Instant Orientation Correction Strategy: Holonomic System-Based Decentralized Control for Nonholonomic Agents

CHANYEONG JEONG1,∗, EUGENE KIM1,∗, MYEONG-HWAN HWANG1, HYO-SUNG AHN2, (Senior Member, IEEE), AND HYUN-ROK CHA1

1Automotive Materials & Components Research and Development Group, Korea Institute of Industrial Technology, Gwangju 61012, South Korea
2Distributed Control and Autonomous System Laboratory, School of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Hyun-Rok Cha (hrcha@kitech.re.kr)

∗Chanyeong Jeong and Eugene Kim are co-first authors.

This work was supported by the Korea Institute of Industrial Technology through the Development of Core Technologies for a Working Partner Robot in the Manufacturing Field under Grant KITECH EO-22-0009.

ABSTRACT This study proposes an instant orientation correction strategy to perform an orientation alignment-based formation control applicable to nonholonomic agents. Generally, an orientation alignment-based formation control cannot be applied to nonholonomic agents with orientation constraints because it is originally a protocol for holonomic agents. Therefore, we improved the orientation alignment-based formation control using an instant orientation correction strategy that modifies the local frame of the nonholonomic agents. To examine the effect of the proposed method, simulations and experiments using representative parameters were conducted for unicycle-modeled nonholonomic agents. As a result, nonholonomic agents could execute a holonomic system-based control law by applying the proposed correction strategy. The simulation confirmed the change in the convergence time and the cumulative error of the angular velocity according to the number of agents and the updated step size of the local desired position. As the updated step size increased, the cumulative error of the angular velocity tended to increase (0.3 rad/s∼0.4rad/s). Also, the convergence time was not dramatically affected by the update step size and the number of the agents (∼400 converge step).

INDEX TERMS Holonomic system, decentralized control, nonholonomic agent, formation control, unicycle model.

I. INTRODUCTION In sub-research fields of decentralized controls, such as consensus theory and formation control, numerous control laws for holonomic systems have been reported [1], [2], [3], [4]. Nevertheless, decentralized control should be applied to nonholonomic systems having constraints in movement, such as vehicles. Notably, several studies have been reported wherein a decentralized control is initially developed for nonholonomic systems, or an algorithm for a holonomic system-based control law is extended by limiting certain conditions [5], [6], [7], [8], [9], [10], [11], [12]. However, existing studies on holonomic systems cannot be directly applied to nonholonomic systems. Therefore, the motivation of this study aims to incorporate control laws targeting holonomic systems into nonholonomic systems.

When implementing holonomic system-based control laws, specially designed robots such as multicopters and omnidirectional-wheeled mobile robots are preferred. Typically, nonholonomic systems are more widely used than holonomic systems, and studies on nonholonomic systems are more practical. Although developing a certain control law by considering nonholonomic systems is more convenient, extending all control laws to various models for nonholonomic systems is time-consuming and inefficient. Therefore, this study develops a strategy that allows nonholonomic systems to use control laws based on holonomic systems. When a nonholonomic system executes a control law targeting

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Li.
holonomic systems, the system cannot produce the desired output owing to the constraint of the moving direction, causing a phenomenon in which an inappropriate value is fed back. In this situation, by modifying the local frame-dependent variables in the original control law, the nonholonomic system can execute the holonomic-system-based control law without the divergence of the control input.

A. RELATED WORKS
Several studies on decentralized control have been reported [1], [2], [13], [14], [15], [16]. Control laws, when local variables and the shape of formation are given, can be classified into displacement-, distance-, and angle-based controls [1]. When implementing distance-based [17], [18], [19], [20], [21] and angle-based control laws [22], [23], [24], decentralized controllers are not affected by the orientation angle of the local frame. Control laws in which the states of controllers depend on the local frame generally assume that the orientation angle of the local frame is identical to that of the global frame [25]. Moreover, the orientation angles of other agents are compared with those of neighboring agents [26], [27], or the velocity constraints of vehicles are monitored [28]. In studies considering both orientation and nonholonomic constraint simultaneously, methods that periodically change the angular velocity of agents [5] or converge the positions and the orientations of the agents to locally given formation with orientations have been employed [11].

B. CONTRIBUTIONS
In this study, we propose an instant orientation correction strategy (IOCS) satisfying nonholonomic constraints by correcting the reference frame of each agent. Because of nonholonomic constraints, numerous decentralized control approaches are not physically feasible when directly applied to nonholonomic systems. Therefore, the proposed IOCS corrects local variables in a local frame used in a holonomic system-based control law considering the difference in the orientation angles between the local frames of the nonholonomic agent and intermediate state.

II. METHODOLOGY
For a state space vector \( x \in \mathbb{R}^n \), the elements of the vector \( x \) are denoted by \( x_1, x_2, \ldots, x_n \in \mathbb{R} \), and the Euclidean norm of \( x \) is denoted by \( ||x|| \). For the vectors \( x_1, x_2, \ldots, x_n \in \mathbb{R}^d \), an integrated vector is denoted by \( [x_1^T, \ldots, x_n^T]^T \in \mathbb{R}^{dn \times 1} \). For the angle element \( \theta_i \), a counter-clockwise rotation matrix is denoted by \( R_\theta \in \mathbb{R}^{d \times d} \). For a variable \( x \), the estimation and the time derivative of \( x \) are denoted by \( \hat{x} \), and \( \dot{x} \), respectively.

A. GRAPH THEORY
The basic graph theory is introduced in [29]. Interactive network can be represented using the graph \( G = (\mathcal{V}, E) \), where \( \mathcal{V} = 1, 2, \ldots, n \) is the set of \( n \) nodes, and \( E \subseteq \mathcal{V} \times \mathcal{V} \) is the edges connecting the nodes. Especially, a set of node \( j \) is connected to a certain node \( i \) through edges are referred to as neighboring nodes \( \mathcal{N}_i = \{ j \in \mathcal{V} : (i,j) \in E \} \), where \((i,j)\) is an edge from \( i \) to \( j \). In this study, we ignore the case of a self-loop \((i,i)\). The direction of an edge is indicated by an arrow and denotes the state observation between nodes, that is, the node located at the tail of the arrow observes the state of that located at the head. A specific node that does not observe any nodes is represented as a leader node.

The graph \( \mathcal{G} \) can represent the adjacency matrix, which includes node and edge information. An adjacency matrix is an \( n \times n \) square matrix, and its elements are determined according to the neighboring relationships between nodes. If a graph has no weight at all edges, the elements of the adjacency matrix for the graph are determined as follows:

\[
a_{ij} = \begin{cases} 1, & (i,j) \in E, \\ 0, & \text{else}. \end{cases}
\]

A consensus system for a node state \( x_i \), which converges to the average of neighboring node states is expressed as

\[
\dot{x}_i = \sum_{j \in \mathcal{N}_i} a_{ij} (x_j - x_i).
\]

For all nodes, (2) can be expressed as

\[
\dot{x} = -Lx,
\]

where \( x \) is a state vector \([x_1, x_2, \ldots, x_n]^T\), and \( L \) is a Laplacian matrix whose elements are defined as

\[
l_{ij} = \begin{cases} -a_{ij}, & (i,j) \in E \text{ and } i \neq j, \\ \sum_{k \in \mathcal{N}_i} a_{ik}, & i = j, \\ 0, & \text{else}. \end{cases}
\]

B. NONHOLONOMIC CONSTRAINT
Constraints are relationships between variables that a system physically obeys. For any system, the configuration space \( Q \), which is the set of all the locations that the system can reach can be defined. This set contains generalized coordinates \( q_1, q_2, \ldots, q_n \), which represent the independent variables that can express the system in a minimal number. If the form of a vector constraint \( q = [q_1, \ldots, q_n]^T \in Q \) consisting of generalized coordinates is in the form of (5), a function \( h_i \) is termed a holonomic constraint.

\[
h_i(q) = 0,
\]

where \( i = 1, \ldots, k < n \), and \( k \) is the number of constraints. Next, (5) can be transformed into (6) by time derivative:

\[
\frac{d}{dt} h_i(q) = \sum_{j=1}^{n} \frac{\partial h_i}{\partial q_j} \frac{dq_j}{dt} = 0.
\]

If we define \( H_i := \left[ \frac{\partial h_i}{\partial q_1}, \ldots, \frac{\partial h_i}{\partial q_k} \right] \), (6) can be simplified by using dot product:

\[
H_i \cdot \dot{q} = 0.
\]

Generally, the constraint of a system is described in the form:

\[
\omega_i \cdot \dot{q} = 0.
\]
If $\omega_i$ can be integrated by all the elements of $q$, that is, $\omega_i$ can be expressed in the form $H_i$, (8) is considered a holonomic constraint. Otherwise, if (8) cannot be integrated in the form (5), it is considered a nonholonomic constraint.

A holonomic system indicates that all the constraints in the system are holonomic, and a system that has at least one nonholonomic constraint is considered a nonholonomic system [30].

### C. REFERENCE FRAME

Variables that require geographical criteria must be defined in a specific reference frame. We can define a global reference frame in the form $^g\Sigma$, which is the absolute basis for the measurement of variables. Moreover, we classify a specific reference frame that can be defined for each agent $i$ in a local frame in the form $^\ell\Sigma_i$, where $\ell$ is the abbreviation of local. The agents are represented by a local frame, that is, the local frame corresponding to each agent has an orientation and origin according to the direction and position of the agent. In particular, we distinguish the local frames for holonomic agents, nonholonomic agents, and intermediate state in the form $^h\Sigma$, $^c\Sigma$, and $^{g\ell}_{h\Sigma}\Sigma$, respectively. Moreover, a common frame $^c\Sigma$ is introduced to compare the variables on each local frame [31]. The common frame has a certain angle to which total agents converge to the specific computed angle determined by the graph node edges and initial angles of each agent as its orientation angle.

For variables, a left superscript, left subscript, right superscript, and right subscript are used as the characteristics of the frame, the characteristics of the agent, the agent as an observation reference, and the agent to which the variable belongs, respectively. Particularly, the explanation of the agent is included in the right subscript and separated by a comma. For example, $^{\ell_j}p_j$ is a relative variable $p$ of an agent $j$ observed in the local frame $^h\Sigma$ of an agent $i$.

### D. ORIENTATION ALIGNMENT-BASED FORMATION CONTROL

An OAFC is a decentralized formation control based on orientation alignment and position estimation [27].

\begin{equation}
^g\hat{\theta}_i = -k_\theta \sum_{j \in N_i} a_{ij} \left( ^6\theta_j - ^6\hat{\theta}_j \right),
\end{equation}

\begin{equation}
^c\hat{p}_i = -k_p \sum_{j \in N_i} a_{ij} \left[ \left( ^c\hat{p}_j - ^c\hat{p}_i \right) - \left( ^c_pj - ^cpi \right) \right] + h v_i,
\end{equation}

\begin{equation}
v_i = -k_p \left( ^6p_i - ^6i \right),
\end{equation}

where $k_\theta$, $k_p$, and $k_p$ are the proper gains, $\theta_i$ is the orientation angle of an agent $i$ with respect to $^g\Sigma$, $^6p_j$ is the relative position of an agent $j$ from an agent $i$, $^c\hat{p}_j$ is the estimated position of an agent $i$ in $^c\Sigma$, and $^c_pj$ is the final desired position of an agent $i$.

An orientation alignment (9) converges the orientations of agents to a consensus angle corresponding to the topology of the graph $G$ over time. In this case, the converged angle functions as a common orientation for all the agents, allowing the comparison of their relative positions. Therefore, the frame with the angle as the orientation is set as $^c\Sigma$. A position estimation (10) estimates the position of the agent $^c\hat{p}_j$ based on the relative positions within $^c\Sigma$. A formation control (11) moves the agent so that the estimated position $^c\hat{p}_j$ in $^c\Sigma$ is the same as the desired position $^c_pj$.

### E. INSTANT ORIENTATION CORRECTION STRATEGY

Due to the nonholonomic constraints, direct implementation of orientation alignment control for the holonomic systems is not applicable to the nonholonomic systems. The core concept of IOCS is to follow the virtual desired positions of agents to eventually follow the local desired positions of the agents.

Fig. 2 shows the entire system including the holonomic system-based control law (HSCL), the nonholonomic control stage (NCS), and the IOC). An HSCL is an equation in which the positional variables of OAFC for nonholonomic systems are replaced with the corrected values of IOCS.

\begin{equation}
m\dot{i} = -k_\theta \sum_{j \in N_i} a_{ij} \left( ^6i - ^6\hat{\theta}_j \right),
\end{equation}

\begin{equation}
^c\hat{p}_i = -k_p \sum_{j \in N_i} a_{ij} \left[ \left( ^c\hat{p}_j - ^c\hat{p}_i \right) - R_{\psi_i}^{-1} \left( ^6p_j - ^6p_i \right) \right] + R_{\psi_i}^{-1} m v_i,
\end{equation}

\begin{equation}
m v_i = -k_p R_{\psi_i} \left( ^6p_i - ^6i \right),
\end{equation}

\begin{equation}
^g\theta \dot{\theta}_i = \mathcal{V} \left( ^6\theta_i \right),
\end{equation}

where $^g\theta_i$ is the orientation angle of a nonholonomic agent $i$, $^6\theta_i$ is the intermediate angle of a nonholonomic agent $i$, $^6p_j$ is the intermediate position of a nonholonomic agent $i$, $^6p_i$ is the intermediate position of a nonholonomic agent $i$,
The orientations of nonholonomic agents diverge. Owing to an is used as the target angular velocity. Therefore, the orientation angle difference between \( \theta \) of each agent. The intermediate state calculated using (14) is input to the control target values in (19) are used as control inputs of nonholonomic agents. Notably, a nonholonomic agent cannot perfectly reflect an intermediate state as its physical motion because of its nonholonomic constraints. Therefore, the intermediate angles and positions are pursued by nonholonomic agents, as the relative positions of an agent \( i \) to other nonholonomic agents also diverges. The IOCS corrects the difference in the orientations between \( \theta \) and \( \hat{\theta} \). Because \( \hat{\theta} \) is used as the desired orientation angle of the nonholonomic agent, the control of the intermediate state as its physical motion because of its nonholonomic constraints. Therefore, the intermediate angles \( \hat{\theta} \) and \( \hat{\theta} \) are pursued by nonholonomic agents, as the target orientation angles and positions. This property is represented by the local target position in (19).

\[
\begin{align*}
\dot{\hat{\theta}} &= -k_\psi L (\hat{\theta} - \hat{\theta} - \psi), \\
\dot{\psi} &= -k_\psi L (\hat{\theta} - \hat{\theta} - \psi), \\
\theta_i &= R^{-1}_{\psi} \theta_i, \\
\hat{\theta}_i &= R_{\psi} \hat{\theta}_i, \\
\hat{\theta}_i &= \hat{\theta}_i - \psi_i,
\end{align*}
\]

where \( \hat{\theta} \) is the estimated orientation angles of virtual holonomic agents, \( \psi \) is the converging angle to the orientation angle difference between \( \hat{\theta} \) and \( \hat{\theta} \), and \( \theta_i \) is the local target position. The major challenge arises from the difference between the orientations of \( \hat{\theta} \) and \( \hat{\theta} \). The mismatch of the orientation occurs because of the disturbance, dynamic model, and controller of nonholonomic systems. If a nonholonomic agent follows (9), we must consider the function \( W(x) \) that represents the aforementioned mismatch effect. \( \hat{\theta} \), which does not match the actual movement of nonholonomic agents, is used as the target angular velocity. Therefore, the orientation angles of nonholonomic agents diverge. Owing to an abnormality in \( \hat{\theta} \), the relative position to other nonholonomic agents also diverges.

The IOCS corrects the difference in the orientations between \( \hat{\theta} \) and \( \hat{\theta} \). Therefore, for the local variables of \( \hat{\theta} \) and \( \hat{\theta} \) to have the same value, a correction is required to rotate the local variables of \( \hat{\theta} \) by an angle difference between the orientation angles of \( \hat{\theta} \) and \( \hat{\theta} \). The orientation angle difference is estimated in (17) and (18). In (13) and (14), the rotation matrix \( R_{\psi} \) is applied to the relative positions of an agent \( j \) with respect to the origin of an agent \( i \). Through this process, \( \psi \) converges to \( \hat{\theta} - \hat{\theta} \). Therefore, the relative positions in \( \hat{\theta} \) are rotated by \( +\psi \) from the orientation of \( \hat{\theta} \), and the relationship between the relative positions of an agent \( j \) observed in \( \hat{\theta} \) and \( \hat{\theta} \) is expressed as

\[
\theta_j = R^{-1}_{\psi} \theta_j.
\]

For (14), the local variables in \( \hat{\theta} \) exist on the left-hand side. Therefore, the correction is achieved by rotating the variables in \( \psi \) on the right-hand side by \( +\psi \) instead. Fig. 3 shows the correction process of the IOCS through the values of the local variables in \( \hat{\theta} \).

The intermediate state calculated using (14) is input to the NCS as the local target for each agent. Because the frames to which the local target and nonholonomic agent belong are different, \( \hat{\theta} \) is converted into \( \hat{\theta} \) using (20).

Correction algorithms have complexities for the dimension \( d \) and number of agents \( n \). The local variable correction of an HSCL has a complexity of \( O(d^2) \) because it is a rotational transformation for a dimension. The IOCS for estimating the required rotation angle \( \psi \) has a complexity of \( O(n^2) \). The HSCL expression has a complexity of \( O(n^2d^2) \).
desired angle, and
position, of a nonholonomic agent
as an old local desired position, of the nonholonomic agent
position is more than a tolerance
old virtual desired position and the updated virtual desired
position is also aligned with that of the moving vector of the
nonholonomic agent moving to the virtual desired
position. To address this challenge, an update
position changes. During the simulation, desired positions
were converted to another position at 300 iteration times.
The initial positions and angles of the agents were randomly
arranged for each simulation. Moreover, the simulation
tions including the initial positions, target positions,
and simulation time were set equally for the condition with
and without the IOCS. The desired positions for nonholo-
nomic agents were (320, 360), (200, 240), (320, 120),
and (440, 240) pixels.

A. SIMULATION: COMPARISON OF HSCL WITH AND
WITHOUT IOCS
The simulations were performed for 300 iterations each,
and the initial positions and angles of the agents were ran-
domly arranged for each simulation. Moreover, the simul-
ation conditions including the initial positions, target positions,
and simulation time were set equally for the condition with
and without the IOCS. The desired positions for nonholo-
nomic agents were (320, 360), (200, 240), (320, 120),
and (440, 240) pixels.

B. SIMULATION: IOCS UNDER A DYNAMIC SCENARIO
A simulation was performed for the condition that the desired
position changes. During the simulation, desired positions
were converted to another position at 300 iteration times.
The initial positions and angles of the agents were randomly
arranged for each simulation. The initial desired positions
were (240, 160), (400, 160), (240, 320), and (400, 320)
pixels. The final desired positions were (320, 360), (400, 80),
(80, 400), and (400, 400) pixels.

C. EXPERIMENT: AGENT CONTROL SYSTEM
As shown in Fig 5, four unicycle-modeled E-pucks were
used as nonholonomic system agents (GCtronic, Swiss). The
initial position of the nonholonomic agents x and y were given
randomly within the range of ±3 m, respectively. Addition-
ally, the orientation angles of the agents were given randomly
within the range of ±π rad. Particularly, the orientation of the
leading agent was aligned to that of \( \theta_s \). Therefore, the con-
sensus angle \( \theta_s \) at which the agents finally converge is 0 rad.
The positions and orientations of nonholonomic agents were
measured by the OpenCV-based object detection algorithm
using a web camera that has a maximum video resolution
of 4096 \times 2160 pixels and a maximum frame rate of 90 fps
(Logitech BRIO, Swiss) [33]. However, to prevent computa-
tional delay, object detection is performed at a resolution of
640 \times 480 pixels and a frame rate of 30 fps. In addition, four
distinctive graphical markers were placed on the surface of
each of the agents to measure the positions and orientations
of each agent. After obtaining the positions and orientation

F. APPLICATION TO NONHOLONOMIC AGENTS
A nonholonomic agent must converge to the desired ori-
tention while executing point-to-point motion. One such
algorithm proposed in [32] generates a reference position
according to the following equation:

\[
\vec{\phi}_{i,\lambda} = \vec{\phi}_{i,\lambda+1} = \Delta \phi_{i,\lambda},
\]

where \( \vec{\phi}_{i,\lambda} \) is a virtual desired position, as a new local desired
position, of a nonholonomic agent \( i \), \( \vec{\phi}_{i,\lambda+1} \) is the corrected
desired angle, and \( \Delta \phi_{i,\lambda} \) is the intermediate position,
as an old local desired position, of the nonholonomic agent \( i \).
The nonholonomic agent moves to the virtual desired position
\( \vec{\phi}_{i,\lambda} \) instead of the original local desired position \( \vec{\phi}_{i,\lambda} \). Fig. 4
shows the geometric relationship between the variables in \( \vec{\phi}_{i,\lambda} \).

However, this algorithm cannot achieve the orientation
alignment for a moving target position. In this study, as
the positions of agents gradually converge to a certain value
over time, this corresponds to a case in which the target
position moves. When the target position moves, the virtual
desired position moves accordingly, and the nonholonomic
agent moves toward the virtual desired position. At this time,
as the virtual desired position continues to move in a direction
independent of the target orientation angle, the orientation
of the nonholonomic agent moving to the virtual desired
position is also aligned with that of the moving vector of the
virtual desired position. To address this challenge, an update
step size \( \lambda \) was introduced. The virtual desired position is
updated to a new target whenever the distance between the
old virtual desired position and the updated virtual desired
position is more than a tolerance \( \lambda \). Otherwise, the old virtual
desired position does not update to a new target.

G. UNICYCLE-MODELED AGENT CONTROL
The linear and angular velocities of the unicycle-modeled
nonholonomic agent are calculated using the position, orienta-
tion, and target position as follows:

\[
\begin{align*}
\vec{\gamma}_{i,\lambda} &= \lambda |\vec{\phi}_{i,\lambda}| \cos(\theta_{i,\lambda}), \\
\vec{\omega}_{i,\lambda} &= \lambda \omega_{i,\lambda} \sin(\theta_{i,\lambda}).
\end{align*}
\]

where \( \gamma \) and \( \omega \) are positive constants. For convenience, the
velocities of the nonholonomic agent were defined as (22).

III. EXPERIMENTAL SETUP
In this section, general experimental conditions and the vari-
ous apparatus are explained.

A. SIMULATION: COMPARISON OF HSCL WITH AND
WITHOUT IOCS
The simulations were performed for 300 iterations each,
and the initial positions and angles of the agents were ran-
domly arranged for each simulation. Moreover, the simulation
tions including the initial positions, target positions,
and simulation time were set equally for the condition with
and without the IOCS. The desired positions for nonholo-
nomic agents were (320, 360), (200, 240), (320, 120),
and (440, 240) pixels.

B. SIMULATION: IOCS UNDER A DYNAMIC SCENARIO
A simulation was performed for the condition that the desired
position changes. During the simulation, desired positions
were converted to another position at 300 iteration times.
The initial positions and angles of the agents were randomly
arranged for each simulation. The initial desired positions
were (240, 160), (400, 160), (240, 320), and (400, 320)
pixels. The final desired positions were (320, 360), (400, 80),
(80, 400), and (400, 400) pixels.
angles of the agents, the control input was transmitted via a Bluetooth protocol.

Image processing and feature detection are performed using oriented FAST and rotated BRIEF (ORB) with OpenCV, and K-means clustering was used to estimate the mean position of the marker detected by the camera [34]. In addition, the direction of the object was determined by calculating the intensity centroid for rotated feature detection using a BRIEF algorithm [35]. Key points and descriptors to distinguish each marker were extracted using ORB from the marker, and the image obtained from the camera were compared with each other. Then, the best-matched section of the image was considered as the position of the agents.

### D. SIMULATION: THE UPDATE STEP SIZE EFFICIENCY

A simulation was performed to analyze the convergence efficiency of each agent according to the update step size $\lambda$ and the number of agents. In each simulation, the cumulative angular differences of all the agents during the simulation between different $\lambda$s were recorded for eight cases, namely $\lambda = 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75$, and $2.0$ (pixels). Therefore, a cumulative mean angular velocity error (CMAE) was introduced to evaluate the effect of $\lambda$:

$$
CMAE = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{L} \frac{\left| \theta_{ij}(\lambda = 0.25) - \theta_{ij}(\lambda \neq 0.25) \right|}{t_j},
$$

where $L$ is the number of simulation rounds, $N$ is the number of agents, $\theta_{ij}(\lambda = 0.25)$ is the angle record of nonholonomic agents when $\lambda$ is zero, $\theta_{ij}(\lambda \neq 0.25)$ is the angle record of nonholonomic agents for eight cases in which $\lambda$ is not 0.25, and $t_j$ is the converging time of the agents for each round. Additionally, the CMAE is calculated for four cases according to the number of agents, namely 3, 4, 5, and 6.

### IV. RESULTS

In this section, the experiment results described in section III are presented. Notably, the IOCS parameters are empirically chosen to control each agent (Table 1).

#### A. SIMULATION: WITH AND WITHOUT IOCS

Fig. 7 and 8 show the visualized the formation and position errors for each nonholonomic agents under holonomic system-based control without proposed IOCS approach.
Fig. 7 and 10 show the visualized the formation and position errors for each nonholonomic agents under holonomic system-based control with proposed IOCS approach.

**B. SIMULATION: IOCS UNDER A DYNAMIC SCENARIO**

Fig. 11 and 12 show the visualized the formation and position errors for each nonholonomic agent under a dynamic scenario in which the initial desired position was switched to another position. The formation control was carried out until all of the agents were located in the final desired position area (12 × 12 pixels).

**C. EXPERIMENTAL RESULTS: POSITIONAL RESULT USING THE INSTANT ORIENTATION CORRECTION STRATEGY**

Fig 13 shows the initial formation and the final formation of agents during the experiment. The formation control was carried out until all of the agents were located in the desired position area (12 × 12 pixels).

Fig 14 and Fig 15 shows an overall measured positions and intermediate positions of each agents during the agent control experiment using IOCS.

Fig 16 shows absolute error between the desired positions and the positions of the agents.
The position error of the simulation under a dynamic scenario. The initial desired position changed to the final desired position after 300 iterations from the start of the simulation.

The initial formation and the final formation of nonholonomic agents. Agents were controlled until all of the agents were located in the desired position area.

Measured positions of agents by the camera when proposed IOCS is applied. The agents were controlled until all of the agents are located in the desired position area (12 x 12 pixels).

Intermediate positions of agents when proposed IOCS is applied. Due to the nonholonomic constraint and the control stage, intermediate position was calculated to follow the local desired positions of agents.

Absolute error between the desired positions and the positions of the agents.

Measured orientation angle during the experiment where $\theta$ is the measured orientation angle of the agents. Note that the consensus angle $\theta_\ast$ is set as 0 rad.

Measured orientation angle during the experiment where $\theta$ is the measured orientation angle of the agents. Note that the consensus angle $\theta_\ast$ is set as 0 rad.
FIGURE 18. Angle difference $\Psi$ between the intermediate frame and nonholonomic frame, where $\Sigma_{m}$ is the intermediate frame, $\Sigma_{n}$ is the nonholonomic frame.

FIGURE 19. Absolute error between the orientation angle of common frame and the nonholonomic frame, where $\Sigma_{c}$ is the common frame, and $\Sigma_{n}$ is the nonholonomic frame.

E. SIMULATION: $\lambda$ AND CONVERGE TIME ANALYSIS

Fig 20 shows cumulative mean angular velocity errors (CMAE) according to each value of $\lambda$ (pixels).

Fig 21 shows required converge step according to each value of $\lambda$ (pixels).

V. DISCUSSION

In this section, the considerations of the experiment and the simulation results in section IV are discussed.

A. INFLUENCE OF IOCS ON HSCL

The effect of the IOCS can be confirmed through simulations for nonholonomic agents. During the iterations, nonholonomic agents without the IOCS do not make formation, and intermediate positions do not converge to estimate the positions (Fig. 7 and 8). Conversely, as expected, nonholonomic agents with the IOCS are arranged to the desired positions after 300 iterations (Fig. 9 and 10). Furthermore, the effect of IOCS can be confirmed even in the case where the formation dynamically changes (Fig. 11, and 12). Therefore, introducing a frame correction strategy such as the proposed IOCS when implementing holonomic control laws for agents that have nonholonomic constraints is effective.

B. POSITION AND ANGULAR FOLLOWABILITY OF IOCS METHOD

As expected, by using the IOCS strategy, the holonomic control method (OAFC method) is applied to nonholonomic agents in the experiment (Fig. 14). As shown in the result, to apply the IOCS method, the reference position frame should be computed in advance (Fig. 15). In the agent control experiment, both the absolute and angular errors between the desired positions and the angles of the agents converged to zero after 15 s (Fig. 16 and Fig. 19). However, by implementing the IOCS method to more high-performance designed robot agents, the converging time may be reduced with
higher gains (Table 1). Therefore, the results show that by using the IOCS, other orientation alignment-based control strategies for holonomic systems can be applied to nonholonomic systems, such as a distributed consensus controller approach [26].

C. RELATIONSHIP BETWEEN λ AND CMAE

When applying the IOCS, the angle difference $\psi$ is critical during the instant correction process because it is fundamental to follow the consensus frame (Fig. 18). Therefore, during the estimation of the absolute error between the orientation angle and nonholonomic frame, a bumping epoch, which can be interpreted as an inconstant angular velocity during agent control, was observed (Fig. 19). As expected, the bumping samples originated from the parameter $\lambda$ (23). Particularly, regardless of the number of agents, the results of the CMAE analysis have shown that as $\lambda$ increases, the cumulative mean angular velocity increases (Fig. 20). This is because $\lambda$ is essentially an updated frequency of the intermediate frame $\Sigma$. Therefore, when applying the IOCS, a suitable value for $\lambda$ should be set in advance for the precise control of multiple agents to prevent unstable angular outputs.

D. CONVERGE TIME TO TARGET POSITIONS

Contrarily, $\lambda$ was relevant to the converging time to the desired points (Fig. 21). Notably, $\lambda$ only contributes to the stability of angular control because the agents immediately change their intermediate frames when the distance of $\lambda$ increases. Therefore, as $\lambda$ decreases, the agents can precisely be controlled in angular and positional aspects. Conversely, when $\lambda$ is unnecessarily small, local frames can be over-updated to degrade the performance of the convergence. In addition, depending on the number of agents, the sensitivity of converging time against $\lambda$ was shown to be different. For instance, the mean coverage step iteration of agents when the number of agents was three was smaller than that of four. Therefore, following the usage of the IOCS, $\lambda$ should be carefully chosen not to degrade the performance of the agent system considering the number of agents.

E. LIMITATIONS AND PRACTICAL FEASIBILITY

During the experiment of the agent control system, the OpenCV algorithm with the top-view camera was used to estimate the positions and orientations of each of the agents. However, due to the precision of the camera and the object detection algorithm, the error of positions and orientation angle remained for about 15 pixels, and $\pm 0.3$ radian (Fig. 16, and 16). On the other hand, because the uncertainty of object detection is not considered in the simulation results, it shows a relatively high alignment accuracy (Fig. 10). Therefore, when applying IOCS in an actual environment, it is important to accurately estimate their each location.

In addition, careful consideration should be given to the situation where implementing IOCS aligns various formations. Due to the non-holonomic constraints, orientation alignment is not guaranteed if the desired position is beyond the domain of dimensions. Therefore, desired position and target should be located in the domain of physically feasible boundaries to align the agents effectively. Furthermore, the agents are able to be located closely until each agent collides and is physically not disturbing. Therefore, an appropriate collision avoidance algorithm should be carefully adopted not to degrade the performance of the system.

VI. CONCLUSION

In this study, a correction method, the IOCS, for nonholonomic dynamics was proposed to perform a control law targeting holonomic agents using nonholonomic systems. The IOCS corrected local variables in a local frame used in a holonomic system-based control law by using the difference in the orientation angle between the local frames of nonholonomic agents and intermediate states. By using the IOCS, the divergence of variables due to the difference between the orientations of local frames was successfully removed and the result was verified by experiment and simulations with unicycle-modeled agents. Therefore, nonholonomic agents could execute a holonomic system-based control law by applying the proposed correction strategy.

REFERENCES

[1] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control,” Automatica, vol. 53, pp. 424–440, Mar. 2015.
[2] S.-M. Kang, M.-C. Park, B.-H. Lee, K.-K. Oh, and H.-S. Ahn, “Distance-based formation control: Background, principal results and issues,” J. Inst. Control, Robot. Syst., vol. 19, no. 5, pp. 398–409, May 2013.
[3] X. Ge, Q.-L. Han, D. Ding, X.-M. Zhang, and B. Ning, “A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems,” Neurocomputing, vol. 275, pp. 1684–1701, Jan. 2018.
[4] Y. Rizk, M. Awad, and E. Tunstel, “Decision making in multiagent systems: A survey,” IEEE Trans. Cognit. Develop. Syst., vol. 10, no. 3, pp. 514–529, Sep. 2018.
[5] Z. Lin, B. Francis, and M. Maggiore, “Necessary and sufficient graphical conditions for formation control of unicycles,” IEEE Trans. Autom. Control, vol. 50, no. 1, pp. 121–127, Jan. 2005.
[6] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, “Leader-follower formation control of nonholonomic mobile robots with input constraints,” Automatica, vol. 44, no. 5, pp. 1343–1349, May 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109807004578
[7] K. Listmann, M. Masalawala, and J. Adamy, “Consensus for formation control of nonholonomic mobile robots,” in Proc. IEEE Int. Conf. Robot. Automat., May 2009, pp. 3886–3891.
[8] T. Liu and Z.-P. Jiang, “Distributed formation control of nonholonomic mobile robots without global position measurements,” Automatica, vol. 49, no. 2, pp. 592–600, Feb. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109812005675
[9] X. Liang, H. Wang, Y.-H. Liu, W. Chen, and T. Liu, “Formation control of nonholonomic mobile robots without position and velocity measurements,” IEEE Trans. Robot., vol. 34, no. 2, pp. 434–446, Apr. 2018.
[10] A. Saradagi, V. Muralidharan, V. Krishnan, S. Menta, and A. D. Mahindrakar, “Formation control and trajectory tracking of nonholonomic mobile robots,” IEEE Trans. Control Syst. Technol., vol. 26, no. 6, pp. 2250–2258, Nov. 2018.
[11] E. Montijano, E. Cristofalo, M. Schwager, and C. Sagues, “Distributed formation control of non-holonomic robots without a global reference frame,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2016, pp. 5248–5254.
[12] Q. Van Tran and H.-S. Ahn, “Distributed formation control of mobile agents via global orientation estimation,” IEEE Trans. Control Netw. Syst., vol. 7, no. 4, pp. 1654–1664, Dec. 2020.
K. K. Oh and H. S. Ahn, “Consensus and cooperation in networked multi-agent systems,” Proc. IEEE, vol. 95, no. 1, pp. 215–233, Jan. 2007.

Y. Quan Chen and Z. Wang, “Formation control: A review and a new consideration,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Aug. 2005, pp. 3181–3186.

W. Ren, R. W. Beard, and E. M. Atkins, “A survey of consensus problems in multi-agent cooperation,” in Proc. Amer. Control Conf., Jun. 2005, pp. 1859–1864.

W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” IEEE Control Syst. Mag., vol. 27, no. 2, pp. 71–82, Apr. 2007.

L. Krick, M. E. Broucke, and B. A. Francis, “Stabilisation of infinitesimally rigid formations of multi-robot networks,” Int. J. Control, vol. 82, no. 3, pp. 423–439, 2009, doi: 10.1080/00207170802108441.

K.-K. Oh and H.-S. Ahn, “Formation control of mobile agents based on inter-agent distance dynamics,” Automatica, vol. 47, no. 10, pp. 2306–2312, Oct. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109811003906

S.-M. Kang, J.-G. Lee, and H.-S. Ahn, “Shape and orientation control of moving formation with local measurements in three-dimensional space,” in Proc. IEEE Conf. Control Technol. Appl. (CCTA), Aug. 2018, pp. 1052–1057.

K.-K. Oh and H.-S. Ahn, “Formation control and network localization via orientation alignment,” IEEE Trans. Autom. Control, vol. 59, no. 2, pp. 540–545, Feb. 2013.

M. Basiri, A. N. Bishop, and P. Jensfelt, “Distributed control of triangular formations with angle-only constraints,” Syst. Control Lett., vol. 59, no. 2, pp. 147–154, Feb. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167699910001595

A. N. Bishop, I. Shames, and B. D. O. Anderson, “Stabilization of rigid formations with direction-only constraints,” in Proc. IEEE Conf. Decis. Control Eur. Control Conf., Dec. 2011, pp. 746–752.

A. Franchi and P. R. Giordano, “Decentralized control of parallel rigid formations with direction constraints and bearing measurements,” in Proc. IEEE 51st IEEE Conf. Decis. Control (CDC), Dec. 2012, pp. 5310–5317.

K. K. Oh and H. S. Ahn, “Formation control of mobile agents based on distributed position estimation,” IEEE Trans. Autom. Control, vol. 58, no. 3, pp. 737–742, Mar. 2013.

E. Montijano, D. Zhou, M. Schwager, and C. Sagues, “Distributed formation control without a global reference frame,” in Proc. Amer. Control Conf., Jun. 2014, pp. 3862–3867.

K.-K. Oh and H.-S. Ahn, “Distributed formation control based on orientation alignment and position estimation,” Int. J. Control, Autom. Syst., vol. 16, no. 3, pp. 1112–1119, Jun. 2018.

X. Yu and L. Liu, “Distributed formation control of nonholonomic vehicles subject to velocity constraints,” IEEE Trans. Ind. Electron., vol. 65, no. 2, pp. 1289–1298, Feb. 2015.

G. Bodis and G. Royle, Algebraic Graph Theory. New York, NY, USA: Springer, 2001. [Online]. Available: https://books.google.co.kr/books?id=dpYIje-ZVUYAC

M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, vol. 3. New York, NY, USA: Wiley, 2006.

H.-S. Ahn, Formation Control: Approaches for Distributed Agents (Studies in Systems, Decision and Control), 1st ed. Springer, Apr. 2019. [Online]. Available: https://books.google.co.kr/books?id=RgMwxQEAACAAJ

F. C. Vieira, A. A. D. Medeiros, P. J. Alsina, and A. P. Aráujo, “Position and orientation control of a two-wheeled differentially driven nonholonomic mobile robot,” in Proc. ICINCO, vol. 2, 2004, pp. 256–262.

G. Bradski and A. Kaehler, “OpenCV,” Dr. Dobb’s J. Softw. Tools, vol. 3, p. 120, 2000. [Online]. Available: http://roswiki.autolabor.com.cn/attachments/Events(2)/ICRA2010Tutorial/ICRA_2010_OpenCV_Tutorial.pdf

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in Proc. Int. Conf. Comput. Vis., Nov. 2011, pp. 2564–2571.

R. Wang, Y. Xia, G. Wang, and J. Tian, “License plate localization in complex scenes based on oriented FAST and rotated BRIEF feature,” J. Electron. Imag., vol. 24, no. 5, Sep. 2015, Art. no. 053011.

**CHANYEONG JEONG** was born in Daegu, South Korea, in 1996. He received the B.S. and M.S. degrees in mechanical engineering from the Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2018 and 2020, respectively. He is currently with the Automotive Materials & Components Research and Development Group, Korea Institute of Industrial Technology. His research interests include control systems and autonomous vehicles.

**EUGENE KIM** was born in Busan, South Korea, in 1992. He received the B.S. degree in engineering from the Nagoya Institute of College, Nagoya, Japan, in 2016, and the M.S. and Ph.D. degrees in engineering from Nagoya University, Nagoya, in 2018 and 2021, respectively. He is currently with the Automotive Materials & Components Research and Development Group, Korea Institute of Industrial Technology. His research interests include functional/inherent safety of machinery, safety-related sensor systems, and automotive/remote driving vehicles.

**MYEONG-HWAN HWANG** received the Ph.D. degree in electrical engineering from Chonnam National University, South Korea, in 2021. He is currently with the Automotive Materials & Components Research and Development Group, Korea Institute of Industrial Technology. His research interests include hybrid electric vehicles (HEVs)/electric vehicles (EVs), and motor drive systems.

**HYO-SUNG AHN** (Senior Member, IEEE) received the B.S. and M.S. degrees in astronomy from Yonsei University, Seoul, South Korea, in 1998 and 2000, respectively, the M.S. degree in electrical engineering from the University of North Dakota, Grand Forks, in 2003, and the Ph.D. degree in electrical engineering from Utah State University, Logan, in 2006.

Since July 2007, he has been with the School of Mechatronics and the School of Mechanical Engineering, Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. Before joining the GIST, he was a Senior Researcher with the Electronics and Telecommunications Research Institute, Daejeon, South Korea. He is currently a Professor at the School of Mechanical Engineering, GIST. His research interests include distributed control, aerospace navigation and control, network localization, and learning control.

**HYUN-ROK CHA** received the Ph.D. degree in physics from the Tokyo Institute of Technology, Tokyo, Japan, in 2009. He was with the Samsung Electronics Research Center, Gwangju, for a period of four years. Since 2004, he has been a Senior Researcher with the Automotive Materials & Components Research and Development Group, Korea Institute of Industrial Technology. He is currently a Core Director of the Mobility Core Components and Materials Center, Seonam Division, Korea Institute of Industrial Technology, and a Professor at the University of Science and Technology. His research interests include e-mobility, electric vehicle (EV) platforms, smart vehicle control hybrid-powered drones, and power electronics.