1} + \ldots + c_1 s + c_0,$$

where the coefficients $c_k,\ k = 1,\ldots,r-1$, are chosen such that poles of the characteristic equation of the reference model lie in the left-half-plane. An observation about the controller (12) is that since the linearized plant model $G_{i,j}(s)$ contains rth order integrator, the controller consists of derivatives of the corresponding component of the error signal $e_{i,j}$ in (4) up to the degree of $r-1$ to satisfy stability condition. In other words, we have

$$\nu_{i,j} = c_{r-1}e_{i,j}^{(r-1)} + \ldots + c_1\dot{e}_{i,j} + c_0 e_{i,j}.$$

However, in most of the robotic applications relative degree of system models are assumed to be $r \leq 2$, that is equivalent to PD type controllers. Also note that, based on the certainty equivalence principle, in the design process of the linear controller (12) we assume the exact values of the plant parameters are used in feedback linearization. In other words, the certainty equivalence principle allows us to assume that the controller in (10) exactly linearizes the plant dynamics, and the estimation errors between α_i and β_i and the corresponding $\hat{\alpha}_i$ and $\hat{\beta}_i$ are neglected.

D. Sliding Mode Controller

As an alternative to model reference approach, we design a sliding mode controller which is known for decreasing error in a finite time or, in other words, it satisfies finite time convergence. Here we use a sliding controller of the form

$$\begin{aligned} \nu_{i,j} &= k_{i,j}\mathrm{sign}(\tilde{e}_{i,j}), \quad \tilde{e}_{i,j} = m_{i,j}^T\bar{e}_{i,j}, \qquad (13)\\ m_{i,j} &= [m_{i,j,r-1}\ \ldots\ m_{i,j,0}]^T, \quad \bar{e}_{i,j} = [e_{i,j}^{(r-1)}\ \ldots\ e_{i,j}]^T. \end{aligned}$$

where $k_{i,j} > 0$ are the controller gains. Choice of the terms of the vector $m_{i,j}$ is done so that the error dynamics is stable or basically the polynomials

$$T_{i,j}(s) = m_{i,j,r-1}s^{r-1} + \ldots + m_{i,j,1}s + m_{i,j,0}, \tag{14}$$

are Hurwitz for all i and j, [3].

V. Simulations and Results

Our simulation environment includes six agents each of whose dynamics are given by

$$\begin{aligned} \dot{p}_i &= v_i, \\ \dot{v}_i &= \alpha_i(t,\xi_i) + \beta_i(t,\xi_i)u_i, \end{aligned}$$

where $p_i = [x_i\ \ y_i]^T \in \mathbb{R}^2$ is the position vector and $v_i = [v_{i,x}\ \ v_{i,y}]^T \in \mathbb{R}^2$ is the velocity vector. As stated in

Assumption 2, the output vector is taken as the position vector, i.e., $y_i = p_i$. Having two-dimensional position and velocity vectors also means that the relative degree of each agent is $r = 2$. In simulations, we use the following values in place of the unknown functions α_i and β_i:

$$\alpha_i(t) = \begin{bmatrix} 0.1\cos(0.5t) \\ 0.2\cos(0.3t) \end{bmatrix}, \ \beta_i(t) = \begin{bmatrix} 0.8 & 0.03 \\ 0.5 & 1.3 \end{bmatrix}, \ i = 1, \ldots, 6.$$

Note that the chosen parameter values do not depend on the state ξ. However one may choose them as a function of the state as described in the general system model (2). Additionally, we choose the frequency of sinusoidal terms very small as indicated in section 3. The agents are initiated from the following points.

$$y_1(0) = \begin{bmatrix} 5 \\ 5 + \frac{1}{\sqrt{3}} \end{bmatrix}, \ y_2(0) = \begin{bmatrix} 4.75 \\ 5 + \frac{\sqrt{3}}{12} \end{bmatrix}, \ y_3(0) = \begin{bmatrix} 4.5 \\ 5 - \frac{\sqrt{3}}{6} \end{bmatrix},$$
$$y_4(0) = \begin{bmatrix} 5 \\ 5 - \frac{\sqrt{3}}{6} \end{bmatrix}, \ y_5(0) = \begin{bmatrix} 5.5 \\ 5 - \frac{\sqrt{3}}{6} \end{bmatrix}, \ y_6(0) = \begin{bmatrix} 5.25 \\ 5 + \frac{\sqrt{3}}{12} \end{bmatrix}.$$

With this choice, the agents constitute an equilateral triangle, and this initial formation is chosen as the desired formation. Then, the initial position of the virtual leader as well as the centroid of the formation becomes $y_v(0) = y_c(0) = [5 \ 5]^T$. For the virtual target we apply the reference path

$$y_v(t) = \begin{bmatrix} 5.05 + 0.3t - 0.05\cos(2t) \\ 8.8 - 3.8\cos(0.5t) \end{bmatrix}.$$

We first apply the model reference controller in place of the linear controller block in simulation. Since relative degree of each agent is $r = 2$, the linear controller C_{ij} simply becomes a PD controller such that

$$C_{ij}(s) = c_1 s + c_0, \quad i = 1, \ldots, 6, \ j = 1, 2.$$

We place the zeros of the characteristic equation at $s = -10$ such that

$$\Delta_{i,j}(s) = s^2 + C_{i,j}(s) = (s + 10)^2,$$

which implies that $c_1 = 20, \ c_0 = 100$. These choices of the parameters results in the response seen in Fig. 3. The virtual target tracks its reference path with small errors. Note also that the orientation of the formation does not change during the motion.

As the second linear controller, we employ the sliding mode controller with the parameter values of

$$k_{i,j} = 260, \ m_{i,j} = [0.002 \ 0.1]^T.$$

The signum function in (13) is replaced by a soft signum function, namely a saturation function, to avoid numerical problems in simulation. Fig. 4 shows the response of the system with this controller. As in the case of PD controller, it satisfies the tracking and formation maintain goals in a well manner. At the same time, the response of the agents is smoother than the response with PD controller at the first few seconds.

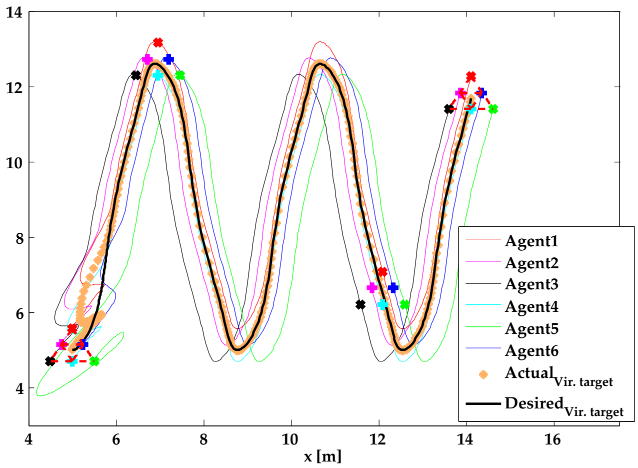

Fig. 3. Movement of a six-agent system along a sinusoidal path with PD controller

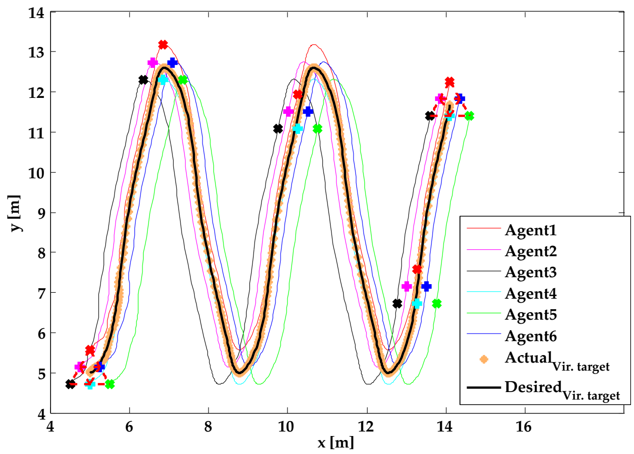

Fig. 4. Movement of a six-agent system along a sinusoidal path with sliding mode controller

Comparison of the distance values between the virtual target and its reference trajectory, which can also be used as performance indicator, is shown in Fig. 5. In that figure, it is clearly seen that the sliding mode controller outperforms the PD controller. Fig. 6 shows the parameter estimation performance of the least squares algorithm for one agent. Initial values of the estimated parameters are chosen as

$$\hat{\alpha}_i(0) = \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix}, \ \hat{\beta}_i(0) = \begin{bmatrix} 0.01 & 0.02 \\ 0.01 & 0.005 \end{bmatrix}, \ i = 1, \ldots, 6.$$

Although magnitude of the estimated parameter values is getting bigger at first times, they converge to constant values as stated in Proposition 4.1.

VI. Conclusion

In this article we have designed two adaptive control schemes for a multi-agent formation control problem under parametric uncertainty. The solution guarantees the stability of the estimated parameters when we treat the unknown functions in

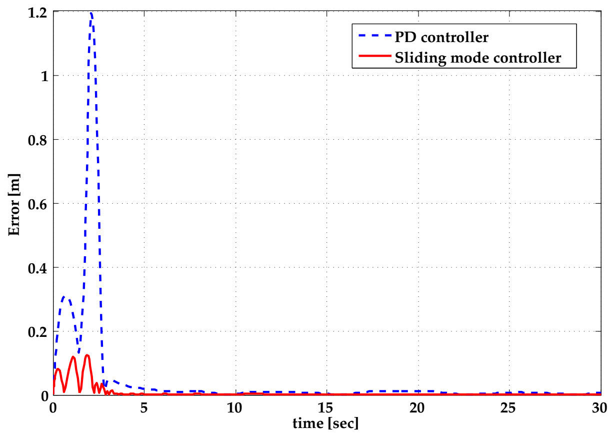

Fig. 5. Distance between virtual leader and its desired path

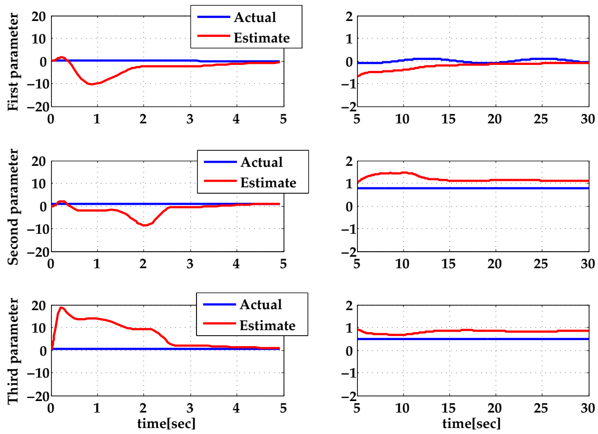

Fig. 6. Parameter estimation for agent 1

the agent dynamics as piecewise constants. With virtual target definition, the controller satisfies formation maintenance and reference tracking objectives. Even though it is not of interest in this paper, collision avoidance can be considered in the controller design easily using Assumption 3. Feedback linearization gives the advantage of design of a linear controller for the resulting linearized model, which stands for a simple PD controller in a two-dimensional robot formation example. We have also designed a sliding mode controller, which showed better performance in the performed simulations. Future research can focus on increasing the robustness properties of the developed algorithm and decreasing its sensitivity to additional disturbances and to sensing and approximation errors.

Acknowledgments

The work of S. Güler, N. Köksal, and B. Fidan is supported by the Canadian NSERC Discovery Grant 116806. N. Köksal holds a MEB (Turkish Ministry of National Education) scholarship. The work of V. Gazi was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant no 109E175.

References

[1] M. Arcak, “Passivity as a Design Tool for Group Coordination, ” *IEEE Trans. Automatic Control,* vol. 52, no. 8, pp. 1380-1390, August 2007.

[2] J.P. Desai, J.P. Ostrowski, and V. Kumar, “Modeling and control of formations of nonholonomic mobile robots, ” *IEEE Trans. Robotics and Automation,* vol. 17, no. 6, pp. 905-908, December 2001.

[3] S. Duran and V. Gazi, “Adaptive formation control and target tracking in a class of multi-agent systems, ” in *Proc. American Control Conference,* pp. 75-80, July 2010.

[4] R. Olfati-Saber and R. M. Murray, “Distributed cooperative control of multiple vehicle formations using structural potential functions,” in *Proc. IFAC World Congr.*, Barcelona, Spain, June 2002.

[5] B. Anderson, Yu Changbin, B. Fidan, and J. Hendrickx, “Rigid graph control architectures for autonomous formations,” *IEEE Control Systems,* vol.28, no.6, pp.48-63, December 2008.

[6] V. Gazi, “Formation control of a multi-agent system using nonlinear servomechanism,” *International Journal of Control*, vol. 78, no. 8, pp. 554–565, 20 May 2005.

[7] M.A. Lewis and K.H. Tan, “High Precision Formation Control of Mobile Robots Using Virtual Structures,” *Autonomous Robots,* vol. 4, no. 4, pp. 387-403, October 1997.

[8] H. Su, X. Wang, and Z. Lin, “Flocking of Multi-Agents With a Virtual Leader,” *IEEE Trans. Automatic Control,* vol. 54, no. 2, pp. 293-307, February 2009.

[9] V. Gazi and B. Fidan, “Coordination and control of multi-agent dynamic systems: Models and approaches,” in *Proceedings of the SAB06 Workshop on Swarm Robotics*, ser. Lecture Notes in Computer Science (LNCS) 4433, E. Sahin, W. M. Spears, and A. F. T. Winfield, Eds. Berlin Heidelberg: Springer-Verlag, 2007, pp. 71–102.

[10] V. Gazi and K. M. Passino, *Swarm Stability and Optimization*. Heidelberg, Germany: Springer-Verlag, January 2011.

[11] V. Gazi, B. Fidan, R. Ordóñez, and M.İ. Köksal, “A Target Tracking Approach for Non-Holonomic Agents Based on Artificial Potentials and Sliding Mode Control,” *Journal of Dynamic Systems, Measurement, and Control,* vol. 134, no. 6, November 2012.

[12] J. Yao, R. Ordóñez, and V. Gazi, “Swarm Tracking Using Artificial Potentials and Sliding Mode Control,” *Journal of Dynamic Systems, Measurement, and Control,* vol. 129, no. 5, pp. 749-754, September 2007.

[13] P.A. Ioannou and B. Fidan, *Adaptive Control Tutorial,* SIAM Society for Industrial & Applied Mathematics, 2006.

[14] J.T. Spooner and K.M. Passino, “Stable adaptive control using fuzzy systems and neural networks,” *IEEE Trans. Fuzzy Systems,* vol.4, no.3, pp.339-359, August 1996.

[15] P. Ogren, E. Fiorelli, and N.E. Leonard, “Cooperative control of mobile sensor networks:Adaptive gradient climbing in a distributed environment,” *IEEE Transactions on Automatic Control ,* vol.49, no.8, pp. 1292-1302, August 2004.

[16] H.K. Khalil, *Nonlinear Systems ,* Prentice Hall, Inc., 2002.

[17] E. Gül and V. Gazi, “Adaptive internal model based formation control of a class of multi-agent systems with switched exosystems,” in *Chinese Control and Decision Conference*, pp. 6–13, Taiyuan, China, May 2012.