Advances in Robotics Research, *Vol. 2, No. 2 (2018) 161-182*
DOI: https://doi.org/10.12989/arr.2018.2.2.161

Particle swarm optimization-based receding horizon formation control of multi-agent surface vehicles

Donghoon Kim[1a], Seung-Mok Lee[2b], Sungwook Jung[3c], Jungmo Koo[3d] and Hyun Myung[*3]

[1]Seadronix, 193 Muji-ro, Yuseong-gu, Daejeon 34051, Republic of Korea
[2]Department of Mechanical and Automotive Engineering, Keimyung University, 1095 Dalgubeol-daero, Dalseo-Gu, Daegu 42601, Republic of Korea
[3]Urban Robotics Lab., Department of Civil Engineering, Korea Advanced Institute for Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

(Received June 13, 2018, Revised June 17, 2018, Accepted June 18, 2018)

Abstract. This paper proposes a novel receding horizon control (RHC) algorithm for formation control of a swarm of unmanned surface vehicles (USVs) using particle swarm optimization (PSO). The proposed control algorithm provides the coordinated path tracking of multi-agent USVs while preventing collisions and considering external disturbances such as ocean currents. A three degrees-of-freedom kinematic model of the USV is used for the RHC with guaranteed stability and convergence by incorporating a sequential Monte Carlo (SMC)-based particle initialization. An ocean current model-based estimator is designed to compensate for the effect of ocean currents on the USVs. This method is compared with the PSO-based RHC algorithms to demonstrate the performance of the formation control and the collision avoidance in the presence of ocean currents through numerical simulations.

Keywords: formation control; receding horizon control; sequential Monte Carlo; unmanned surface vehicle; collision avoidance

1. Introduction

In the field of robotics, a multi-agent robot system is an actively studied topic. The multi-agent robot system can perform coverage or cooperative tasks including inspection or exploration of unknown environments, reconnaissance or logistics in a large area, and cleaning or transportation, which is advantageous compared to a single robot system. In order to perform the tasks successfully, a formation the robots in their group while tracking a working path. The relative positions are given in the forms of specific patterns that can be changed depending on tasks or

∗Corresponding author, Professor, E-mail: hmyung@kaist.ac.kr
[a]Ph.D., E-mail: donghoon.kim@seadronix.com
[b]Professor, E-mail: seungmok@kmu.ac.kr
[c]Ph.D. Student, E-mail: sungwook87@kaist.ac.kr
[d]Ph.D. Student, E-mail: jungmokoo@kaist.ac.kr

 ISSN: 2287-4976 (Print), 2287-4984 (Online)

environments. Strategies for collision avoidance and disturbance adaptation are also needed to control the formation of unmanned surface vehicles (USVs) because the USV operates in ocean environments that are often more dynamic than ground environments.

Many approaches have been presented to solve the formation control techniques in ocean environments and can be classified into three major categories; leader-follower framework (Breivik *et al.* 2008, Wang *et al.* 2009, Peng *et al.* 2014, Fahimi 2007, Shojaei *et al*. 2015, Kim *et al*. 2016), virtual structure strategy (Fiorelli *et al*. 2006, Ihle *et al*. 2006, Ghommam and Mnif 2009, Børhaug *et al*. 2011, Burger *et al*. 2009, Almeida *et al*. 2010, Lee *et al*. 2014), and behavior-based method (Balch and Arkin 1998, Reif and Wang 1999, Arrichiello *et al*. 2006). The leader-follower framework has an architecture consisting of one leader robot and several follower robots, and provides control laws for follower robots to control their relative positions with respect to the leader robot. The virtual structure strategy considers a formation as a single rigid geometric structure and assigns desired trajectories for the structure. The motion of each robot is translated from that of the structure. The behavior-based method generates the control action for a vehicle by averaging weighted importance of desired behaviors, such as maintain-formation, move-to-goal, avoid-robot, and avoid-static-obstacle (Balch and Arkin 1998).

The formation control problem has been dealt with various robust control or optimization techniques. Skjetne *et al*. (2003) proposed a vectorial backstepping-based formation control method for multiple surface vehicles, where a decentralized controller based on a gradient update and variable synchronization was designed, and its experimental validation was introduced by Ihle *et al*. (2004). Breivik *et al*. (2008) studied an integrator backstepping controller as an approach to a guided leader-follower formation control algorithm for fully actuated vehicles. Peng *et al*. (2014) proposed a neural network-based controller combining adaptive filtering and backstepping techniques. Kim *et al*. (2016) solved the formation control problem by using a guidance-based leader-follower algorithm and a path planning considering the cruising performance of USVs. Børhaug *et al*. (2011), Burger *et al*. (2009) have studied a coordinated straight-line following problem using nonlinear synchronization strategies. Robust feedback control algorithms were proposed to solve formation control problems by considering the uncertainty of parameters or external disturbances in ocean environments: a Lagrange approach to coping with uncertain disturbance by Ihle *et al*. (2006), a sliding mode control by Fahimi (2007), and an adaptive backstepping approach considering time delay between vehicles by Ghommam and Mnif (2009). Furthermore, adaptive control algorithms using an observer for the disturbance such as ocean currents have been studied by Aguiar and Pascoal (2007), Almeida *et al*. (2010), and Peng *et al*. (2014). However, collision avoidance is not significantly considered or solved by the behavior-based control using an artificial potential field (Benjamin *et al*. 2006, Arrichiello *et al*. 2006).

Model predictive control (MPC), also known as receding horizon control (RHC), is one of the techniques utilized to solve finite horizon optimization problems with input and state constraints (Fontes 2001). MPC-based approaches have been applied to the control of a swarm of mobile robots or unmanned aerial vehicles (UAVs) (Chen *et al*. 2010, Gu and Hu 2005). In the field of surface vehicles, Oh and Sun (2010) proposed MPC-based line-of-sight (LOS) guidance and path tracking for a single surface vehicle. Fahimi (2007) incorporated nonlinear MPC into leader-follower formation control of multiple surface vehicles in the presence of static obstacles. Distributed RHC (dRHC) has been studied for the distributed control of multi-agent vehicles where the key issues of dRHC are a reduction of computational complexity and ensuring stability. Gu and Hu (2006) have demonstrated the stability of dRHC by adding a terminal state penalty term to the cost function. Dunbar and Murray (2006) employed an undirected communication

graph to stabilize the multi-agent robot control. Fukushima *et al*. (2013) solved the collision problem during formation control by adding constraints, but this approach suffers from the issue of computational complexity. The computational complexity increases with problem dimension, prediction horizon time, or the number of constraints. Several studies have addressed evolutionary algorithm (EA)-based MPC for formation control, in particular, particle swarm optimization (PSO) has been adopted. Dual mode nonlinear MPC for multiple UAVs with a modified PSO and a collision avoidance strategy has been proposed by Duan and Liu (2010). Duan *et al*. (2013) proposed a hybrid optimization strategy combined PSO and genetic algorithm (GA) for reconfiguration of multiple UAVs' formation. However, these studies do not include stability analyses. Lee *et al*. (2015) have proposed a dynamic cooperatively coevolving PSO (dCCPSO) for multiple mobile robots, which provides real-time operation while guaranteeing asymptotical stability. They also proposed receding horizon PSO (RHPSO) incorporating the collision avoidance strategy to stabilize formation control of a swarm of mobile robots even during formation reconfiguration (Lee and Myung 2015). However, there have been few studies of MPC-based formation control for a group of USVs because it is necessary to consider the conditions of the marine environment in order to apply the studies in other fields to the field of USVs. A swarm of USVs is exposed to disturbances such as ocean currents, wind, and waves while performing marine tasks, so the stability of their formation control must be robust to the disturbances. Also, the formation pattern can be changed depending on the operation or environment. For example, a pattern in single file is advantageous for movement between two points or transportation, and a pattern in line abreast is suitable for performing a coverage operation. Furthermore, since a collision between vehicles can occur when the pattern is changed, or the vehicles are disturbed, a strategy to prevent collisions is needed.

This paper proposes a novel RH formation control of multiple surface vehicles in the presence of ocean currents. The RHC is based on PSO, which includes a particle initialization method based on sequential Monte Carlo (SMC), collision avoidance, and guaranteed stability. The formation control problem is defined by using a kinematic model of surface vehicle and a kinematic controller that includes disturbance feedback. The conditions to guarantee asymptotical stability are derived and utilized to select proper parameter values including control gains, weight, and maximum speed. In addition, to satisfy the conditions of stability, a novel particle initialization method based on SMC is proposed. This method makes the PSO contain several particles that satisfy or are close to terminal state condition at initial generation. Furthermore, a collision avoidance method composed of collision expectation and particle modification is proposed. The proposed method is compared with conventional PSO and RHPSO proposed by Kennedy and Eberhart (1995) and Lee and Myung (2015), respectively, to demonstrate the enhancement of formation control performance through numerical simulations.

The discriminative features of the proposed method are summarized as follows: First, the velocity observer for ocean current estimation is integrated in the proposed method and is used for feedback control of the RHC. This feature minimizes steady-state error, even if the velocity of ocean currents increases. In addition, the SMC-based algorithm generates initial particles which satisfy or are close to the terminal state considering ocean currents. The SMC searches the objective particles using a kernel function with high probability and enhances the convergence rate of the PSO optimization process. Finally, collision avoidance that takes into account the physical dimensions, i.e., the radius of a robot, is also included in the optimization process. This method modifies the trajectory of any particles expected to collide with other robots or adds penalties to the cost values of the particles. Through simulations, this method effectively prevents collisions

when formation patterns are changed despite the disturbance of ocean currents.

The remaining part of this paper is organized as follows. The formation control problem defined in the RHC framework and observer design for the ocean current estimation are presented in Section 2. Section 3 proposes PSO-based RH formation control and stability analysis, SMC-based particle initialization algorithm, and collision avoidance method. In Section 4, numerical simulations and comparison results for the multi-USV formation control problem are presented. Finally, conclusions and the summary of this paper are contained in Section 5.

2. Problem formulation

2.1 Problem definition

This study deals with a formation control problem of a group of N underactuated surface vehicles. It is assumed that each vehicle's surge and yaw motions are directly driven by two parallel independent thrusters, and its sway motion can't be directly controlled. The motion of the surface vehicle j, $\boldsymbol{\eta}_j = [x_j, y_j, \psi_j]^T$, is defined by the position of (x_j, y_j) and the orientation of ψ_j in the global frame as shown in Fig. 1. The velocities of the vehicle, $\boldsymbol{\nu}_j = [u_j, v_j, \omega_j]^T$, are defined by the linear velocities of (u_j, v_j) in surge and sway directions, respectively, and the angular velocity of ω_j in a body-fixed frame. To consider the motion caused by ocean currents, the ocean current disturbance, $\mathbf{V}_{jc} = [V_{jc}\cos\psi_{jc}, V_{jc}\sin\psi_{jc}, 0]^T$, is defined by the speed of V_{jc} and the direction of ψ_{jc} with respect to (x_j, y_j). Ocean currents are assumed as irrotational and slowly-varying. The motion model of the vehicle j is described as the following kinematic equation (Fossen 2002)

$$\dot{\boldsymbol{\eta}}_j = \mathbf{R}_j \boldsymbol{\nu}_{jr} + \mathbf{V}_{jc}, \tag{1}$$

where $\boldsymbol{\nu}_{jr} = \boldsymbol{\nu}_j - \mathbf{R}_j^T \mathbf{V}_{jc}$ is the relative velocity defined by the vehicle's velocity relative to the ocean current disturbance. In other words, $\boldsymbol{\nu}_{jr}$ is a velocity derived by the motion and thruster forces excluding disturbances. $\mathbf{R}_j = \mathbf{R}(\psi_j) = [\cos\psi_j, -\sin\psi_j, 0; \sin\psi_j, \cos\psi_j, 0; 0,0,1]$ is the rotation matrix from the body-fixed frame to the global frame.

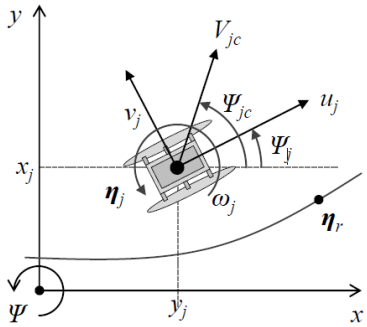


Fig. 1 Reference frames such as global and body-fixed frames and surface vehicle j are presented. The motion state $\boldsymbol{\eta}_j = [x_j, y_j, \psi_j]^T$ is defined in the global frame. The velocity $\boldsymbol{\nu}_j = [u_j, v_j, \omega_j]^T$ is defined in the body-fixed frame. The ocean current disturbance estimated at (x_j, y_j) is expressed by V_{jc} and ψ_{jc}. The reference path is presented as $\boldsymbol{\eta}_r$

The group of vehicles is controlled to track the reference path of $\boldsymbol{\eta}_r$ and configure the desired formation while preventing collisions between vehicles. The desired formation is defined by $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$ where each $\mathbf{p}_j = [x_{jp}, y_{jp}, 0]^T$ is a relative position with reference to $\mathcal{P}$'s center that tracks $\boldsymbol{\eta}_r$. The reference path of $\boldsymbol{\eta}_r$ is continuous and differentiable for all time and its orientation of $\psi_r = \arctan2(\dot{y}_r, \dot{x}_r)$ is tangent to the path. The desired state of the vehicle j, $\boldsymbol{\eta}_{jd} = [x_{jd}, y_{jd}, \psi_{jd}]^T$, is defined by

$$\boldsymbol{\eta}_{jd} = \frac{1}{N_j + \mu_j}\left(\sum_{i \in \mathcal{N}_j}(\boldsymbol{\eta}_i + \mathbf{p}_{ji}) + \mu_j(\boldsymbol{\eta}_r + \mathbf{p}_j)\right), \tag{2}$$

where $\mathcal{N}_j$ is a set of neighbor vehicles of the vehicle j; N_j is the number of components of $\mathcal{N}_j$; and $\mathbf{p}_{ji} = \mathbf{p}_j - \mathbf{p}_i$ is the desired displacement between the two vehicles j and i. μ_j is a Boolean variable to indicate that $\boldsymbol{\eta}_r$ is available to the vehicle j. The state error of each vehicle is obtained in the body fixed frame as follows

$$\boldsymbol{\eta}_{je} = \mathbf{R}_j^T(\boldsymbol{\eta}_{jd} - \boldsymbol{\eta}_j), \tag{3}$$

where $\mathbf{R}_j^T = \mathbf{R}_j^{-1}$ is the transformation matrix from the global frame to the body-fixed frame. Differentiating Eq. (3) with respect to time and substituting Eq. (1), the error dynamics of formation $\dot{\boldsymbol{\eta}}_{je} = [\dot{x}_{je}, \dot{y}_{je}, \dot{\psi}_{je}]^T$ is derived as

$$\begin{aligned} \dot{\boldsymbol{\eta}}_{je} &= \dot{\mathbf{R}}_j^T(\boldsymbol{\eta}_{jd} - \boldsymbol{\eta}_j) + \mathbf{R}_j^T(\dot{\boldsymbol{\eta}}_{jd} - \dot{\boldsymbol{\eta}}_j) \\ &= -\omega_j \mathbf{S}\mathbf{R}_j^T(\boldsymbol{\eta}_{jd} - \boldsymbol{\eta}_j) + \mathbf{R}_j^T(\mathbf{R}_{jd}\boldsymbol{\nu}_{jd} - \mathbf{R}_j\boldsymbol{\nu}_{jr} - \mathbf{V}_{jc}) \\ &= -\omega_j \mathbf{S}\boldsymbol{\eta}_{je} - \boldsymbol{\nu}_{jr} + \mathbf{R}_{je}\boldsymbol{\nu}_{jd} - \mathbf{R}_j^T\mathbf{V}_{jc} \end{aligned} \tag{4}$$

where $\mathbf{R}_{jd} = \mathbf{R}(\psi_{jd})$, $\mathbf{R}_{je} = \mathbf{R}(\psi_{je})$, $\boldsymbol{\nu}_{jd} = [u_{jd}, 0, \omega_{jd}]^T$, $u_{jd} = \sqrt{\dot{x}_{jd}^2 + \dot{y}_{jd}^2}$, $\omega_{jd} = \dot{\psi}_{jd}$, $\psi_{jd} = \arctan2(\dot{y}_{jd}, \dot{x}_{jd})$, and $\mathbf{S} = [0, -1, 0; 1, 0, 0; 0, 0, 0]$ is the skew-symmetric matrix.

2.2 Control objectives

The control law is designed for formation control during path tracking. At first, in a similar way to Chen *et al*. (2010) and Lee and Myung (2015), the control input $\mathbf{u}_{jr} = [u_{jr}, \omega_{jr}]^T$ is defined to stabilize the error state as follows

$$\mathbf{u}_{jr} \triangleq \mathbf{u}_{jd} - \mathbf{u}_{jc} + \mathbf{u}_{je} = \begin{bmatrix} u_{jd}\cos\psi_{je} - V_{jc}\cos(\psi_{jc} - \psi_j) + u_{je} \\ \omega_{jd} + \omega_{je} \end{bmatrix}, \tag{5}$$

where $\mathbf{u}_{jd} = [u_{jd}\cos\psi_{je}, \omega_{jd}]^T$ is a control input for $\boldsymbol{\eta}_{jd}$ and is derived from linear and angular velocities in the body-fixed frame; $\mathbf{u}_{jc} = [V_{jc}\cos(\psi_{jc} - \psi_j), 0]^T$ is a control input against the ocean current disturbance in the body-fixed frame; and $\mathbf{u}_{je} = [u_{je}, \omega_{je}]^T$ is a control input for the error state. Substituting Eq. (5) into Eq. (4) yields the following error dynamics incorporating the controller

$$\begin{pmatrix} \dot{x}_{je} \\ \dot{y}_{je} \\ \dot{\psi}_{je} \end{pmatrix} = \begin{pmatrix} \omega_j y_{je} - u_{je} \\ -\omega_j x_{je} + u_{jd}\sin\psi_{je} - V_{jc}\sin(\psi_{jc} - \psi_j) \\ -\omega_{je} \end{pmatrix}, \tag{6}$$

where v_{jr} is set to 0.

The stability of error dynamics is presented with the assumption that the system inputs are constrained as $0 \le u_{jd} \le u_{\max}$, $|\omega_{jd}| \le \omega_{\max}$. Let's consider a control Lyapunov function as

$$\mathrm{V}(\boldsymbol{\eta}_{je}) = \frac{1}{2}\left(x_{je}^2 + y_{je}^2 + \psi_{je}^2\right). \tag{7}$$

The time derivative of Eq. (7) is obtained using Eq. (6) as

$$\begin{aligned}\dot{V}(\boldsymbol{\eta}_{je}) &= x_{je}\dot{x}_{je} + y_{je}\dot{y}_{je} + \psi_{je}\dot{\psi}_{je} \\ &= x_{je}\left(\omega_j y_{je} - u_{je}\right) + \psi_{je}\left(-\omega_{je}\right) + y_{je}\left(-\omega_j x_{je} + u_{jd}\sin\psi_{je} - V_{jc}\sin\left(\psi_{jc} - \psi_j\right)\right) \\ &= -u_{je}x_{je} - \omega_{je}\psi_{je} + y_{je}\left(-\omega_j x_{je} + u_{jd}\sin\psi_{je} - V_{jc}\sin\left(\psi_{jc} - \psi_j\right)\right).\end{aligned} \tag{8}$$

If $y_{je} \to 0$ as $t \to \infty$ is guaranteed, then $\dot{V}(\boldsymbol{\eta}_{je}) = -u_{je}x_{je} - \omega_{je}\psi_{je} \le 0$, and $x_{je} \to 0$ and $\psi_{je} \to 0$ as $t \to \infty$ are satisfied with the conditions of $u_{je}x_{je} \ge 0$ and $\omega_{je}\psi_{je} \ge 0$. Hence, the system can be asymptotically stable when the condition of $y_{je} = 0$ is satisfied. This stability condition can be guaranteed by employing the RHC algorithm and defining a terminal state region.

2.3 Receding horizon control

The receding horizon control (RHC) algorithm is a technique for searching the optimal solution by predicting the trajectory of a model and minimizing a cost along the predictive time span of $(t, t+T]$. The error dynamics Eq. (6) can be converted as a system function based on the error of formation and control input as follows

$$\dot{\boldsymbol{\eta}}_{je}(t) = f\left(\boldsymbol{\eta}_{je}(t), \mathbf{u}_{je}(t)\right). \tag{9}$$

For the RHC algorithm-based optimization, the cost function for each robot is designed as follows

$$J_j\left(t, \boldsymbol{\eta}_{je}(t), \mathbf{u}_{je}(t)\right) = g_j\left(\boldsymbol{\eta}_{je}(t+T)\right) + \int_t^{t+T} L_j\left(\boldsymbol{\eta}_{je}(\tau), \mathbf{u}_{je}(\tau)\right)d\tau, \tag{10}$$

where $g_j\left(\boldsymbol{\eta}_{je}(t+T)\right) = (\gamma/2)\boldsymbol{\eta}_{je}(t+T)^T\boldsymbol{\eta}_{je}(t+T)$ is the Lyapunov function introduced in Eq. (7) for the terminal state penalty, $L_j\left(\boldsymbol{\eta}_{je}(\tau), \mathbf{u}_{je}(\tau)\right) = \boldsymbol{\eta}_{je}(\tau)^T Q\boldsymbol{\eta}_{je}(\tau) + \mathbf{u}_{je}(\tau)^T R\mathbf{u}_{je}(\tau) + h(\tau)P$ is a running cost function, $Q > 0$ and $R > 0$ are symmetric weight matrices, $P > 0$ is a penalty weight for the collision, $h(\tau)$ is a collision detection function, $\gamma > 0$ is a scalar weight, and T is the finite prediction time. If a collision is expected at time τ, $h(\tau)$ is 1, otherwise $h(\tau)$ is 0.

The cost minimization problem in the RHC algorithm can be formulated as a constrained optimization problem for a multivariate non-linear system as follows

$$\mathbf{u}_{jr}^* = \mathrm{argmax}_{\mathbf{u}_{jr}} J_j\left(t, \boldsymbol{\eta}_{je}(t), \mathbf{u}_{je}(t)\right), \tag{11}$$

subject to the following constraints

$$\dot{\boldsymbol{\eta}}_{je}(\tau) = f\left(\boldsymbol{\eta}_{je}(\tau), \mathbf{u}_{je}(\tau)\right), \tag{12a}$$

$$0 \le u_{jr}(\tau) \le u_{\max}, \tag{12b}$$

$$\left|\omega_{jr}(\tau)\right| \leq \omega_{\max}, \tag{12c}$$

$$\boldsymbol{\eta}_{je}(t+T) \in \Omega, \tag{12d}$$

where $t \in (t, t+T]$, $u_{\max} > 0$ and $\omega_{\max} > 0$ are the maximum control input values, and Ω is the terminal state region.

2.4 Particle swarm optimization

The particle swarm optimization (PSO) algorithm is employed to solve the multi-dimensional nonlinear optimization problem of the RHC algorithm. The PSO algorithm is a kind of swarm intelligence and is the population-based optimization technique developed by Kennedy and Eberhart (1995). The population composed of many particles that are considered potential solutions and move with random velocities in a search space. The particles move randomly and influence each other until the best particle is found, which minimizes or maximizes the cost value. The best of each particle throughout all generations is selected as a personal best, and the best among the personal best solutions is selected as a global best.

The optimization procedure is based on the positions and velocities of all particles corresponding to the states of the system. Let $\mathbf{x}_l^i$ the position vector of the particle i at l-th generation, $\mathbf{v}_l^i$ its velocity vector, $\mathbf{y}^i$ the personal best, and $\hat{\mathbf{y}}$ the global best. The velocity and position vectors of each particle are updated by the following equations

$$\mathbf{v}_{l+1}^i = w_l \mathbf{v}_l^i + c_1 R_1 \left(\mathbf{y}^i - \mathbf{x}_l^i\right) + c_2 R_2 \left(\hat{\mathbf{y}} - \mathbf{x}_l^i\right), \tag{13a}$$

$$\mathbf{x}_{l+1}^i = \mathbf{x}_l^i + \mathbf{v}_{l+1}^i, \tag{13b}$$

where w_l is the inertia weight; c_1 and c_2 are the acceleration coefficients; $R_1 \in [0,1]$ and $R_2 \in [0,1]$ are uniform and independent random numbers.

While some particles converge to the local optimal solution, other particles find other local optima at the same time. In addition, the particles interact with each other and move towards the optimal solution with reference to the best particle's motion. Thus, the PSO algorithm has the advantages of fast convergence rate and global optimization in problem space when compared to other metaheuristic optimization techniques such as evolutionary algorithms and tree search-based approaches, so is suitable for the formation control problem.

2.5 Sequential Monte Carlo

Sequential Monte Carlo (SMC) introduced by Liu and Chen (2001), Moral (1996) is adoped to generate initial particles that satisfy Ω for fast convergence of the PSO algorithm. The SMC algorithm is also a population-based method in the manner of a recursive sequential importance weight sampling and a particle resampling. The distribution of desired solutions is estimated by a set of importance weight values obtained by Monte Carlo sampling that is the process of partial observation using particles (Moral *et al*. 2006). These particles are resampled based on the distribution and propagated toward the best solution depending on importance weight. The importance weight values can be calculated for the optimization purpose using a kernel function, for example, a Gaussian kernel function.

Let $\hat{\mathbf{x}}_k^i$ be multi-dimensional particle i at k-th step, $f(\hat{\mathbf{x}}_k^i)$ be observation values for each particle, y_d be the desired value. The importance weight sampling and the particle resampling are processed with the kernel function $\mathcal{K}(\cdot)$ by following equations

$$w_k^i = \mathcal{K}(y_d - f(\hat{\mathbf{x}}_k^i), \hat{\mathbf{x}}_k^i), \tag{14a}$$

$$\{\hat{\mathbf{x}}_{k+1}^i, w_{k+1}^i\} = \text{Resample}(\hat{\mathbf{x}}_k^i, w_k^i). \tag{14b}$$

2.6 Observer design for ocean currents

In surface environments including the ocean and a river, the surface vehicle is forced to move along an unwanted trajectory because of the presence of external disturbances such as ocean currents, water flow, wind, and waves. Among the disturbances, the influence of ocean currents and water flow is relatively large. Thus, ocean currents are incorporated into the control techniques for marine vehicles as a factor of disturbance. Burger *et al*. (2009) and Almeida *et al*. (2010) proposed an adaptation method for estimating unmodeled parameters involving a spatially constant ocean current model in the manner of feedback control. Aguiar and Pascoal (2007) and Peng *et al*. (2014) proposed observers based on both the system error and the ocean current model to estimate the spatially variable ocean currents.

In this paper, similar to previous studies, the dynamic model of the spatially variable ocean currents introduced by Fossen (2002) is used to design the estimator. Let $\mathbf{V}_{jc} = \widehat{\mathbf{V}}_{jc} + \widetilde{\mathbf{V}}_{jc}$ be the ocean current disturbance, where $\widehat{\mathbf{V}}_{jc}$ is its estimate and $\widetilde{\mathbf{V}}_{jc}$ is its estimation error at the position of vehicle j. The kinematics can be written with its error

$$\mathbf{v}_j(t) = \hat{\mathbf{v}}_j(t) + \tilde{\mathbf{v}}_j(t) = \mathbf{R}_j \boldsymbol{\nu}_{jr} + \widehat{\mathbf{V}}_{jc} + \tilde{\mathbf{v}}_j(t), \tag{15}$$

where $\mathbf{v}_j(t) = \dot{\boldsymbol{\eta}}_j(t)$, $\tilde{\mathbf{v}}_j(t)$ is caused by the disturbance such as the ocean current. $\widehat{\mathbf{V}}_{jc} + \tilde{\mathbf{v}}_j(t)$ can be considered as $\mathbf{V}_{jc}$, and so, $\widetilde{\mathbf{V}}_{jc} = K_{jc}\tilde{\mathbf{v}}_j$ is derived, where $K_{jc} > 0$ is a constant. The ocean current model described in Appendix A is utilized to design an observer for the ocean current disturbance. For the estimation purpose, substituting V_c and w_v in Eq. (31) to $\widehat{\mathbf{V}}_{jc}$ and $\widetilde{\mathbf{V}}_{jc}$, respectively, the ocean current disturbance of vehicle j is observed by

$$\mathbf{V}_{jc}(t) = \lambda \widehat{\mathbf{V}}_{jc}(t-1) + (1-\lambda)\widetilde{\mathbf{V}}_{jc}(t-1) + \lambda K_{jc}\tilde{\mathbf{v}}_j(t). \tag{16}$$

$\mathbf{V}_{jc}(t)$ is used to obtain the control input signal as shown in Eq. (5).

3. Modeling strategy

The formation control problem for a multi-surface vehicle system Eq. (9) can be resolved by using the proposed PSO-based RHC technique. This technique generates the control inputs of the multi-vehicles by minimizing the cost function of the RHC problem Eq. (10) using PSO and guaranteeing its stability using the SMC-based particle initialization algorithm. The procedure of the proposed method is shown in Algorithm 1.

Following the algorithm introduced by Fontes (2001), the RHC algorithm generates a control input for $[t_k : t_k + \delta t]$ by using an optimization method, where $t_k = t_0 + \delta t \cdot k$ is the time step,

δt is the sampling time, and $k = 0, 1, 2, \ldots$. At first, both an input sequence and an error state sequence along the predictive horizon time of T are generated, then the terminal penalty and the running cost are calculated. This process is iteratively performed until the terminal state error converges to 0 or prediction time of δt is over. To integrate the PSO optimization into the RHC procedure, the control input sequence of the vehicle j, $u_j(\tau; t_k) = \left[u_j(\tau; t_k), \omega_j(\tau; t_k)\right]^T$, is assigned to the position vector of the PSO as follows (Lee *et al*. 2015, Lee and Myung 2015)

$$\mathbf{x}_{j,l}^i = \left[u_j(t_k; t_k), u_j(t_{k+1}; t_k), \ldots, u_j(t_{k+M-1}; t_k), \omega_j(t_k; t_k), \omega_j(t_{k+1}; t_k), \ldots, \omega(t_{k+M-1}; t_k)\right] \quad (17)$$

where $0 \leq u_j(t_{k+m}; t_k) \leq v_{\max}$ and $\left|\omega_j(t_{k+1}; t_k)\right| < \omega_{\max}$ for $m = 0,1, \ldots, M-1$; $M = T/\delta t$ is the number of prediction steps.

Algorithm 1 Pseudocode of the proposed PSO for RHC

Acquire $\hat{\mathbf{y}}_n$ from neighbors $n = 0, \ldots, N, \forall n \in \mathcal{N}_j$
Generate initial particle swarm using SMC-based particle initialization algorithm
$l \leftarrow 0$
$\mathbf{y}_j^i \leftarrow \mathbf{x}_{j,l=0}^i, i \in$ [1, …, SwarmSize]
repeat
  for each particle $i \in$ [1, …, SwarmSize] **do**
    if $J_j(\mathbf{x}_{j,l}^i) < J_j(\mathbf{y}_j^i)$ **then**
      $\mathbf{y}_j^i \leftarrow \mathbf{x}_{j,l}^i$
    end if
    if $J_j(\mathbf{y}_j^i) < J_j(\hat{\mathbf{y}}_j)$ **then**
      $\hat{\mathbf{y}}_j \leftarrow \mathbf{y}_j^i$
    end if
  end for
  Update velocity and position values of each particle using Eqs. (13a)-(13b)
  $l \leftarrow l + 1$
until Terminal condition is satisfied
Transmit $\hat{\mathbf{y}}_j$ to neighbors $\mathcal{N}_j$

As shown in Algorithm 1, before beginning the algorithm, the best particles of neighbors are acquired. Prior to the first generation, the proposed SMC-based algorithm generates initial particles $\mathbf{x}_{j,l=0}^i$ that satisfy or are close to the terminal state condition Eq. (12d). At each generation, the probability of collision for each particle is calculated, and the cost function of $J_j(\cdot)$ is calculated. Then, the personal best of $\mathbf{y}_j^i$ and the global best of $\hat{\mathbf{y}}_j$ are updated. This step is repeatedly performed for all particles until l reaches the maximum generation of $L_{\max}$. After the routine is terminated, the global best of $\hat{\mathbf{y}}_j$ is selected and $\hat{\mathbf{y}}_j(t_k; t_k)$ is used as the control input of $\mathbf{u}_j$ for a time duration $[t_k; t_k + \delta t]$.

Dunbar and Murray (2006) proposed two requirements for the stabilization of distributed receding horizon formation control. The stabilization of formation control of the multi-vehicle system is a highly coupled problem, so it is difficult to solve in a distributed system. First, they decoupled the cost function for the stabilization problem by employing the concept of a pair-wise neighbor. The pair-wise neighbor is used to define a formation. Only one of two neighbor vehicles can refer the other vehicle's state, and every vehicle has at least one neighbor vehicle, which allows each robot to connect to the equilibrium position, i.e., the centroid of a formation pattern. This formation architecture decouples the cost function and provides stability conditions of the distributed formation control system. Regarding implementation, the cost of communication is reduced. They also proposed the other requirement as a compatibility constraint such that the

(a) 4 vehicles (b) 5 vehicles

Fig. 2 Examples of the network structures for formation control with four and five vehicles. Each arrow means the accessibility of neighbor vehicle's state. $\boldsymbol{\eta}_j$ and $\boldsymbol{\eta}_r$ indicate the state of vehicle j and the reference path, respectively

differences between the actual state trajectory of a vehicle and the predicted stated trajectories by neighbor vehicles must be small enough. This constraint can be ensured with sufficiently fast update. For this reason, the network structure of formation is set according to the examples shown in Fig. 2.

3.1 Stability of formation control

To guarantee the stability of the proposed formation control method, we employ the stability condition of RHC as follows

$$\dot{g}_j\left(\boldsymbol{\eta}_{je}(t)\right) + L_j\left(\boldsymbol{\eta}_{je}(t), \mathbf{u}_{je}(t)\right) \leq 0, \tag{18}$$

where $t = t_k + T$. If there exits the control input of $\mathbf{u}_{je}(t)$ that satisfies this condition for any terminal state of $\boldsymbol{\eta}_{je}(t) \in \Omega$, the system is considered to be asymptotically stable. This condition is proved by Fontes (2001) and Gu and Hu (2006). Using the system stability conditions Eq. (8), the terminal state feedback controller is selected as

$$u_{je}(t) = K_{jx}x_{je}(t),\ \omega_{je}(t) = K_{j\psi}\psi_{je}(t), \tag{19}$$

where $K_{jx} > 0$ and $K_{j\psi} > 0$ are control gains. The stability condition Eq. (18) can be written using Eq. (6), Eq. (8), Eq. (19), $Q = \mathrm{diag}(q_x, q_y, q_\psi)$, and $R = \mathrm{diag}(r_u, r_\omega)$ as

$$\begin{aligned}
&\dot{g}_j\left(\boldsymbol{\eta}_{je}(t)\right) + L_j\left(\boldsymbol{\eta}_{je}(t), \mathbf{u}_{je}(t)\right) \\
&= -\gamma K_{jx}x_{je}^2 - \gamma K_{j\psi}\psi_{je}^2 + \gamma y_{je}\left(u_{jd}\sin\psi_{je} - V_{jc}\sin\left(\psi_{jc} - \psi_j\right)\right) \\
&\quad + q_x x_{je}^2 + q_y y_{je}^2 + q_\psi \psi_{je}^2 + r_u u_{je}^2 + r_\omega \omega_{je}^2 \\
&= -\left(\gamma K_{jx} - q_x - r_u K_{jx}^2\right)x_{je}^2 - \left(\gamma K_{j\psi} - q_\psi - r_\omega K_{j\psi}^2\right)\psi_{je}^2 \\
&\qquad + q_y y_{je}^2 + \gamma y_{je}\left(u_{jd}\sin\psi_{je} - V_{jc}\sin\left(\psi_{jc} - \psi_j\right)\right)
\end{aligned} \tag{20}$$

The following inequality conditions are required to the negative condition of Eq. (18) be satisfied.

$$\gamma K_{jx} - q_x - r_u K_{jx}^2 \geq 0,\ \gamma K_{j\psi} - q_\psi - r_\omega K_{j\psi}^2 \geq 0. \tag{21}$$

Also, the terminal state condition is defined as

$$\Omega = \{x_{je} \in \mathbb{R}, y_{je} \in \mathbb{R} | x_{je} \geq 0, y_{je} = 0\}. \tag{22}$$

In addition, the control inputs calculated by the controller must satisfy the boundary constraints of Eqs. (12b) and (12c) as follows

$$0 \leq u_{jr}(t) \leq u_{\max}, \tag{23a}$$

$$-\omega \leq \omega_{jr}(t) \leq \omega_{\max}. \tag{23b}$$

These constraints are used to derive additional conditions of the control gains by following the lemmas proposed by Lee and Myung (2015). Using Eq. (5), the lower and upper boundaries of Eq. (23a) become

$$-u_{jd}\cos\psi_{je} + V_{jc}\cos(\psi_{jc} - \psi_j) \leq u_{je}(t) \leq u_{\max} - u_{jd}\cos\psi_{je} + V_{jc}\cos(\psi_{jc} - \psi_j).$$

Since the $x_{je} \geq 0$ is the terminal state region, the boundary condition of the control gain, K_{jx}, is derived as follows

$$\frac{\sup_{\tau>0}(u_{jd}(\tau)+V_{jc}(\tau))}{x_{je}} \leq K_{jx} \leq \frac{u_{\max}-\sup_{\tau>0}(u_{jd}(\tau)+V_{jc}(\tau))}{x_{je}}. \tag{24}$$

From this boundary condition, the condition of $u_{\max} \geq 2\sup_{\tau>0}\left(u_{jd}(\tau) + V_{jc}(\tau)\right)$ is derived. In a similar manner, the boundary condition of $K_{j\psi}$ is obtained from Eq. (23b) as follows

$$0 < K_{j\psi} \leq \frac{\omega_{\max}}{\pi}. \tag{25}$$

If two control gains of K_{jx} and $K_{j\psi}$ satisfy Eqs. (24)-(25), respectively, the terminal state feedback controller Eq. (19) can meet the terminal state condition of Ω.

3.2 Particle initialization algorithm

In this study, the SMC algorithm is incorporated into the proposed PSO algorithm. The SMC algorithm generates particles that satisfy the terminal state condition Eq. (22). Lee *et al*. (2015) have proposed a method to repair particles for the similar purpose by adjusting angular velocities

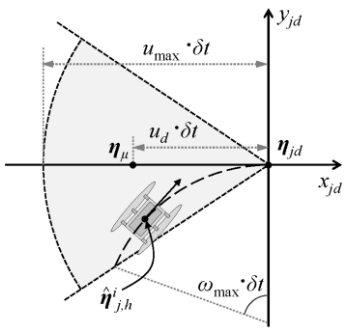


Fig. 3 The set of $\hat{\boldsymbol{\eta}}^i_{je,h}$ candidate positions of vehicles at $M-1$ time step is presented as a shaded region. The center position of the kernel function $\mathcal{K}$ is is marked as $\boldsymbol{\eta}_\mu$

of each particle. This method is a straightforward and fast algorithm. However, by using this approach, it is difficult to adjust both linear and angular velocities. Also, the constraints of the external disturbances can't be incorporated. Thus, the SMC algorithm is utilized to generate initial particles considering the ocean current disturbance.

Algorithm 2 Pseudocode of SMC-based particle initialization algorithm

Initialize particles of $\hat{\mathbf{x}}^i_{j,h=0}$, $i \in [1, \ldots,$ Number of Particles$]$ randomly with constraints Eqs. (12b)-(12c)
repeat
  for each particle $i \in [1, \ldots,$ Number of Particles$]$ **do**
    Calculate $\hat{\boldsymbol{\eta}}^i_{j,h}$ using Eq. (3) and $\hat{\mathbf{x}}^i_{j,h}$ ▷ prediction step
    $\hat{\boldsymbol{\eta}}^i_{je,h} \leftarrow \boldsymbol{\eta}_{jd}(t_k + T; t_k) - \hat{\boldsymbol{\eta}}^i_{j,h}$
    $w^i_{j,h} \leftarrow \mathcal{K}(\hat{\boldsymbol{\eta}}^i_{je,h})$ ▷ $\mathcal{K}$ is a kernel function Eq. (26)
  end for
  Store all particles satisfing Eq. (27) in particle buffer of $\mathcal{X}_j$
  Resample particles using Eq. (14b)
  $h \leftarrow h + 1$
until Terminal condition is satisfied
$\mathbf{x}_{j,l=0} \leftarrow \mathcal{X}_j$ or $\hat{\mathbf{x}}^i_{i,h}$

The proposed SMC method uses randomly generated initial particles. The steps including prediction using Eq. (6), importance weight calculation, and resampling are repeated until the terminal condition is satisfied. For each particle, the position at time step $t_{\mathrm{k+M-1}}$ is estimated as $\hat{\boldsymbol{\eta}}^i_{j,h}$ using Eq. (3) and $\hat{\mathbf{x}}^i_{j,h}$. The error position of $\hat{\boldsymbol{\eta}}^i_{je,h}$ is calculated using $\boldsymbol{\eta}_{jd}(t_k + T; t_k)$ and $\hat{\boldsymbol{\eta}}^i_{j,h}$, which means a path to a vehicle move for the last time step $(t_{k+M-1:k+M}; t_k)$ and is used to calculate the importance weight of $w^i_{j,h}$ based on a kernel function of $\mathcal{K}$. This kernel function is defined by Gaussian distribution as follows

$$\mathcal{K} \sim \mathrm{Gaussian}\left(\boldsymbol{\eta}_{\mu}, \sigma^2\right), \tag{26}$$

where $\boldsymbol{\eta}_{\mu} = \boldsymbol{\eta}_{jd}(t_k + T; t_k) - \mathbf{R}\left(\psi_{jd}\right)\left[u_{jd} \cdot \delta t, 0, 0\right]^T$ is a target position; u_{jd} is the desired speed at $\boldsymbol{\eta}_{jd}(t_k + T; t_k)$; $u_{jd} \cdot \delta t$ is a distance traveled during the last time step; and σ is the standard deviation of the Gaussian distribution. In addition, a penalty is given when the condition of $\left|\hat{\psi}^i_{je,h}/\delta t\right| > \omega_{\max}$ to exclude any particles that cannot satisfy terminal state condition Eq. (22). The shaded region as shown in Fig. 3 is the set of positions estimated by particles without the penalty and with satisfying following conditions

$$\frac{\left|\hat{\psi}^i_{je,h}\right|}{\delta t} < \omega_{\max} \text{ and } \frac{\left|\sqrt{\hat{x}^{i2}_{je,h} + \hat{y}^{i2}_{je,h}}\right|}{\delta t} < u_{\max}. \tag{27}$$

These particles are stored in particle buffer of $\mathcal{X}_j$. $\mathcal{X}_j$ is passed to $\mathbf{x}_{j,l=0}$. After the routine is over, the control input for the last time step, $\left(u_j(t_{k+M-1}; t_k), \omega_j(t_{k+M-1}; t_k)\right)$, is calculated using Eq. (19) and added to $\mathbf{x}_{j,l=0}$.

3.3 Collision avoidance

The collision avoidance is a critical issue in formation control of a swarm of vehicles when the vehicles reshape their formation while tracking a reference path. In the RHC algorithm, the

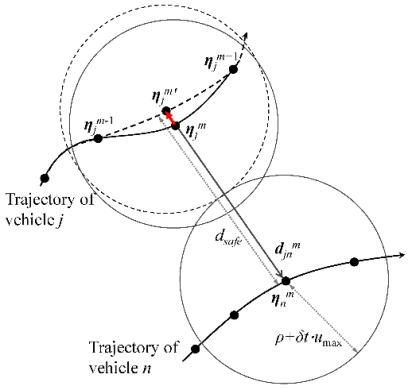


Fig. 4 The trajectories of the vehicles j and n are presented with solid curved lines. The circles present the safety range at the vehicles' positions. The two vehicles following the trajectories are expected to collide at the time step of m. Thus $\boldsymbol{\eta}_n^m$ is moved to $\boldsymbol{\eta}_n^{m\prime}$ so as to the distance between the two vehicles become larger than d_{safe}. $\rho + u_{\max} \cdot \delta t$ is a radius of a vehicle preventing collisions

Algorithm 3 Pseudocode of collision avoidance

for each step m = 1, …, M **do**
  $\mathbf{d}_{jn}^m \leftarrow \boldsymbol{\eta}_n^m - \boldsymbol{\eta}_j^m$
  if $|\mathbf{d}_{jn}^m| < d_{safe}$ **then**
    Compute $\eta_j^{m\prime}$ using Eq. (28).
    ▷ Compute new orientations.
    $\psi_j^{m\prime} \leftarrow \angle|\boldsymbol{\eta}_j^{m\prime} - \boldsymbol{\eta}_j^{m-1}|$
    $\psi_j^{m+1\prime} \leftarrow \angle|\boldsymbol{\eta}_j^{m+1} - \boldsymbol{\eta}_j^{m\prime}|$
    ▷ Compute new linear and angular velocities.
    $u_j'(t_{k+m-1}; t_k) \leftarrow |\boldsymbol{\eta}_j^{m\prime} - \boldsymbol{\eta}_j^{m-1}|/\delta t$
    $u_j'(t_{k+m}; t_k) \leftarrow |\boldsymbol{\eta}_j^{m+1} - \boldsymbol{\eta}_j^{m\prime}|/\delta t$
    $\omega_j'(t_{k+m-1}; t_k) \leftarrow (\psi_j^{m\prime} - \psi_j^{m-1})/\delta t$
    $\omega_j'(t_{k+m}; t_k) \leftarrow (\psi_j^{m+1\prime} - \psi_j^{m\prime})/\delta t$
    $\omega_j'(t_{k+m+1}; t_k) \leftarrow \omega_j(t_{k+m+1}; t_k) + (\psi_j^{m+1} - \psi_j^{m+1\prime})/\delta t$
    if $u_j'(t_{k+m-1:k+m}; t_k)$ are not satisfied Eq. (12b) **then**
      terminate with collision penalty ← 1
    end if
    if $\omega_j'(t_{k+m-1:k+m}; t_k)$ are not satisfied Eq. (12c) **then**
      terminate with collision penalty ← 1
    end if
    ▷ Assign new velocities to the input sequence.
    $u_j(t_{k+m-1:k+m}; t_k) \leftarrow u_j'(t_{k+m-1:k+m}; t_k)$
    $\omega_j(t_{k+m-1:k+m}; t_k) \leftarrow \omega_j'(t_{k+m-1:k+m}; t_k)$
  end if
end for

collision avoidance problem can be solved by using predicted control inputs. A collision avoidance method is proposed by Lee and Myung (2015), which rejects a particle expected to collide with neighbors. They detected a collision when the distance between two particles in each prediction step is closer than d_{safe}, where $d_{safe} = 2(\rho + \delta t \cdot v_{\max})$ and ρ is the radius of a vehicle. In this study, a method of detecting a collision and repairing a particle is proposed. Its process is described as a pseudocode in Algorithm 3. As demonstrated by Lee and Myung (2015), d_{safe} is used to anticipate the potential of collision as shown in Fig. 4. Let $\mathbf{d}_{jn}^m = \boldsymbol{\eta}_n^m - \boldsymbol{\eta}_j^m$ be the vector of the relative position between vehicles j and n at m-th prediction step, $\boldsymbol{\eta}_n^m$ and $\boldsymbol{\eta}_j^m$ be the positions of the vehicles' predicted trajectories. When $|\mathbf{d}_{jn}^m| < d_{safe}$, the position $\boldsymbol{\eta}_j^m$ is moved as

much as $d_{safe} - \left|\mathbf{d}_{jn}^{m}\right|$ as follows

$$\boldsymbol{\eta}_n^{m\prime} = \boldsymbol{\eta}_n^m - \left(d_{safe} - \left|\mathbf{d}_{jn}^{m}\right|\right)\angle\mathbf{d}_{jn}^{m}, \tag{28}$$

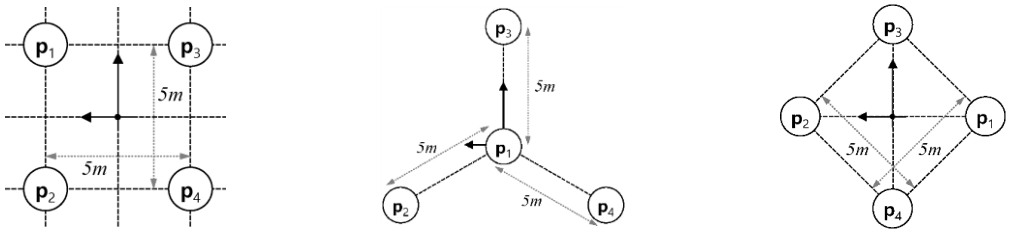
where $\angle\mathbf{d}_{jn}^{m}$ is the direction of the vector. $\boldsymbol{\eta}_n^{m\prime}$ is used to repair the control inputs of the particle by computing new directions, $\psi_j^{m\prime}$ and $\psi_j^{m+1\prime}$, linear and angular velocities, $u_j'(t_{k+m-1:k+m}; t_k)$ and $\omega_j'(t_{k+m-1:k+m}; t_k)$, as presented in Algorithm 3. If the repaired control inputs satisfy Eqs. (12b)-(12c), respectively, the new values are assigned to the control input sequence of the particle. Otherwise, this particle is regarded to occur a collision and is given a penalty of the cost Eq. (10).

4. Simulation results of receding horizon formation control

In this section, the simulation results and analyses are presented. The proposed formation control method is compared with the conventional PSO-based RHC and RHPSO (Lee and Myung 2015) to verify the performance of formation control and collision avoidance. The simulations are conducted for the three methods with the same parameter settings. Each PSO optimizes the formation control problem with 50 particles, the maximum number of generations, $L_{\max} = 50$, the acceleration coefficients, $c_1 = 2.0$ and $c_2 = 2.0$, and the inertia weight of each generation, $w_l = 0.9 - 0.5 \times l/L_{\max}$, where l is the generation index. w_l decreases as the generation number increases, which is used to adjust between global search and local search (Duan and Liu 2010, Lee and Myung 2015). The maximum linear velocity and angular velocity are set to $V_{\max} = 2.0\ m/s$ and $\omega_{\max} = \pi/2$, respectively. The control parameters are selected to satisfy constraints Eq. (21) as the terminal penalty weight of $\gamma = 2.0$, the weight matrices of $Q = \mathrm{diag}(0.2, 0.2, 0.02)$ and $R = \mathrm{diag}(0.04, 0.04)$, the collision penalty value of $P = 50$, and control gains of $K_{jx} = 0.2$ and $K_{j\psi} = 0.2$. The constant values for the collision avoidance are set to robot radius of $\rho = 0.5\ m$, the sampling period of $\delta t = 0.2\ s$, and thus, the safety distance is derived to $d_{safe} = 1.4\ m$. The number of prediction step M is set to 10. The disturbance for each vehicle is generated by the ocean current model Eq. (16) with $V_{jc} = 0.4\ m/s \pm 15\%$ and $\psi_{jc} = 45° \pm 10\%$.

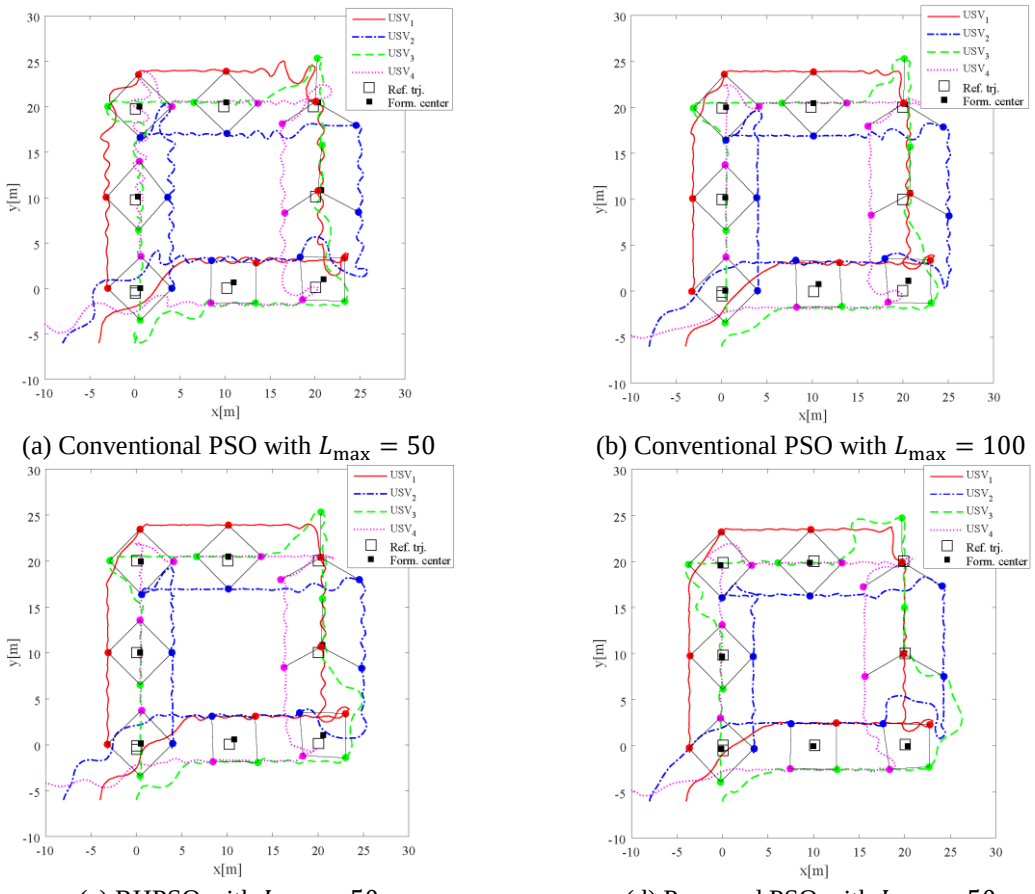
Four USVs are prepared for the simulation with initial positions of $\boldsymbol{\eta}_1 = [-4, -6, \pi/2]^T$, $\boldsymbol{\eta}_2 = [-8, -6, \pi/2]^T$, $\boldsymbol{\eta}_3 = [0, -6, \pi/2]^T$, and $\boldsymbol{\eta}_4 = [-12, -6, \pi/2]^T$, respectively. Additionally, the reference trajectory of the square-shaped path is used to verify path tracking performance and the effectiveness of the collision avoidance while changing the formation pattern, as follows: $\boldsymbol{\eta}_r(t) = [0.5t, 0, 0]^T$ for $0 \le \mathrm{t} < 40$ s; $\boldsymbol{\eta}_r(t) = [20, 0.5t, \pi/2]^T$ for $40 \le \mathrm{t} < 80$ s; $\boldsymbol{\eta}_r(t) = [20 - 0.5t, 20, \pi]^T$ for $80 \le \mathrm{t} < 120$ s; and $\boldsymbol{\eta}_r(t) = [0, 20 - 0.5t, -\pi/2]^T$ for $120 \le \mathrm{t} < 160$ s. During the simulation, the transitions of formation occur twice as presented in Fig. 5. Three patterns of the formation such as P_1, P_2 and P_3 and their retention time are described.

The simulations are performed 30 times for each algorithm. Fig. 6 shows the best trajectories generated by the conventional PSO approaches with both $L_{\max} = 50$ and $L_{\max} = 100$, RHPSO with $L_{\max} = 50$, and the proposed PSO-based method with $L_{\max} = 50$, respectively. Each best trajectory is selected based on the error of $e(t)$ defined by Eq. (29) and the occurrence of collision. The minimum error and the occurrence were preferred. In the case of the conventional PSO, collisions aren't taken into account, so collisions occur in all results. The result of the conventional PSO shows a large oscillation in the trajectories of the vehicles compared with other results as shown in Fig. 6. The difference is considered as coming from the particle repair method of



(a) P_1(0≤t≤40 s) (b) P_2 (40≤t≤80 s) (c) P_3(t>80 s)

Fig. 5 Formation patterns with four vehicles. The minimum distance between two vehicles is set to 5 *m*. The positions of the vehicles are indicated as $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$ with respect to the reference trajectory $\boldsymbol{\eta}_r$. Each caption indicates the retention time of the pattern

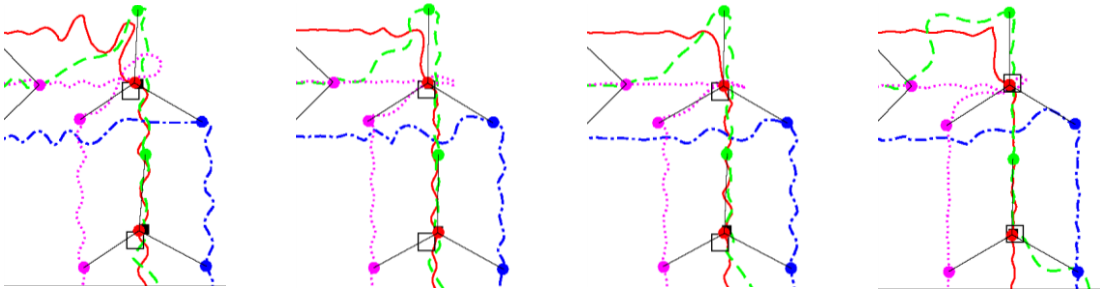


(a) Conventional PSO with $L_{\max} = 50$ (b) Conventional PSO with $L_{\max} = 100$

(c) RHPSO with $L_{\max} = 50$ (d) Proposed PSO with $L_{\max} = 50$

Fig. 6 Results of formation control simulations for four USVs. Trajectories, formation patterns, sampled positions of four USVs are presented. Dots indicate the sampled positions, solid straight lines indicate the formation patterns, dark squares indicate the center of formation, and light squares indicate the reference trajectory. These are sampled at every 20 s

RHPSO and the particle initialization method the proposed PSO. Since the two methods make



(a) Conventional PSO with $L_{\max}$=50 (b) Conventional PSO with $L_{\max}$=100 (c) RHPSO with $L_{\max}$=50 (d) Proposed PSO with $L_{\max}$=50

Fig. 7 The results of simulations from 60 *s* to 110 *s* are presented. The legends of the results are same to those of Fig. 6

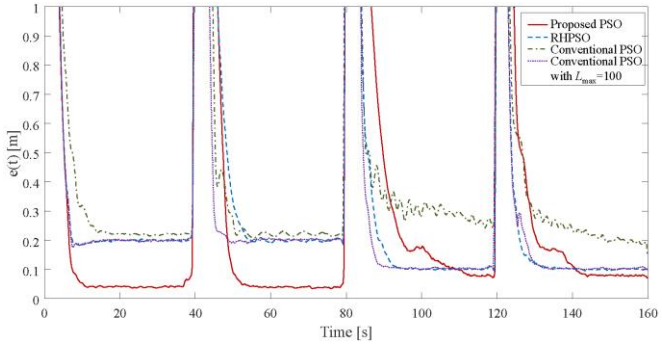


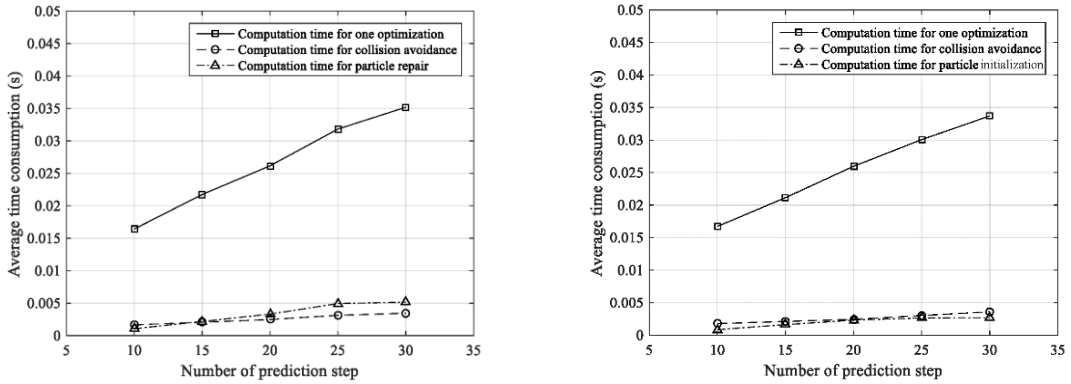
Fig. 8 Mean values of $e(t)$ generated by the proposed PSO-based, RHPSO-based, conventional PSO-based, and conventional PSO (with *L*max=100)-based approaches for 30 runs, respectively. The proposed method shows minimum error even if there exists the ocean current disturbance

particles satisfy the terminal state constraint with high probability, the vehicles can be rapidly stabilized. Additional simulations using the conventional PSO-based method with L_{max}=100 are carried out to compare the results. As the increase of the maximum number of generation doubles, the stabilization performance becomes comparable. Fig. 7 shows parts of the results from 60 s to 110 s. The oscillation of trajectories rapidly decreases in Fig. 7(b) compared with Fig. 7(a). Through the results, the particle repair and initialization methods are analyzed to provide the effect of reducing the number of generations. Furthermore, the results demonstrate that the proposed PSO makes less error than other methods when comparing the errors of the four methods as shown in Fig. 8.

An index to measure the performance of the algorithms are defined as follows

$$e(t) = \sqrt{\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\eta}_{je}^{T} Q \boldsymbol{\eta}_{je}}, \tag{29}$$

where $e(t)$ means the error computed at time t and $N = 4$. Q is set as $\mathrm{diag}(0.2,0.2,0)$ to measure the mean error of the relative positions of the vehicles. Based on this index, the mean errors of the simulation results are shown in Fig. 8. This figure shows that the errors increase at the time of 40 *s*, 80 *s* and 120 *s* and converge to their steady-state values. The conventional method with $L_{\max} = 100$ shows fast convergence rate, but collisions always occurred. The proposed PSO-based

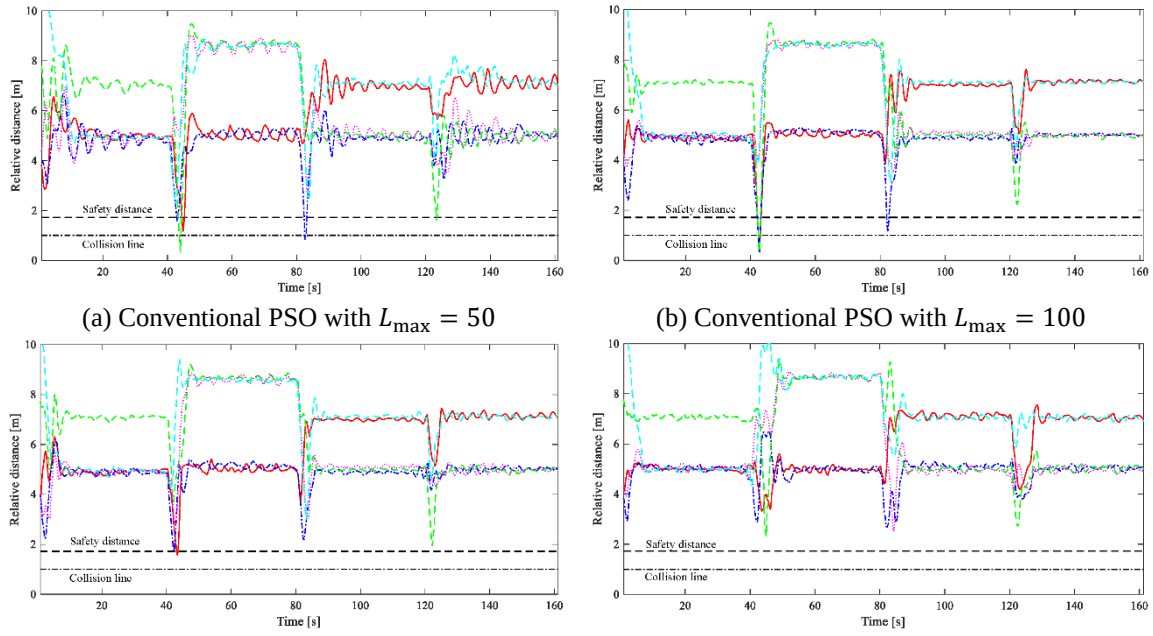


(a) RHPSO (b) Proposed PSO

Fig. 9 Consumption time for one optimization of one USV. Computation time for the collision avoidance, the particle repair, the particle initialization is also presented with a dashed line and dashed-dotted line, respectively

Table 1 Result of collision occurrence and success rate of collision avoidance

	RHPSO		Proposed PSO	
	N_{co}	p_{ca}(%)	N_{co}	p_{ca}(%)
$V_c = 0.2\,m/s$	32	64	2	97.8
$V_c = 0.3\,m/s$	27	70	6	93.3
$V_c = 0.4\,m/s$	30	67.7	15	83.3



(a) Conventional PSO with $L_{\max} = 50$ (b) Conventional PSO with $L_{\max} = 100$

(c) RHPSO with $L_{\max} = 50$ (d) Proposed PSO with $L_{\max} = 50$

Fig. 10 Relative distances among four USVs of the best result are presented. Collisions occur when the conventional PSO is used

method shows fast convergence rate, the lowest steady-state error, but convergence time delay occurred due to collision avoidance at the time of 80 *s* and 120 *s* respectively.

The relative distances among the vehicles are presented in Fig. 10. The collision line and safety line as shown in the figure are defined by a collision threshold of 2ρ and $d_{safe} = 2(\rho + u_{\max} \cdot \delta t)$, respectively. If the relative distance between two vehicles is lower than the collision threshold, a collision occurs. In the cases of RHPSO and the proposed PSO, the relative distance errors among vehicles increases near the corners because the two methods select sequences of control inputs to avoid collisions.

4.1 Computation time comparison

During the simulations, the computation time consumed to optimize the control input is measured with a set of the number of prediction steps including 10, 15, 20, 25, and 30. The computation time to perform the collision avoidance, the particle repair method, and the particle initialization method is also measured for one optimization process. The algorithms have been implemented with C++ language. All simulations were conducted on a desktop computer with 3.4 GHz CPU and 8 GB memory. Fig. 9 shows the average consumption time for the optimization of one vehicle using the RHPSO and the proposed PSO, respectively. The average computation time for the optimization, the collision avoidance, the particle repair, the particle initialization linearly increased as the number of prediction steps increases. However, it is confirmed that the optimization process can be computed sufficiently within the time step δt=0.2 *s*.

4.2 Collision avoidance

Collision avoidance is the major feature of both the proposed PSO and RHPSO-based approaches. To verify the effectiveness of collision avoidance, the number of collision occurrences is counted or 30 runs. The probability of collision between two vehicles increases when the formation pattern changes or the trajectory bends sharply. Thus, three points such as (20, 0), (20, 20), and (0, 20) are selected to count the occurrence of the collision, N_{co}. Vehicles arrive at these points 40, 80, and 120 seconds after the start. If the relative distance between any two vehicles decreases under 2ρ as shown in Fig. 10, the collision occurrence increases by 1. The disturbance imposed on the vehicles depends on the ocean current speed, so three tests are performed with the conditions of V_c=0.2 *m*/*s*, V_c=0.3 *m*/*s* and V_c=0.4 *m*/*s*. The collision occurrence, Nco, and the success rate of collision avoidance, p_{ca}, are listed in Table 1. The success rate is measured by using $p_{ca} = (1 - N_{co}/N_{total}) \times 100(\%)$, where N_{total}=90 is the number of collision observations at the three points for 30 runs. The success rate of the RHPSO-based method is obtained as the average of 67.7 %. Most of the collisions occurred when switching formation from P_1 to P_2. On the other hand, the proposed PSO-based method demonstrates a high success rate compared with the RHPSO-based method. This result shows that the proposed method can avoid collisions even if there are significant disturbances.

5. Conclusions

This study proposes a novel RH formation control of a group of underactuated USVs in the presence of ocean currents. The formation control problem was formulated for the RHC algorithm based on the kinematic model of a USV. The observer for the ocean current disturbance based on

the ocean current model is proposed to cope with external disturbances. A novel method for collision avoidance was proposed to prevent a collision during coordinated path tracking, and its effectiveness is verified through the simulations. The SMC-based particle initialization algorithm generates the initial particles of the PSO to satisfy the terminal state constraint for guaranteeing asymptotical stability. In addition, through the derivation of the stability conditions of control parameters, the appropriate parameter values were selected. Finally, the performance of the proposed PSO-based RH formation control was demonstrated through numerical simulations. The proposed technique is effective to cooperatively track a path in the presence of ocean currents and to prevent collision between vehicles when the robots reconfigure their formation. For further study, the authors plan to focus on finding the minimum adjacent USVs needed to calculate the formation error. If more USVs belong to the group, the optimization becomes more complicated. Therefore, the search algorithm will alleviate the complexity of the optimization constraints. In addition, we are considering the use of time-varying 2D current maps. This will provide a simulation environment similar to the actual ocean environment and will contribute to validating the applicability of the proposed approach to real vehicles.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (grant number NRF-2013R1A1A1A05011746).

References

Aguiar, A.P. and Pascoal, A.M. (2007), "Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents", *J. Control*, **80**(7), 1092-1108.

Almeida, J., Silvestre, C. and Pascoal, A. (2010), "Cooperative control of multiple surface vessels in the presence of ocean currents and parametric model uncertainty", *J. Robust Nonlin. Control*, **20**(14), 1549-1565.

Arrichiello, F., Chiaverini, S. and Fossen, T.I. (2006), "Formation control of underactuated surface vessels using the null-space-based behavioral control", *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October.

Balch, T. and Arkin, R.C. (1998), "Behavior-based formation control for multirobot teams", *IEEE T. Robot. Autom.*, **14**(6), 926-939.

Benjamin, M.R., Leonard, J.J., Curcio, J.A. and Newman, P.M. (2006), "A method for protocol-based collision avoidance between autonomous marine surface craft", *J. Field Robot.*, **23**(5), 333-346.

Børhaug, E., Pavlov, A., Panteley, E. and Pettersen, K.Y. (2011), "Straight line path following for formations of underactuated marine surface vessels", *IEEE T. Control Syst. Technol.*, **19**(3), 493-506.

Breivik, M., Hovstein, V.E. and Fossen, T.I. (2008), "Ship formation control: A guided leader-follower approach", *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July.

Burger, M., Pavlov, A., Børhaug, E. and Pettersen, K.Y. (2009), "Straight line path following for formations of underactuated surface vessels under influence of constant ocean currents", *Proceedings of the American Control Conference*, St. Louis, Missouri, U.S.A., June.

Chen, Y. and Wang, Z. (2005), "Formation control: A review and a new consideration", *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, July.

Chen, J., Sun, D., Yang, J. and Chen, H. (2001), "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme", *J. Robot. Res.*, **29**(6), 727-747.
Duan, H. and Liu, S. (2010), "Non-linear dual-mode receding horizon control for multiple unmanned air vehicles formation flight based on chaotic particle swarm optimisation", *IET Control Theor. Appl.*, **4**(11), 2565-2578.
Duan, H., Luo, Q., Shi, Y. and Ma, G. (2013), "Hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration", *IEEE Comput. Intel. Mag.*, **8**(3), 16-27.
Dunbar, W.B. and Murray, R.M. (2006), "Distributed receding horizon control for multi-vehicle formation stabilization", *Automatica*, **42**(4), 549-558.
Fahimi, F. (2007), "Non-linear model predictive formation control for groups of autonomous surface vessels", *J. Control*, **80**(8), 1248-1259.
Fahimi, F. (2007), "Sliding-mode formation control for underactuated surface vessels", *IEEE T. Robot.*, **23**(3), 617-622.
Fiorelli, E., Leonard, N.E., Bhatta, P., Paley, D.A., Bachmayer, R. and Fratantoni, D.M. (2006), "Multi-AUV control and adaptive sampling in Monterey Bay", *IEEE J. Ocean. Eng.*, **31**(4), 935-948.
Fontes, F.A. (2001), "A general framework to design stabilizing nonlinear model predictive controllers", *Syst. Control Lett.*, **42**(2), 127-143.
Fossen, I.T. (2002), *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, in *Marine Cybernetics*.
Fukushima, H., Kon, K. and Matsuno, F. (2013), "Model predictive formation control using branch-and-bound compatible with collision avoidance problem, *IEEE T. Robot.*, **29**(5), 1308-1317.
Ghommam, J. and Mnif, F. (2009), "Coordinated path-following control for a group of underactuated surface vessels", *IEEE T. Ind. Electron.*, **56**(10), 3951-3963.
Gu, D. and Hu, H. (2005), "A stabilizing receding horizon regulator for nonholonomic mobile robots", *IEEE T. Robot.*, **21**(5), 1022-1028.
Gu, D. and Hu, H. (2006), "Receding horizon tracking control of wheeled mobile robots", *IEEE T. Control Syst. Technol.*, **14**(4), 743-749.
Ihle, I.A.F., Jouffroy, J. and Fossen, T.I. (2006), "Formation control of marine surface craft: A Lagrangian approach", *IEEE J. Ocean. Eng.*, **31**(4), 922-934.
Ihle, I.A.F., Skjetne, R., and Fossen, T.I. (2004), "Nonlinear formation control of marine craft with experimental results", *Proceedings of the 43rd IEEE Conference on Decision and Control*, Nassau, Bahamas, May.
Kennedy, J. and Eberhart, R. (1995), "Particle swarm optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, WA, Australia, December.
Kim, H., Kim, D., Kim, H., Shin, J.U. and Myung, H. (2016), "An extended any-angle path planning algorithm for maintaining formation of multi-agent jellyfish elimination robot system", *J. Control Autom. Syst.*, **14**(2), 598-607.
Lee, S.M. and Myung, H. (2015), "Receding horizon particle swarm optimisation-based formation control with collision avoidance for non-holonomic mobile robots", *IET Control Theor. Appl.*, **9**(14), 2075-2083.
Lee, S.M., Kim, H., Lee, S. and Myung, H. (2014), "Nash equilibrium-based geometric pattern formation control for nonholonomic mobile robots", *Adv. Robot. Res.*, **1**(1), 41-59.
Lee, S.M., Kim, H., Myung, H. and Yao, X. (2013), "Cooperative coevolutionary algorithm-based model predictive control guaranteeing stability of multirobot formation", *IEEE T. Control Syst. Technol.*, **23**(1), 37-51.
Liu, J.S. and Chen, R. (1998), "Sequential Monte Carlo methods for dynamic systems", *J. Am. Stat. Assoc.*, **93**(443), 1032-1044.
Moral, P.D. (1996), "Non-linear filtering: Interacting particle resolution", *Markov Proc. Related Fields*, **2**(4), 555-581.
Moral, P.D., Doucet, A. and Jasra, A. (1996), "Sequential Monte Carlo samplers", *J. Royal Stat. Soc. Ser. B Stat. Met.*, **68**(3), 411-436.
Oh, S.R. and Sun, J. (2010), "Path following of underactuated marine surface vessels using line-of-sight

based model predictive control", *Ocean Eng*., **37**(2-3), 289-295

Peng, Z., Wang, D., Wang, H., and Wang, W. (2014), "Coordinated formation pattern control of multiple marine surface vehicles with model uncertainty and time-varying ocean currents", *Neural Comput. Appl*., **25**(7-8), 1771-1783.

Reif, J.H. and Wang, H. (1999), "Social potential fields: A distributed behavioral control for autonomous robots", *Robot. Auton. Syst*., **27**(3), 171-194.

Shojaei, K. (2015), "Leader-follower formation control of underactuated autonomous marine surface vehicles with limited torque", *Ocean Eng*., **105**, 196-205.

Skjetne, R., Ihle, I.A.F. and Fossen, T.I. (2003), "Formation control by synchronizing multiple maneuvering systems", *Proceedings of the 6th IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, September.

Wang, Y., Yan, W. and Yan, W. (2009), "A leader-follower formation control strategy for AUVs based on line-of-sight guidance", *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation*, Changchun, China, August.

HM

Appendix

A. Ocean current model

Ocean currents, $\mathbf{V}_\mathrm{c}$, are assumed to be irrotational and slow-varying, are specified by the speed of $V_\mathrm{c} \in [V_{c,\min}, V_{c,\max}]$ and the direction of ψ_c with reference to the global frame, where $V_{c,\min}$ and $V_{c,\max}$ are the minimum and maximum speed. The ocean current model introduced by Fossen (2002) is a form of the 1st Gauss-Markov process defined in two-dimensional space as follows

$$\dot{V}_\mathrm{c} + \lambda_v V_\mathrm{c} = w_v(t), \tag{30}$$

where λ_v is the time constant value and $\omega_v(t)$ is a Gaussian random process. Its state transition equation is approximated as follows

$$V_\mathrm{c}(t+1) = \lambda_v V_\mathrm{c}(t) + (1-\lambda_v)\lambda_v \overline{V}_\mathrm{c}(t) + \sqrt{1-\lambda_v^2} w_v(t), \tag{31}$$

where $\overline{V}_\mathrm{c}(t)$ is the mean value of $V_\mathrm{c}(t)$. ψ_c is generated by the similar process to Eqs. (30)-(31) with the parameters of λ_ψ and $w_\psi(t)$.