Leader-following Formation Navigation with Virtual Trajectories for Dynamic Multi-agents*

Manabu SUZUKI†, Ryo KOBAYASHI‡, Kazushi NAKANO‡, Tetsuro FUNATO‡ and Yoshihiro MATSUI§

In recent years, cooperative control of multi-agents, especially formation control, has attracted considerable interest among control engineers. As one of the formation control methods, we deal with the leader-following formation navigation (LFFN). The LFFN is defined such that a human operator or an autonomous leader agent forms a queue of agents to guide multiple followers. The followers pass along the leader’s trajectory with the same motion as the leader. When the leader passes along a safe route to avoid obstacles, the followers also pass along a similar safe trajectory. However, when using the LFFN, the followers can shape only a single column formation.

In this study, we first extend the above LFFN’s usage to parallel single columns using adjoining virtual leaders. Next, we consider virtual leaders adjoining the leader to design virtual trajectories along which virtual leaders passed and virtual leader’s velocities on their path. We design the followers’ target points which track the virtual leader’s path with designed virtual velocity. Using our proposed method, the followers shape a formation which is not a single column but a multiple column type. Finally, we demonstrate the effectiveness of our newly proposed method through simulation and experimentation.

1. Introduction

In recent years, the cooperative control of multi-agents, especially formation control, has attracted considerable interest. Comparing to a single agent performing alone, multiple cooperative agents are expected to improve the efficiency of operation, flexibility and fault-tolerance of the system. In addition, the formation control of multi-agents can be applied to terrain data acquisition[1], search in disaster sites[2], the intelligent transport system (ITS)[3], formation flight of unmanned air vehicles (UAV)[4], satellite formation[5], etc.

The following methods for formation control have been proposed: behavior-based method[6]; game theoretic method[7]; virtual structure method [8,9]; and leader-following method [10,11]. In the behavior-based control, several desired behaviors defined by distances between agents are prescribed for each agent, and the control input is generated by the relative importance of each behavior. The game theoretic method uses the game theory to design a controller to maintain a formation. The virtual structure method assumes that agents composing a formation are connected by a rigid body, and controls these agents to maintain the formation. The leader-following method assigns one agent to a leader and assigns other agents to followers which track the leader with a formation. The important idea of these methods is that the agents move while maintaining the distances between agents in order to achieve a specified formation shape.

On the other hand, as one of formation control methods, we focus on the leader-following formation navigation (LFFN) as shown in Fig. 1. The LFFN is based on the idea that a human operator or an autonomous leader agent forms a queue to guide other follower agents. Unlike other methods, the LFFN shapes a formation by forcing the followers to track the leader’s trajectory in the same motion as the leader.

Fig. 1 Leader-following formation navigation
after a suitable time. When the leader passes along a safe route to avoid obstacles, the followers also pass along a similar safe trajectory.

In order to apply the LFFN method to a real environment, we have proposed methods of collision avoidance based on leaving the leader’s path or adjusting delay times of followers from the leader[12], achievement of efficient movements of followers around the initial time[13] and maintenance of formation-distance[14]. Finally, a simulation with ten formation by combining the method for maintaining formation-distance[14]. which move with the virtual velocities in parallel, the trajectory. When the followers reach the target points of the followers which track the virtual path and velocity as a desired motion to the followers away from the leader, and regard the virtual leader’s a virtual leader which locates at a certain distance by using virtual trajectories in parallel. We consider a large number of agents, it takes much time to move all the agents. Moreover, when a group of agents is composed of a large number of agents, it takes much time to move all the agents.

This paper discusses an extension of the LFFN by using virtual trajectories in parallel. We consider a virtual leader which locates at a certain distance away from the leader, and regard the virtual leader’s path and velocity as a desired motion to the followers in Fig. 2. The virtual trajectory and velocity can be designed by online geometric calculation. We design target points of the followers which track the virtual trajectory. When the followers reach the target points which move with the virtual velocities in parallel, the LFFN has been achieved for the virtual leaders. The proposed method makes it possible to shape a formation with multiple columns and to demand a complex formation by combining the method for maintaining formation-distance[14]. Finally, a simulation with ten vehicle robots and experimentation with six vehicle robots demonstrates the effectiveness of our proposed method.

2. Leader-following Formation Navigation

This paper considers a leader-following formation navigation (LFFN) problem. There are \((n+1)\) agents in two dimensional space consisting of one leader and \(n\) followers. The leader is numbered as \(i=0\). The followers are adequately numbered as the \(i\)-th follower \(i \in \mathbb{N}, \mathbb{N} = \{1, 2, \cdots, n\}\). The leader is controlled by a human operator, and lets the leader’s position be \(q_0(t) \in \mathbb{R}^2\). In using usual cars, it is necessary to take a constraint condition into consideration. In this study, however, we use omni-directional driven cars which do not have a constraint condition as a preceding paragraph consideration. Each follower is controlled by a suitable local controller, and is moved according to the second-order differential equation

\[
\ddot{q}_i(t) = u_i(t).
\]

The variables \(q_i(t), u_i(t) \in \mathbb{R}^2\) represent the coordinate and input of follower \(i\), respectively, and the follower’s motion is determined by the initial state \((q_i(0), \dot{q}_i(0))\) and input \(u_i(t)\). Note that we assume that the initial time is 0 without loss of generality.

First, we formulate the leader’s path and velocity. We transform the leader’s position \(q_0(t)\), a time-varying function, into an arc-length parameterized function. Let an arc-length \(s(t)\) be

\[
s(t) = \int_0^t \|q_0(\sigma)\| \, d\sigma,
\]

and the inverse function of Eq. (2) be \(t(s)\). Then, the leader’s path can be represented as \(\phi_L(s) = q_0(t(s))\), where \(\phi_L : \mathbb{R}_+ \rightarrow \mathbb{R}^2\) is an arc-length parameterized function. Similarly, the leader’s velocity is described as \(v_0(s) = \|\dot{q}_0(t)\|_{t=s}\) at each point \(s\) using a function \(v_0 : \mathbb{R}_+ \rightarrow \mathbb{R}_+\). As shown in Fig. 3, the leader’s path is parameterized by \(s\), and the leader’s velocity at the point \(s\) is given by \(v_0(s)\). The leader robot moves in accordance with a motion equation, and so \(\phi_L\) and \(v_0\) are generated from leader’s position \(q_0(t)\). We can assume that \(\phi_L\) is of class \(C^2\), and that \(v_0\) is of class \(C^1\) because these assumptions are necessary conditions for using \(\phi_L\) and \(v_0\) in calculations.

We expect that the followers can track the leader’s path with the same velocities behind the leader, which is called “leader-following formation navigation” (LFFN). Each follower is expected to track the path \(\phi_L\) in a correct order, and the velocity of the follower at each point \(s\) is expected to be \(v_0(s)\), as shown in Fig. 3. Then, we can define the geometric task such that the position \(q_i(t)\) of follower \(i\) moves toward the reference path \(\phi_L\). Next, we define the dynamic task such that the follower \(i\) moves with an assigned velocity \(v_0(s)\) along the path \(\phi_L\) in the positive direction of \(s\). When the parameterized arc-length point where the follower \(i\) passes at time \(t\) on the leader’s path is represented as \(s_i(t)\), these tasks for the LFFN are formulated as

![Fig. 2 Extension by virtual trajectory](image)

![Fig. 3 Reference path and assigned velocity](image)
\[
\lim_{t \to \infty} \|q_i(t) - \phi_L(s_i(t))\| = 0, \quad (3)
\]
\[
\lim_{t \to \infty} |s_i(t) - v_0(s_i(t))| = 0. \quad (4)
\]

The variable \( s_i(t) \) is called “trajectory parameter”, which is assumed to be an increasing function.

Since the followers are behind the leader, their trajectory parameters \( s_i(t) \) have to be less than the leader’s one \( s_0(t) \). We request the followers to move on behind another in order of \( i \), that is
\[
s_0(t) > s_1(t) > s_2(t) > \cdots > s_n(t), \forall t \geq 0. \quad (5)
\]

Moreover, we have to avoid collisions between agents. Let the agent \( i \)'s radius be \( r_i \) and the collision avoidance task be defined as
\[
||q_i(t) - q_j(t)|| > r_i + r_j, \quad (i, j) \in \mathcal{L}, \forall t \geq 0, \quad (6)
\]
where \( \mathcal{L} = \{(i, j) : 0 \leq i < j \} \) is the set of the agent’s pairs.

Our objective for the followers is to realize the collision-free LFFN Eqs. (3)-(6). We use the controller employed by [13], which consists of two parts to achieve a collision-free LFFN. Note that this controller has achieved Eqs. (3)-(6) in the previous study [13].

The first part is for the input \( u_i \) for the geometric task (3):
\[
u_i(t) = \ddot{\phi}_L(s_i(t)) - k_{q_i}(\dot{q}_i(t) - \dot{\phi}_L(s_i(t))) - k_{s_i}(\ddot{s}_i(t) - \ddot{\phi}_L(s_i(t))), \quad (7)
\]
where the design parameters \( k_{q_i}, k_{s_i} \in \mathbb{R} \) are positive constants. This equation consists of acceleration input, position feedback and velocity feedback.

The second part is for the parameter adjustment for the dynamic task (4) and the collision avoidance task (6):
\[
s_i(t) = \frac{d\nu_i(s_i(t))}{ds_i} s_i(t) - k_s(v_0(s_i(t)) - s_i(t)) + \frac{dF(s_i(t))}{ds_i}, \quad (8)
\]
\[
F(s_i(t)) = \sum_{j \neq i} f_{ij}(||\phi_L(s_i(t)) - \phi_L(s_j(t))||), \quad (9)
\]
where the design parameter \( k_s \in \mathbb{R} \) is a positive constant and \( f_{ij} \) is a non-negative decreasing function. Equation (8) consists of acceleration input and velocity feedback terms and a term for collision avoidance.

Assume that the initial values of the trajectory parameter \( s_i(0) \) fulfill the order condition (5) at time 0
\[
s_0(0) > s_1(0) > s_2(0) > \cdots > s_n(0). \quad (10)
\]

Note that the leader’s path \( \phi_L(s) \) is defined for the domain \( \mathbb{R}_+ \), and is not defined for \( s < 0 \), since the leader moves from the initial position \( \phi_L(0) \) in the positive direction of \( s \). This problem is solved by our previous method [13].

The LFFN method does not consider any type of formation shape because a formation made by the LFFN method is only of column type. Moreover, when an agent group is composed of a large number of agents, it takes much time to move that group. In order to solve the above problem, we consider virtual leaders which are at certain distances away from the leader, and regard one’s paths and velocities as desired motions to the followers. We will present a concrete design procedure in the next section.

### 3. Extension of LFFN with Virtual Trajectory

This section considers an extension of the leader-following formation navigation (LFFN) by using \( m \) virtual trajectories. This paper designs virtual trajectories and velocities in order to track the follower by geometric calculation. We define that the leader’s direction at point \( s \) on \( \phi_L \) is \( \angle \phi_L(s - ds)ds \). Note that a prime “~” is a differential with respect to an arc-length \( s \), \( \frac{d}{ds} \) and \( \angle \phi_L \) is differentiated \( \angle \phi_L \) with respect to arc-length \( s \). As an example, we consider the case of \( 2 \) virtual agents. These agents are called “virtual leader”, and the virtual leader’s direction is defined as the same as the leader’s one. The virtual leader is adequately numbered as the \( k \)-th virtual leader. A virtual leader always presents at the distance \( l_k > 2r_0 \) (in the direction \( \dot{\theta}_k \in (-\pi/2, \pi/2)(k = 1, 2, \cdots, m) \) apart from the front of the leader. When the leader has moved, the virtual leaders also have moved together. The path \( \dot{\phi}_k \) along which the virtual leader passed is called “virtual trajectory”.

As shown in Fig. 4, the virtual trajectory \( \hat{\phi}_k \) is given by geometric calculation with the leader trajectory \( \phi_L(s) \) as follows:
\[
\hat{\phi}_k(s) = \phi_L(s + \alpha_k(s)) + l_k \left( \frac{\sin(\dot{\theta}_k)\cos(\angle \phi_L(s - ds + \alpha_k(s - ds))ds)}{\angle \phi_L(s - ds + \alpha_k(s - ds))ds} \right),
\]
where \( \alpha_k \) is the difference between the arc-length of the leader and that of virtual leader. Note that the leader’s trajectory parameter may be different from the virtual leader’s one because virtual leader’s position is defined at the distance \( l_k \) (in the direction \( \dot{\theta}_k \)) apart from the front of the leader. Now, \( \alpha_k \) is given by
\[
\alpha_k(s) = l_k \int_0^s \sin(\angle \dot{\phi}_k(s))dt. \quad (12)
\]

Equation (12) shows that when the follower moves only a fixed distance \( s_0 \) along the virtual trajectory \( \dot{\phi}_k \), the leader moves a distance \( s_0 = s_0 + \alpha_k(s_0) \) along the trajectory \( \phi_L \).

As shown in Fig. 5, the velocity \( \dot{\phi}_k \) on the virtual trajectory is called “virtual leader’s velocity”, and is given by
\[ \ddot{v}_k(s) = v_0(s + \alpha_k(s)) - l_k \sin(\phi_L(s + \alpha_k(s)))ds. \] (13)

Equation (13) shows that the velocity of the virtual leader which is presented outside of the leader is faster than their velocities when the leader turns. Similarly, the velocity of the virtual leader which is presented outside of that, is slower than the one’s velocity. At that time, the virtual velocity might be negative due to the leader’s quick turn. For this reason, by using the geometrical relationships, the leader’s turning \( \phi_L \) at point \( s \) on the trajectory is limited as follows:

\[ \dot{\phi}_L(s) < \sin^{-1}\left( \frac{v_0(s)}{l_k} \right) \frac{1}{ds}. \] (14)

Equation (14) shows that the leader cannot make a quick turn when the distance \( l_k \) between the leader and the virtual leaders is long or the leader’s velocity is slow.

Thus, if each follower tracks the trajectory parameter \( s_1(t) \) moved with \( \dot{v}_k(s_1(t)) \) on the virtual trajectory \( \phi_L \), they satisfy the control objective Eqs. (3) and (4) which are rewritten as \( \phi_L \) and \( \dot{v}_k \) from \( \phi_L \) and \( v_0 \), respectively:

\[ \lim_{t \to \infty} \| q_i(t) - \phi_L (s_1(t)) \| = 0, \] (15)
\[ \lim_{t \to \infty} | s_i(t) - \phi_L (s_1(t)) | = 0. \] (16)

To achieve the control objective Eqs. (15) and (16), the controller are rewritten as follows:

\[ u_i(t) = \dot{\phi}_L (s_1(t)) - k_{vi}(q_i(t) - \phi_L (s_1(t))) \]
\[ - k_{qi}(\dot{q}_i(t) - \phi_L (s_1(t))), \] (17)

\[ s_1(t) = \frac{d\phi_L(s_1(t))}{ds_1} s_1(t) - k_{vi}(\dot{q}_i(t) - \phi_L(s_1(t))) + \frac{dF(s_1(t))}{ds_1}, \] (18)

\[ F(s_1(t)) = \sum_{j \neq i} f_{ij} (| \dot{\phi}_L (s_1(t)) - \phi_L (s_1(t)) |), \] (19)

where the design parameter \( k_{qi}, k_{vi} \in \mathbb{R} \) and \( k_i \in \mathbb{R} \) are positive constant and \( f_{ij} \) is a non-negative decreasing function.

Comparing to other formation shaping methods[9] and[11], the proposed method can prevent the follower from crossing other trajectories, and can realize collision avoidance and maintenance of formation-distance with the parameter adjustment.

4. Simulations

This section, we simulate the leader-following formation navigation (LFFN) with ten vehicle robots \( n = 9 \) in a flat plane using our proposed method. The dynamic model of the vehicle robot is governed by the second-order differential equation (1).

We consider an extension of the LFFN by using two virtual trajectories \( n = 2, k \in \{0, 1, 2\} \). The virtual leaders’ distances are \( l_1 = 0.25, l_2 = 0.2 \) and \( \theta_1 = \pi/2, \theta_2 = -\pi/2 \). The initial positions of the vehicle robots are \( q_0(0) = [0, -0.5], q_1(0) = [-0.25i, -0.5][i = 1, 2, 3], q_2(0) = [-0.25(i - 3), -0.25(i - 2)](i = 4, 5, 6) \) and \( q_i(0) = [-0.25i - 6, -0.7](i = 7, 8, 9) \), respectively. Also, we let the initial direction of the leader be \( \theta_0 = 0 \) in the global information. Then robots are placed, such that \( s_1, s_2 \) and \( s_3 \) are on the trajectory \( \phi_L, s_4, s_5 \) and \( s_6 \) are on actual trajectory \( \hat{\phi}_1 \), and \( s_7, s_8 \) and \( s_9 \) are on actual trajectory \( \hat{\phi}_2 \), respectively. For this reason, \( \hat{\phi}_1, \hat{\phi}_2 \in \mathbb{N} \), \( N_k = \{1 + 3k, 2 + 3k, 3 + 3k\} \). The initial trajectory parameters are given as \( s_0(0) = 0, s_1(0) = s_4(0) = s_2(0) = s_3(0) = -0.25, s_7(0) = s_6(0) = -0.5 \) and \( s_8(0) = s_9(0) = -0.75 \), respectively. The leader path and the leader’s velocity are given by

\[ \phi_L(s) = \begin{cases} [s, -0.5]^T, & s < 0 \\ [s, -0.5]^T, & s \in [0, r_\phi] \\ [r_\phi (2\sin(w_1) + 1) - 2r_\phi \cos(w_1) - 0.1]^T, & s \in [r_\phi, s_0] \\ [0.6, s - (0.3 + 2r_\phi \pi)]^T, & s \in [s_0, s_1] \\ [r_\phi (2\cos(w_2) + 1) + 2r_\phi \sin(w_2) + 0.1]^T, & s \in [s_1, 2s_0] \\ [r_\phi - (s - 2r_\phi (1 + 2\pi)), 0.5]^T, & s \in [2s_0, \infty] \end{cases} \]

\[ r_\phi = 0.2, \quad s_0 = r_\phi (1 + 2\pi), \quad s_1 = s_0 + r_\phi \]

\[ w_1 = \frac{s - r_\phi}{4r_\phi}, \quad w_2 = \frac{s - (s_0 + r_\phi)}{4r_\phi}, \]

\[ v_0(s) = \]
virtual leaders are depicted by the circle, the followers
Similarly, the velocities of the trajectory parameters
Let the design parameters be \( k_i = 2 \) and the weight
Furthermore, we use the parameter adjustments Eq. (8)
In Eq. (20), when \( i = 4 \) or 6, \( s_{i-1} \) is rewritten as \( s_{(0.5i-1)} \).
The first virtual leader’s velocity \( \hat{v}_1 \) is slower than the leader’s
velocity \( v_0 \) after \( s = 0.2 \) because his trajectory is inside
the leader’s trajectory, as in Fig. 6. In contrast,
the second virtual leader’s velocity \( \hat{v}_2 \) is faster than leader’s velocity \( v_0 \) after \( s = 0.2 \) because his trajectory is outside
the leader’s trajectory as in Fig. 6. Thus, we see that according to the proposed method (13), the two virtual leaders have adapted their velocity to
the velocity of the leader.
Next, the results of the velocities tracking of the trajectory parameter \( s_i(t) \) are shown in Fig. 8. Tracking of target velocities of the trajectory parameters
\( v_0, \hat{v}_1 \) and \( \hat{v}_2 \) is shown in the upper, middle and lower figures, respectively. The velocities of the trajectory parameters \( \hat{s}_i(t), i = 0, 1, 2, 3 \) are done by the solid, broken, dashed-dotted and chain lines, respectively.
Similarly, the velocities of the trajectory parameters \( \hat{s}_k \) and \( \hat{s}_i(t), i = 4, 5, 6 \) (i.e., 7, 8, 9) are depicted by the
solid, broken, dashed-dotted and chain lines, respectively. We confirm that the velocity of the trajectory parameter \( \hat{s}_i(t) \) decelerates to avoid a collision and tracks each target velocity \( v_0, \hat{v}_1 \) and \( \hat{v}_2 \).
Finally, the paths of the leader and the followers
1, 2, \ldots, 9 are shown in Fig. 9. The leader and the virtual leaders are depicted by the circle, the followers

\[
\begin{align*}
0.03 \sin \left( \frac{50\pi(s + 0.27)}{9} \right) + 0.04, & \quad s \in [0, r_v) \\
0.02 \sin \left( \pi(s + (r_v \pi - 0.17)) \right) + 0.05, & \quad s \in [r_v, s_2) \\
0.02 \sin((s - (2r_v \pi - 0.1))5\pi) + 0.05, & \quad s \in [s_2, s_2 + r_v) \\
0.01 \sin((s - r_v(\pi + 2))2.5) + 0.06, & \quad s \in [s_2 + r_v, 2s_2) \\
0.01 \sin((s - (4r_v \pi - 0.1))\pi) + 0.04, & \quad s \in [2s_2, \infty) \\
\end{align*}
\]

Finally, we consider the velocity \( v_0 \) (solid line) of the leader and the velocities \( \hat{v}_1 \) (broken line) and \( \hat{v}_2 \) (dash-dotted line) of two virtual leaders, which are derived from \( v_0 \) by the proposed method (13). The first virtual leader’s velocity \( \hat{v}_1 \) is slower than the leader’s velocity \( v_0 \) after \( s = 0.2 \) because his trajectory is inside the leader’s trajectory, as in Fig. 6. In contrast, the second virtual leader’s velocity \( \hat{v}_2 \) is faster than leader’s velocity \( v_0 \) after \( s = 0.2 \) because his trajectory is outside the leader’s trajectory as in Fig. 6. Thus, we see that according to the proposed method (13), the two virtual leaders have adapted their velocity to the velocity of the leader.

Finally, the paths of the leader and the followers
1, 2, \ldots, 9 are shown in Fig. 9. The leader and the virtual leaders are depicted by the circle, the followers

\[
\begin{align*}
F(s_i(t)) &= f_{ij}(s_i(t) - s_{i-1}(t)), \quad i \in \mathbb{N}_k. 
\end{align*}
\]
1, 4, 7 are depicted by the square, the followers 2, 5, 8 are depicted by the rhombus and the followers 3, 6, 9 are depicted by the asterisk, respectively. We confirm that the followers have moved from the bottom left to the top left along each target path. The followers’ final formation is different from their initial formation because they decelerate by a term for collision avoidance in the parameter adjustment (8) and (18).

Therefore, our proposed method realizes an extension of the LFFN by using virtual trajectories.

5. Experimentation

This section, we demonstrate the effectiveness of our proposed method through experimentation. We show the experimental environment in Fig. 10. Experimental field is square of one side 1.83m, six robots (e-pack), and camera (IMI-TECHIMC-15FT) to detect the field and the robots. The robot’s diameter, height and shape are 0.1m, 0.05m, and cylinder. One robot is a leader, and the others are followers. Information from the camera is calculated on PC. After that information concerning the leader and followers’ local positions is sent to the robots with wireless LAN.

Each of the robots has a local controller and his velocity is controlled with Eq. (1). We use Eq. (7) as the upper level controller. The virtual leaders’ distances are \( l_1 = 0.25, l_2 = 0.2 \) and \( \theta_1 = \pi/2, \theta_2 = -\pi/2 \). The initial positions of the vehicle robots are \( q_0(0) = [0, -0.5], q_i(0) = [-0.25i, -0.5] (i = 1), q_i(0) = [-0.25(i-1), -0.25] (i = 2, 3) \) and \( q_i(0) = [-0.25(i-3), -0.7] (i = 4, 5) \), respectively. Also, the initial direction of the leader is set to be \( \theta_0 = 0 \) in the global coordination. Then robots are placed, such that \( s_1 \) is on the trajectory \( \phi_L \), \( s_2 \) and \( s_3 \) are on actual trajectory \( \phi_1 \), and \( s_4 \) and \( s_5 \) are on actual trajectory \( \phi_2 \), respectively. The initial trajectory parameters are given as \( s_0(0) = 0, s_1(0) = s_2(0) = s_4(0) = -0.25, \) and \( s_3(0) = s_5(0) = -0.5 \) respectively.

The leader path, the leader’s velocity and parameters have the same values as simulation. Furthermore, we use the parameter adjustments Eq. (8) and (18). \( s_0, s_1 \) and \( s_2 \) are the trajectory parameters of the leader and the virtual leaders 1 and 2.
Figure 11 shows the actual trajectory $\phi_0$ of the leader (solid line) and two virtual trajectories $\phi_1, \phi_2$ of two virtual leaders (broken and dash-dotted lines), which are designed by the proposed method Eq. (11). We can see that the virtual trajectories $\phi_1, \phi_2$ are generated with $\theta_1 = \pi/2$ and $\theta_2 = -\pi/2$ according to Eq. (11) along the trajectory of the leader.

In Fig. 12 we consider the velocity $v_0$ (solid line) of the leader and the velocities $v_1$ (broken line) and $v_2$ (dash-dotted line) of two virtual leaders, which are derived from $v_0$ by the proposed method (13). The first virtual leader’s velocity $v_1$ is slower than the leader’s velocity $v_0$ after $s=0.2$ because his trajectory is inside the leader’s trajectory, as in Fig. 11. In contrast, the second virtual leader’s velocity $v_2$ is faster than the leader’s velocity $v_0$ after $s=0.2$ because his trajectory is outside the leader’s trajectory as in Fig. 11. Thus, we see that according to the proposed method (13), the two virtual leaders have adapted their velocities to the velocity of the leader.

Next, the results of the velocity tracking of the trajectory parameter $s_i(t)$ are shown in Fig. 13. Tracking of target velocities of the trajectory parameters $v_0$, $v_1$ and $v_2$ are shown in the upper, middle and lower figures. The velocity of the trajectory parameter $s_i(t)$, $i=0,1$ are depicted by the solid and broken lines, respectively. Similarly, the velocities of the trajectory parameters $s_i(t)$, $i=2,3$ are depicted by the solid, broken, and dashed-dotted lines, respectively. We confirm that the velocity of the trajectory parameter $s_i(t)$ decelerates to avoid a collision and tracks each target velocity $v_0$, $v_1$ and $v_2$.

Finally, the paths of the leader and the followers 1, 2, $\cdots$, 5 are shown in Fig. 14. The leader is depicted by the circle, the followers 1, 2, 4 are depicted by the square, and the followers 3, 5 are depicted by the rhombus, respectively. We confirm that the followers have moved from the bottom left to the top left along each target path. The followers’ final formation is different from their initial formation because of the same reason in the simulation.

Therefore, our proposed method realizes an extension of the LFFN by using virtual trajectories. From this experimental results, accurate trajectories have been achieved, and the effectiveness of this method is confirmed.

6. Conclusion

This paper considered to extend the leader-following formation navigation (LFFN) by using the virtual trajectories in parallel. We considered a virtual leader which is located at a certain distance away from the leader, and regarded the virtual leader’s path and velocity as a desired motion to the followers in Fig. 2. The virtual trajectory and velocity were designed by online geometric calculation. We designed target points of the followers which tracked the virtual trajectories. When the followers reached the target points which moved with the virtual velocities, the LFFN had also been achieved for the virtual leaders. The proposed method made it possible to shape a formation with multiple columns and also to demand a more complex formation by combining the method for maintaining formation - distance[14]. Finally, simulations with ten vehicles and experimentation with six vehicles were demonstrated to show the effectiveness of our proposed method. In the future, we will focus on an application of the LFFN to vehicles with non-holonomic constraints.

References

[1] L. E. Navarro-Serment, R. Grabowski, C. J. J. Paredis and P. K. Khosla: Modularity in small distributed robots; the SPIE Conference on Sensor Fusion and Decentralized Control in Robotic System II, Vol. 3839, pp. 297–306 (1999)
[2] T. Nakatsuka, K. Sakurama and K. Nakano: Teleportation system for multiple mobile robots with autonomous collision avoidance; 36th SICE Symposium on Intelligent Systems, Vol. 36, pp. 97–100 (2009) (in Japanese)
[3] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui and H. Fujii: Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications; IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 3, pp. 155–161 (2002)
[4] R. L. Raffard, C. J. Tomlin and S. P. Boyd: Distributed optimization for cooperative agents: application to formation flight; the 43rd IEEE Conference on Decision and Control, pp. 2453–2459 (2004)
[5] S. D. Amico and O. Montenbruck: Proximity operations of formation-flying spacecraft using an eccentricity/inclination vector separation; Journal of Guidance, Control, and Dynamics, Vol. 29, No. 3, pp. 554–563 (2006)
[6] J. R. T. Lawton, R. W. Beard and B. J. Young: A decentralized approach to formation maneuvers; IEEE Transactions on Robotics and Automation, Vol. 19, No. 6, pp. 933–941 (2003)
[7] M. M. Korjani, A. Afkhar, M. Norouzitallab and M. B. Menhaj: Dynamic autonomous agent positioning based on computational intelligence; the International Multi Conference of Engineers and Computer Scientists 2009, pp. 150–155 (2009)
[8] J. P. Desai, J. Ostowski and V. Kumar: Controlling formations of multiple mobile robots; the 1998 IEEE International Conference on Robotics & Automation, pp. 2864–2869 (1998)
[9] Q. Li and Z. P. Jiang: Pattern preserving path following of unicycle teams with communication delays; Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, pp. 3051–3056 (2009)
[10] R. Fierro, P. Song, A. Das and V. Kumar: Cooperative control of robot formations; Cooperative Control and Optimization, R. Murphey and P. Pardalos (Eds.), Applied Optimization, Vol. 66, Ch. 5, pp. 73–93, Kluwer Academic Publishers (2002)
[11] X. Chen, Y. Jia, J. Du and F. Yu: Column formation control of multi-robot systems with input constraints; 50th IEEE Conference on Decision and Control and
Kazushi Nakano (Member)

He received the Dr. Eng. degree from Kyushu University, Fukuoka, Japan, in 1982. From 1980, he was a Research Associate at Kyushu University. From 1986, he was an Associate Professor at Fukuoka Institute of Technology. From 1999, he was a Professor in the Graduate School of Informatics and Engineering, the University of Electro-Communications (UEC), Tokyo, Japan. He is currently a Member of the Board of Directors and a Vice-president of UEC. His interests include system identification/control and their applications. He is a member of IEEE, SICE, IEEJ and ISCIE.

Tetsuro Funato

He received the B.E., M.E., and Ph.D. degrees from the Department of Mechanical and Control Engineering, Tokyo Institute of Technology, Tokyo, Japan, in 2003, 2005, and 2008, respectively. From 2008 to 2012, he was a postdoctoral researcher in the Department of Mechanical Engineering, Kyoto University. In 2012, he was a Research Fellow of CREST of Japan Science and Technology Agency (JST). Since 2013, he has been an Assistant Professor in the Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Tokyo, Japan. His research interests include neural control of locomotion and system biomechanics.

Yoshihiro Matsui (Member)

He received the B.E., M.E. and Ph.D. degrees all in Electronic Engineering from the University of Electro-Communications, Tokyo, Japan, in 1985, 1987 and 1999, respectively. Since 1987, he had been working for Matsushita Electric Co., Ltd. for 6.5 years. He is now a professor in the Department of Electrical Engineering at the National Institute of Technology, Tokyo College. His current research interests include control system design and analysis. He is a member of SICE, IEEJ and ISCIE.