Formation control scheme with reinforcement learning strategy for a group of multiple surface vehicles

Pham Dinh Duong
Nguyen Xuan Khai
Dang Van Trong

Assoc. Prof. Dao Phuong Nam

Hanoi University of Science and Technology

Outlines

1. Introduction (1)

Abstract

- This project integrates formation tracking control and optimal control for a fleet of multiple surface vehicles (SVs)
- The proposed scheme comprises 2 core components: (1) a high-level displacement-based formation controller and (2) a low-level reinforcement learning (RL)-based optimal control strategy for individual SV agent

Figure: Collaborative SVs (Source: Internet)

1. Introduction (2)

The proposed formation control structure for multiple SVs

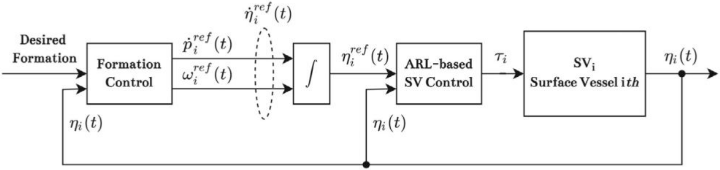

Figure: The proposed control scheme

1. Introduction (3)

The high-level displacement-based controller

- employs a modified gradient method
- guide the SVs in achieving desired formations
- translates the desired formation and trajectory into individual reference trajectories that are feasible

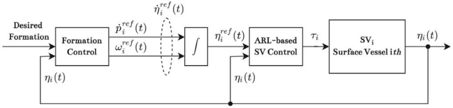

Figure: The high-level controller

1. Introduction (4)

The low-level reinforcement learning (RL)-based optimal control strategy

- incorporates the RL algorithm to solve Optimal control problem
- transforms the time-varying closed agent system into an equivalent autonomous system

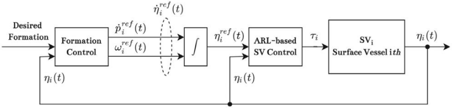

Figure: The low-level controller

2. The problem formulation and preliminaries
2.1. Mathematical model of each agent (1)

The full-actuated model of SV

- $\dot{\eta}_i(t) = J'(\eta_i)\zeta_i(t)$
- $M(\eta_i)\dot{\zeta_i}(t) + C(\zeta_i)\zeta_i + D(\zeta_i)\zeta_i + g(\eta_i) = \tau_i + \Delta_i(\eta_i, \zeta_i, \dot{\zeta}_i, t)$

Where:

- $\eta_i = [x_i, y_i, \psi_i]^T$: position and heading angle in the earth-fixed frame
- $\zeta_i = [u_i, v_i, r_i]^T$: surge, sway and yaw linear velocities in the SV body-fixed frame
- τ_i: the control input
- $J'(\eta_i) = \begin{bmatrix} cos(\psi_i) & -sin(\psi_i) & 0 \\ sin(\psi_i) & cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $\Delta_i(\eta_i, \zeta_i, \dot{\zeta}_i, t)$ is the dynamic uncertainties and external disturbances in the SVs model.

2.2. Graph theory and control objective (1)

Graph theory

For a network comprising n SV agents:

- represent this network using a graph group $G = (S, \varepsilon)$ where $S = 1, , n$ defines the vertex set, and $\varepsilon \subset SxS$ defines the connection set.
- each SV's location is denoted by $p_i = [x_i, y_i]^T => $ a group can be represented as $p = [p_1^T, ..., p_n^T]$.

The high-level formation control aims to guide all SVs from their initial positions to a desired configuration, connecting them internally through constant relative positions $p_l^* - p_{h(l,h) \in \varepsilon}^*$

2.2. Graph theory and control objective (2)

Control objective

The objectives of this work are more complex compared to displacement-based formation control tasks and trajectory tracking control

- This work's objectives encompass both high-level formation tracking and low-level trajectory tracking for each SV agent
- Formation control schemes typically address the kinematic models of mobile robot agents for tracking a desired geometric pattern
- The RL-based optimal control3 focuses on individual SV agent control
- The displacement-based formation controller for multiple SVs deals with interactions between agent pairs

3. Formation Control Strategy
3.1. Nonholonomic constraint in robotic systems (1)

Nonholonomic constraint:

$$\begin{cases} \dot{\bar{x}}_i = \bar{v}_i cos\bar{\psi}_i, \\ \dot{\bar{y}}_i = \bar{v}_i sin\bar{\psi}_i, \\ \dot{\bar{\psi}}_i = \bar{\omega}_i, \end{cases} \quad (1)$$

Remark

- the "bar" symbols denote the conceptual variables associated with the high-level formation controller
- $\dot{\bar{x}}_i sin\bar{\psi}_i - \dot{\bar{y}}_i cos\bar{\psi}_i = 0$
- There exists $\bar{h}_i(\bar{x}_i, \bar{y}_i, \bar{\psi}_i) = 0$

3.2. High-level displacement-based formation control design (1)

A Lyapunov function candidate

$$V = \frac{1}{4} \sum_{l \in S} \sum_{h \in K_l} ||(\bar{p}_l - \bar{p}_h) - (\bar{p}_l{}^* - \bar{p}_h{}^*)||^2 \tag{2}$$

To leverage the negative definiteness of $\frac{d}{dt}V(\bar{e})$, where $\bar{e}(\bar{p})$ is the synchronization error vector, the conventional gradient control law is adapted as follows:

$$\begin{cases} \dot{\bar{p}}_j = h_j h_j^T f_j, \\ \dot{h}_j = (I - h_j h_j^T) f_j, j \in S \end{cases} \tag{3}$$

3.2. High-level displacement-based formation control design (2)

Based on (3), the high-level displacement-based formation control protocol can be implemented for each SV:

$$\begin{cases} \dot{\bar{x}}_j = \bar{v}_j cos\bar{\psi}_j, \\ \dot{\bar{y}}_j = \bar{v}_j sin\bar{\psi}_j, \\ \bar{v}_j = [cos\bar{\psi}_j, sin\bar{\psi}_j](-(\mathcal{L} \otimes I)(\bar{p}_j - \bar{p}_j{}^*)), \\ \bar{\omega}_j = [-sin\bar{\psi}_j, cos\bar{\psi}_j](-(\mathcal{L} \otimes I)(\bar{p}_j - \bar{p}_j{}^*)), \end{cases} \tag{4}$$

Remark

- In contrast to prior work, which primarily concentrates on a formation control structure integrated with position loops, our proposed formation control scheme distinguishes between the high-level formation control protocol and the low-level dynamic control for each agent
- $\frac{d}{dt} V = -\sum_{j \in S} f_j^T h_j h_j^T f_j \leq 0$ when examined along the dynamic

3.2. High-level displacement-based formation control design (3)

Deprived low-level tracking references for each SV

$$\begin{cases} \dot{x}_{di} = \bar{v}_i cos\psi_i, \\ \dot{y}_{di} = \bar{v}_i sin\psi_i, \\ \dot{\psi}_{di} = \bar{\omega}_i, \end{cases} \quad (5)$$

Integrating these derivatives at each time step yields the tracking references $\eta_{di} = [x_{di}, y_{di}, \psi_{di}]^T$ for our multiple SV systems.

3.3. Low-level RL-based control design for each SV (1)

The low-level tracking controller comprises 2 components: $\tau_i = u_i + \tau_{di}$

- a RL policy for a transformed autonomous model u_i
- a model-based component τ_{di} (related to η_{di} and the mathematical model of each SV) where

$$\tau_{di} = M(\eta_i)\frac{d}{dt}v_{di} + C(v_{di})v_{di} + D(v_{di})v_{di} + g(\eta_i) \quad (6)$$

$$\begin{cases} \dot{\eta}_i = J'(\eta_i)v_i \\ v_{di}(t) = J'^{-1}(\eta_i)(\frac{d\eta_{di}}{dt} - \beta_{\eta i}z_{\eta i}), \\ z_{\eta i} = \eta_{di} - \eta_i, \\ z_{vi} = v_i - v_{di}, \end{cases} \quad (7)$$

$\beta_{\eta i}$ is a positive definite matrix

3.3. Low-level RL-based control design for each SV (2)

Extend the tracking error model of each SV

$$\frac{d}{dt}X_i = \begin{bmatrix} -M^-1l(z_{vi} + v_{di}(z_{\eta i}, \eta_{di})) + M^-1l(v_{di}(z_{\eta i}, \eta_{di})) \\ J'(z_{\eta i} + \eta_{di})z_{vi} - \beta_{\eta i} z_{\eta i} \\ h_1(\eta_{di}) \end{bmatrix} + \begin{bmatrix} M^-1 \\ 0 \\ 0 \end{bmatrix} u_i \tag{8}$$

$$=> \frac{d}{dt}X_i = C_i(X_i) + D_i(X_i)u \tag{9}$$

where:

- $X_i = [z_{vi}^T, z_{\eta i}^T, \eta_{di}^T]^T$
- $l(y) = C(y)y + D(y)y$

3.3. Low-level RL-based control design for each SV (3)

Cost function

We introduce an Optimal control scheme to minimize this infinite horizon integral cost function:

$$J_i(X_i, u_i) = \int_0^\infty h_i(X_i(\tau), u_i(\tau))d\tau = \int_0^\infty \left(X_i^T Q_i' X_i + u_i^T R_i u_i\right) d\tau \qquad (10)$$

where $Q'_i = Q'^T_i > 0 \left(\in \mathbb{R}^{9\times 9}\right)$ and $R_i = R_i^T > 0 \left(\in \mathbb{R}^{3\times 3}\right)$.

HJB equation:

$$H\left(X_i, u_i^*, \frac{\partial V_i^*}{\partial X_i}\right) = r(X_i(\tau), u_i^*(\tau)) + \frac{\partial V_i^*}{\partial X_i}(K_i(X_i) + L_i(X_i)u_i^*) = 0 \qquad (11)$$

3.3. Low-level RL-based control design for each SV (4)

Optimal policy $u_i^*(X_i)$ can be obtained by solving the optimization problem using the Bellman function $V_i^*(X_i)$:

$$\min_{u_i(X_i)\in(\ )} H\left(X_i, u_i, \frac{\partial V_i^*}{\partial X_i}\right) = \left\{ r_i(X_i(\tau), u_i(\tau)) + \frac{\partial V_i^*}{\partial X_i}\left(K_i(X_i) + L_i(X_i)u_i\right)\right\} = 0 \quad (12)$$

Function approximation using neural networks (NN)

We approximate the Bellman function and the optimal controller using a critic NN and an actor NN:

$$\begin{cases} \widehat{V}_i(X_i) &= \widehat{W}_{ci}^T \Psi_i(X_i) \\ \widehat{u}_i(X_i) &= -\frac{1}{2} R^{-1} G_i^T(X_i)\left(\frac{\partial \Psi_i}{\partial x_i}\right)^T \widehat{W}_{ci} \end{cases} \quad (13)$$

3.3. Low-level RL-based control design for each SV (5)

Squared Bellman error as a function of

$$\delta_{hjb,i} = \widehat{H}_i(X_i, \widehat{u}_i, \frac{\partial \widehat{V}_i}{\partial X_i}) - H_i^*(X_i, u_i^*, \frac{\partial V_i^*}{\partial X_i}) \tag{14}$$

$$= \widehat{\mathrm{W}}_{ci}^T \sigma_i(X_i, \widehat{u}_i) + \frac{1}{2} X_i^T Q X_i + \frac{1}{2} \widehat{u}_i^T R \widehat{u}_i \tag{15}$$

where $\sigma_i(X_i, \widehat{u}_i) = \frac{\partial \Psi_i}{\partial X_i}(F_i(X_i) + G_i(X_i)\widehat{u}_i)$ is the regression vector of critic part

3.3. Low-level RL-based control design for each SV (6)

We minimize the squared Bellman error by the following update rules:

Training law for the critic weights

$$\frac{d}{dt}\widehat{W}_{ci} = -k_{ci}\lambda \frac{\sigma_i}{1 + v_i \sigma_i^T \lambda_i \sigma_i} \delta_{hjb,i} \tag{16}$$

$\lambda_i(t) \in \mathbb{R}^{N \times N}$ is a symmetric matrix:

$$\frac{d}{dt}\lambda_i = -k_{ci}\lambda_i \frac{\sigma_i \sigma_i^T}{1 + v_i \sigma_i \lambda_i \sigma_i^T} \lambda_i; \quad \lambda_i(t_s^+) = \lambda_i(0) = \varphi_{0i} I \tag{17}$$

Training law for the actor weights

$$\frac{d}{dt}\widehat{W}_{ai} = -\frac{k_{a1}}{\sqrt{1+\sigma_i^T \sigma_i}} \frac{\partial \Psi_i}{\partial X_i} G_i R^{-1} G_i^T \frac{\partial \Psi_i}{\partial X_i}^T (\widehat{W}_{ai} - \widehat{W}_{ci}) \delta_{hjb,i} - k_{a2}(\widehat{W}_{ai} - \widehat{W}_{ci})$$

4. Results

4.1. The parameters of model and control scheme (1)

$$M = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 19 & 0.72 \\ 0 & 0.72 & 2.7 \end{bmatrix}$$

$$C(v) = \begin{bmatrix} 0 & 0 & -19v_y - 0.72v_z \\ 0 & 0 & 20v_x \\ 19v_y + 0.72v_z & -20v_x & 0 \end{bmatrix}$$

$$D(v) = \begin{bmatrix} 0.72 + 1.3|v_x| + 5.8v_x^2 & 0 & 0 \\ 0 & 0.86 + 36|v_y| + 3|v_z| & -0.1 - 2|v_y| + 2|v_z| \\ 0 & -0.1 - 5|v_y| + 3|v_z| & 6 + 4|v_y| + 4|v_z| \end{bmatrix}$$

The chosen smooth activation function $\Psi(X)$:

$\Psi(X) =$
$[X_1^2, X_1X_2, X_1, X_3, X_2^2, X_2X_3, X_3^2, X_1^2X_7^2, X_2^2X_8^2, X_3^2X_9^2, X_1^2X_4^2, X_2^2X_5^2, X_3^2X_6^2]^T$

4.2. Multi-agent formation controller verification
4.2.1. Flower-shaped formation (1)

High-level controller scheme

- The graph Laplacian matrix:

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

- The agents will move in a flower-shaped formation if an appropriate vector such as $a^* = [25, 0, 0, 0, 12.5, 22, 12.5, 22]^T$ is added to (4):

$$\begin{cases} \bar{v}_j = [cos\bar{\psi}_j, sin\bar{\psi}_j](-(\mathcal{L} \otimes I)(\bar{p}_j - \bar{p}_j{}^* - a^*)) \\ \bar{\omega}_j = [-sin\bar{\psi}_j, cos\bar{\psi}_j](-(\mathcal{L} \otimes I)(\bar{p}_j - \bar{p}_j{}^* - a^*)) \end{cases} \tag{18}$$

4.2.1. Flower-shaped formation (2)

The formation of surface vehicles follows a straight line

$\eta_d(t) = [t, t, \pi/4]^T$

Initial states of four agents (i=1,2,3,4)

- $\eta_i(0) = [0, 0, 0]^T$
- $v_i(0) = [0, 0, 0]^T$

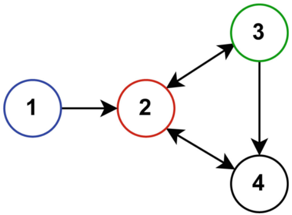

Figure: Communication graph of four agents

4.2.1. Flower-shaped formation (3)

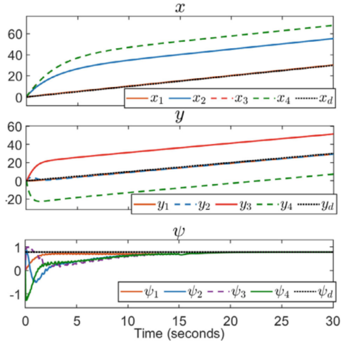

Figure: Tracking trajectories of four agents

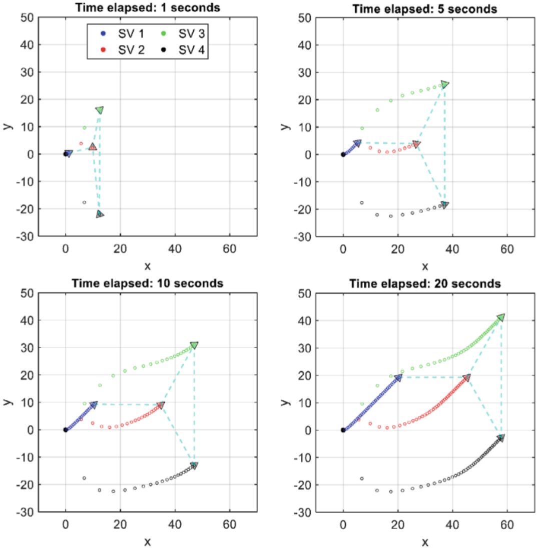

Figure: Flower-shaped formation illustration of

4.2.2. Square formation (1)

High-level controller scheme

- The graph Laplacian matrix:

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

- The agents will move in a square formation if an appropriate vector such as $a^* = [0, 0, d, -d, 0, -d]^T$ is added to (4), where d is the length of the square which is set to be 25 m

4.2.2. Square formation (2)

The formation of surface vehicles follows a straight line

$\eta_d(t) = [12sin(0.2t), -12cos(0.2t), 0.2t + \pi/2]^T$

Initial states of four agents

- $\eta_1(0) = [0, 0, 0]^T$
- $\eta_2(0) = [20, 0, 0]^T$
- $\eta_3(0) = [10, 0, 0]^T$
- $\eta_4(0) = [-10, 0, 0]^T$
- $v_i(0) = [0, 0, 0]^T, i = 1, 2, 3, 4$

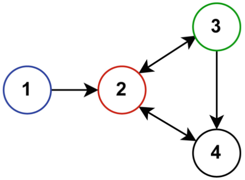

Figure: Communication graph of four agents

4.2.2. Square formation (3)

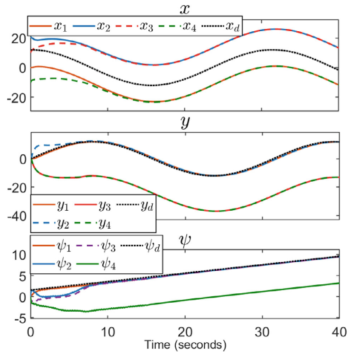

Figure: Tracking trajectories of four agents

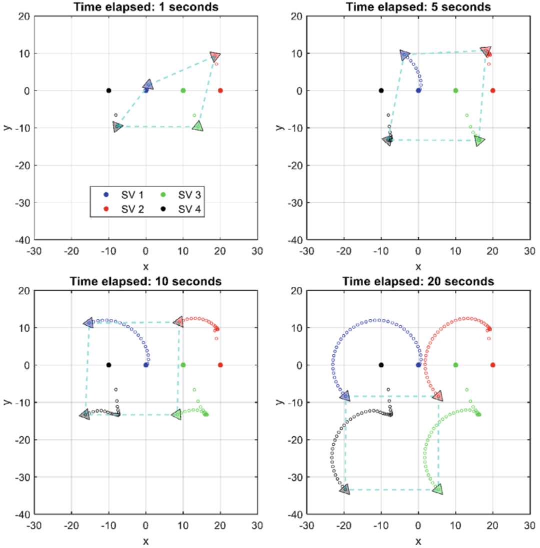

Figure: Square formation illustration of four

4.2.3. Diamond formation with more agents (1)

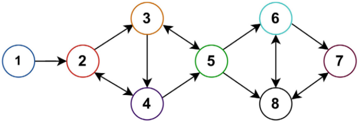

Figure: Tracking trajectories of four agents

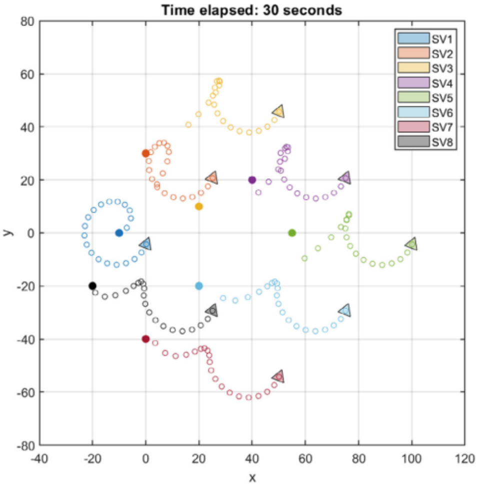

Figure: Square formation illustration of four agents at different timestamps

4.3. RL-based tracking controller verification and comparison

4.3.1. The Convergence of weights (1)

- the weights remain virtually unchanged after just t = 15 s
- There are minor weight variations at t = 15 s due to the cessation of artificial probing noise, but overall, the convergence to an optimal policy is evident

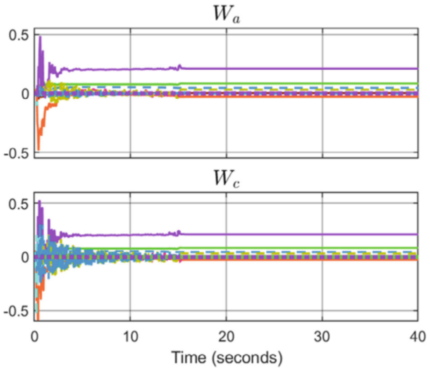

Figure: The convergence of actor and critic weights for agent 1

4.3.2. Advantages of the RL-based method compared to a non-RL policy (1)

The baseline method

- Keep the same outer loop formation control algorithm
- In the lower-level controller,the RL control input is deactivated, while the kinematic and feed-forward design remain unchanged

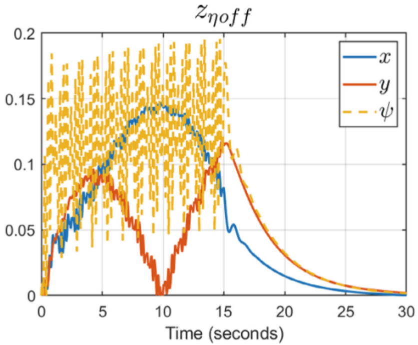

Figure: Trajectory tracking error without RL policy.

4.3.2. Advantages of the RL-based method compared to a non-RL policy

Cost function comparison

- The metric is formulated as follows:

$$J_\Sigma = \int_0^T \left(\eta_i^T Q \eta_i + \tau_i^T R \tau_i\right) dt \tag{19}$$

- The cumulative cost with RL is consistently smaller than that without RL.

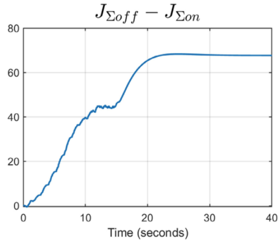

Figure: Trajectory tracking error without RL policy.

Conclusion

Development direction

- The authors plan to conduct experimental validation and extend the low-level tracking controller with model-free RL algorithms that do not necessarily require complete system dynamics.
- Direct implementation of RL algorithms to solve multi-agent control problems in nonlinear systems with uncertainty and disturbance is considered as a feasible approach for further research.


Thank you