

# The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method

Yuanchang Liu\*, Richard Bucknall

*Department of Mechanical Engineering, University College London,  
Torrington Place, London WC1E 7JE, UK*

---

## Abstract

By deploying multiple USVs as a formation fleet, benefits such as wide mission area, improved system autonomy and increased fault-tolerant resilience can be achieved. To efficiently and effectively navigate the USV formation, path planning algorithms are required to generate optimal trajectories and provide practical collision avoidance manoeuvres. In particular, as the USV is underactuated and is restricted by various motion constraints, this paper has presented a new algorithm named the ‘angle-guidance Fast Marching Square’ (AFMS), to make the generated path compliant with vehicle's dynamics and orientation restrictions. Based upon the AFMS, a formation path planning algorithm has been proposed to guide the USVs safely navigating in a cluttered environment. In addition, the formation forming problem has been specifically investigated with the algorithm being developed to make the USVs capable of forming the desired shape by following the trajectories from random initial configurations (positions and orientations). In order to eliminate the potential collision risks occurring on the route, a novel priority scheme based upon the distance to the closest point of approaching (DCPA) has also been proposed and developed. Algorithms have been validated on the computer-based simulations and are proven to work effectively in different environments.

---

\*Corresponding author. Tel: +44 (0)20 3108 9495;  
*Email address:* `yuanchang.liu.10@ucl.ac.uk` (Yuanchang Liu)

*Keywords:* USV formation, path planning, USV heading, angle-guidance fast marching square

---

## 1. Introduction

With the advance of the autonomy technology as well as the drive of demand, there has been an increasing development of unmanned surface vehicles (USVs) in recent decade. The applications of USVs include military utilisations such as sea patrol (Bertaska et al., 2015) and coastal guarding (Manley, 2008), and civilian or scientific deployments such as environmental monitoring (Sharma et al., 2014) and bathymetric survey (Kitts et al., 2012). However, as mentioned in Campbell et al. (2012), most existing USV platforms are restricted by the problem of semi-autonomy, which prevents the USV from performing complex tasks requiring long endurance times. To overcome this, a promising strategy is to employ multiple USVs in formation fleet to allow cooperative operations. With sufficient monitoring and support amongst the vehicles, the USV formation would be able to exploit wide mission area, and have improved system robustness and increased fault-tolerant resilience (Liu and Bucknall, 2015).

A large number of studies have investigated the USV formation control, of which the primary aim is to design a series of low-level controllers for each vehicle's actuation system to maintain or track the desired positions and orientations relative to a defined reference point (Shojaei, 2015). Favoured control strategies include the leader-follower control (Chen et al., 2009; Peng et al., 2013), the virtual structure approach (Lewis and Tan, 1997; Mehrjerdi et al., 2011) and the behaviour-based formation control (Balch and Arkin, 1998; Monteiro and Bicho, 2002). By comparing and analysing the literature, it is found that not only can formation maintenance be achieved, but in some work high-level autonomous behaviour such as the collision avoidance can also be fulfilled by adding additional controllers (Mastellone et al., 2008; Mahacek et al., 2012).

However, the design of appropriate controllers, especially for USV formation, requires significantly more computational resources, which might not be an ideal option for the real-time navigation where fast response times are always needed (Lee et al., 2015). An alternative to this is to solve the problem from

the formation path planning perspective. Efficient path planning algorithms are capable of generating optimal way-points or paths within short time periods, and through employing simple trajectory controllers the formation performance can be enormously improved.

Hao and Agrawal (2005) first proposed a formation path planning framework for unmanned ground vehicles in a dynamic environment by using the A\* algorithm. Garrido et al. (2011) and Gómez et al. (2013) then employed this framework for mobile robots formation, and replaced the A\* by the fast marching method (FMM) to generate better trajectories. Liu and Bucknall (2015) improved these two works for USV formations application and specifically addressed the problem of collision avoidance with moving vessels in a practical maritime environment. However, there are two limitations associated with these researches, especially for the USV formation implementation. One limitation is the ignoring the vehicle's dynamics characteristics. The A\* algorithm can only ensure the shortest path to be generated and requires additional path smoother to improve the features of the path to make it more feasible for a practical controller to follow. The FMM algorithm provides better performance in terms of the path smoothness and continuity than the A\* does, however it still does not fully address the dynamic constraints, especially without considering the vehicle's heading angles. The details of this problem will be explained further in the next section.

The second limitation when the algorithm being designed is the formation forming requirement, which is specially important for USVs. In the aforementioned works, the assumption has been made that the formation starts the mission with the desired shape already been formed. However, this is not a reasonable assumption as the USVs are normally launched offshore and are prone to drift caused by currents or waves, which makes it impractical to have the formation manually oriented in the correct shape and heading before the start of the mission. A more effective method is to make the USVs start from their random positions and move into the formation *en route*. Hence, it is worth breaking down the USV formation path planning into two separate stages, i.e. the Formation Forming and the Formation Path Planning.

In this work a practical solution is offered to overcome these two limitations. A new angle guidance path planning algorithm based on the FMM, named angle-guidance fast marching square (AMFS), has been designed and developed specifically to make the generated path more compliant with the USV's

dynamics. The AFMS is then used as the base algorithm for the USV formation path planning. Two individual algorithms are developed to solve the formation forming and formation path planning problems, respectively. By using the developed algorithms, the USV formation is better able to form the required shape safely and efficiently by following the generated trajectories. Some favourable formation behaviours such as the adaptive and flexible shape formation can be achieved to assure the navigation safety, which is the primary requirement in the maritime environment.

It also should be noted that while the formation shape is being formed, mutual collision of the USVs is an issue that needs to be addressed. Therefore, in this paper, a novel priority scheme based upon the distance to the closest point of approaching (DCPA) has been proposed to help with the elimination of the conflicts while the USVs are travelling. Compared with other priority algorithms such as the velocity based approach (Liu et al., 2014) and the environment based approach (Clark et al., 2002), the DCPA one is more suitable for marine applications as the DCPA is able to fast assess the collision risk and help to provide the most appropriate collision avoidance decisions (Tam and Bucknall, 2010a).

The rest of the paper is organised as follows. Section 2 reviews the state-of-the-art algorithms in terms of the USV path planning. Section 3 specifically introduces the FMM algorithm as well as its improved version, the fast marching square (FMS) algorithm, which is able to increase the overall path safety. Section 4 describes the proposed AFMS method and compares it with the conventional FMM. Section 5 introduces the developed formation forming and formation path planning algorithms. The proposed algorithms and methods are verified by simulations in Section 6. Section 7 concludes the paper and discusses the future work.

## **2. Review of USV path planning algorithms**

As regards USV path planning algorithm development, there has been a number of research methodologies. In the early stages, the evolutionary algorithm (EA) has been largely adopted to search for the feasible navigation route. Smierzchalski (1999) first employed the genetic algorithm to generate an optimised route for marine vessel and developed a solution for collision

avoidance with moving ships by adopting ship domain areas. A similar approach has also been used in Tam and Bucknall (2010b) but with special emphasis on generating practical evasive manoeuvres adhering to COLREGs.<sup>1</sup> Tsou and Hsueh (2010) implemented the ant colony algorithm to design a decision-making system which can assist vessels navigating in the maritime environment and also obey the COLREGs. However, the major drawbacks of the EA are the inconsistency and incompleteness of the searching result, which hinders its adoption for practical purposes as the properties of the algorithm outputs cannot be guaranteed (Tam and Bucknall, 2013).

Hence, in recent years, the main effort has focused on using a deterministic search algorithm, such as the grid-based and artificial potential field (APF) methods, for USV path planning. Xue et al. (2011) improved the APF to provide a safe collision free path in congested environments with multiple vessels to be avoided. Naeem et al. (2012) designed a COLREGs compliant path planner by using a modified A\* algorithm. A path trimmer was integrated into the A\* algorithm to smooth the generated path making it more feasible for the vehicle to follow. Tam and Bucknall (2013) proposed a cooperative path planning algorithm for USV with the main aim focusing on increasing the practicability and the completeness of the algorithm. Improved consistency and completeness can be seen through these algorithms to assure the same navigation path can be generated as long as the planning environment does not change. However, these works consider the path planning problem from the conventional perspectives, i.e. generating the path with the least distance cost while avoiding obstacles. It still needs to improve the path's quality, especially to make it compliant with USV's dynamics characteristics.

Kim et al. (2014) proposed a novel angular rate constrained algorithm based on the *Theta\**, which is a grid-based method similar to A\*. The main concept behind this work is to redefine the feasible grid points in the planning space by referring to the turning rate of the USV, i.e. any points that are beyond the maximum turning range of the USV will be removed and not considered when generating the path. However, because additional functions were required in this algorithm, the overall computational time was increased, which takes

---

<sup>1</sup>COLREGs stands for The International Regulations for Preventing Collisions at Sea 1972, which should be followed by ships and other vessels at sea to prevent collisions between two or more vessels.

more than double the time of the conventional *Theta\**. In addition, because the algorithm still belongs to the grid-based method, the continuity of the path is highly dependant on the resolution of the planning space, which in turn affects the path's performance if a less rasterised space is used.

Recently, the fast marching method (FMM) based path planning algorithm becomes a new approach to generate smooth and continuous trajectory. The FMM shares the similar concept to the AFM of searching for the path based on the potential field. However, differing from the conventional way of combining all fields to generate the total potential field; the FMM produces the potential field by simulating the propagation of an electromagnetic wave, and the generated potential field does not suffer from the local minima problem, which is the main drawback of the APF (Garrido et al., 2008).

The FMM calculates the path by using the gradient descent method over the potential field from the end point to the start point, and one of the most appealing features of the generated path is the guaranteed smoothness given that the generated potential field does not have any discontinuity (Gómez et al., 2013). No more path smoother is required to process the path, and the path can be easily executed by a proper controller in practical applications (Álvarez et al., 2015). In the meantime, the FMM is fast in computational speed, which further promotes its utilisation in real-time navigation such as on autonomous underwater vehicle (Petres et al., 2007), unmanned aerial vehicle (Garrido et al., 2013) and even the USV platforms (Liu et al., 2015a; Xu et al., 2013)

However, the implementation of the FMM to USVs needs more considerations. A crucial factor is that USV, as a nonholonomic system, is underactuated during most of operation time making it have low manoeuvrability and motion flexibility. Even though the smooth trajectory provided by the FMM is able to be followed by the vehicle and is compliant with vehicle's dynamics, the heading angle of the USV, especially during the launching stage, should also be taken into account. For example, from our previous experiments carried out on a true USV platform, *Springer* USV, it has been found that if a 'perfect' initial heading can be set up when the USV started its mission, good performance can be achieved by the vehicle with the actual taken route staying relatively close to the planned path. However, if the USV had a 'bad' heading angle, i.e. pointing roughly opposite to the direction of the desired path, then a large offset will occur during the initial stages as the vehicle

is required to keep adjusting its direction (Rui, 2014). Such situations may impose additional risks for the USV as the vehicle may collide with obstacles as a result of the path generated exceeding the craft's dynamic constraints, and it is thus required to modify the path planning algorithm to consider the orientation of the USV at the start of the transit and design a feasible route constrained to within the vehicle's turning capability.

Fig. 1 provides an illustration of how the path should be planned and altered according to the USV's turning capability at the launch stage. In this example, the USV is travelling with the heading angle towards northwest. The turning capability of the USV is represented by the cone shaped turning range. The path in green is calculated without consideration of the turning capability and exceeds the turning range of the USV making the path potentially untrackable. However, the path in black has taken the USV's turning capability into account. It can be observed that even though the black line is generated at greater distance cost (the green line is shorter), it is located within the USV's turning range constraint and better ensures the USV is able to effectively adjust its heading angle and follow the path thus defined.

In this paper, to specifically solve vehicle heading angle problem, the 'angle-guidance fast marching square' (AFMS) algorithm that has therefore been designed and developed. The AFMS is able to generate a path according to the vehicle's actual heading angle by using a novel 'guidance range' concept. The 'guidance range' is built upon the vehicle's specific dynamics and actual heading, both at outset and during the transit. The specific AFMS algorithm structure will be given in Section 4.

### 3. The FMM based path planning algorithm

#### 3.1. The fast marching method (FMM)

The FMM was first proposed by J. Sethian in 1996 to iteratively solve the Eikonal equation to simulate the propagation of interface (Sethian, 1996). The Eikonal equation has the form as:

$$|\nabla(T(x, y))|V(x, y) = 1 \quad (1)$$

where  $T(x, y)$  is the interface arrival time at point  $(x, y)$  and  $V(x, y)$  is the interface propagating speed. Eq.1 belongs to the partial differential equation (PDE) and its numerical solution can be obtained via the upwind differential method. In Fig. 2, suppose  $(x, y)$  is the point that  $T(x, y)$  needs to be solved. The neighbour of  $(x, y)$  is a point set containing four elements  $(x + \Delta x, y)$ ,  $(x - \Delta x, y)$ ,  $(x, y + \Delta y)$ ,  $(x, y - \Delta y)$ .  $T(x, y)$  can be obtained as:

$$T_1 = \min(T_{(x-\Delta x, y)}, T_{(x+\Delta x, y)}) \quad (2)$$

$$T_2 = \min(T_{(x, y-\Delta y)}, T_{(x, y+\Delta y)}) \quad (3)$$

$$|\nabla T_{(x, y)}| = \sqrt{\left(\frac{T_{(x, y)} - T_1}{\Delta x}\right)^2 + \left(\frac{T_{(x, y)} - T_2}{\Delta y}\right)^2} \quad (4)$$

$$\left(\frac{T_{(x, y)} - T_1}{\Delta x}\right)^2 + \left(\frac{T_{(x, y)} - T_2}{\Delta y}\right)^2 = \frac{1}{(V_{(x, y)})^2} \quad (5)$$

where  $\Delta x$  and  $\Delta y$  are the grid spacing in the x and y directions. The solution of Eq. 5 is given by

$$T_{(x, y)} = \begin{cases} T_1 + \frac{1}{V_{(x, y)}} & \text{if } T_2 \geq T \geq T_1, \\ T_2 + \frac{1}{V_{(x, y)}} & \text{if } T_1 \geq T \geq T_2, \\ \text{quadratic solution of Eq.5} & \end{cases} \quad (6)$$

The FMM adopts the fundamentals of Dijkstra algorithm; however, instead of employing the classical rectilinear distance metric, it uses the Eq.6 to update the cost function, which measures the distance in Euclidean metric. Therefore, compared with Dijkstra, the result (or the path) provided by the FMM is more continuous. The specific FMM algorithm is described in Algorithm 1 by using an example of simulating an interface propagation process over a grid map. In the initialisation process, the algorithm first assigns all the grid points with the arrival time of infinity. Then, grid points are grouped into three different categories, i.e. the *Far*, *Known* and *Trial* point sets. Such a categorisation method is similar to Dijkstra's and the specific meanings of each group are as follows:



- *Far*: contains grid points with undecided arrival time value ( $T$ ). In the first time step when running the FMM, all grid points except the start points belong to *Far*;
- *Known*: contains grid points with decided arrival time values ( $T$ ). Such values will not be changed when the algorithm is executed;
- *Trial*: contains grid points with calculated arrival time values ( $T$ ); however, values may be changed when the algorithm is running.

During each iteration of the algorithm, the point  $a$  with the smallest  $T$  will be selected from the *Trial* set and added into the *Known* set. It should be noted that such a process is the classic ‘find-min’ operation in computer science, which is considerably time-consuming when large number of points are involved. Therefore, in order to increase the efficiency of the algorithm, an appropriate data structure is required to store the *Trial* set, and the Fibonacci heap data structure is preferred to be used as its *time complexity* of ‘find-min’ operation is  $\mathcal{O}(1)$  (Cormen, 2009). For the point  $a$ , all the neighbour points will be updated by using Eq. 6 to have new arrival time values, and neighbour points located in the *Far* point set will then be moved into the *Trial* point set for next iteration. The algorithm will be terminated until the *Trial* point set is empty.

### 3.2. The FMM based path planning algorithm

The FMM based path planning algorithm is described in Algorithm 2. Consider the planning space ( $\mathbf{M}$ ), where the algorithm is to be applied, has a representation of a binary map and is perfectly rasterised. The algorithm first reads in  $\mathbf{M}$  and calculates its according speed matrix ( $\mathbf{V}$ ). The speed matrix ( $\mathbf{V}$ ) has the same size as  $\mathbf{M}$  and defines the interface propagation speed at each point in  $\mathbf{M}$ . Based on  $\mathbf{V}$ , the FMM is executed to calculate an arrival time matrix  $\mathbf{T}_{FMM}$ , and upon the time matrix  $\mathbf{T}_{FMM}$ , the optimal path is finally searched by applying the gradient descent method from the end point to the start point.

A more intuitive way to interpret FMM based path planning is from the potential field perspective. In Fig. 3a, two round obstacles are located near the centre of the map; while the start and end points are at the northwest and southeast corners respectively. The map is represented by a binary grid

---

**Algorithm 1** *Fast Marching Method* Algorithm

---

**Require:** configuration space ( $\chi$ ), start point ( $p_{start}$ )

- 1: assign all the grid points in  $\chi$  with the cost of Infinity ▷ Initialisation
- 2:  $T(p_{start}) \leftarrow 0$
- 3:  $Far \leftarrow$  all grid points in  $\chi$
- 4:  $Known \leftarrow$  all grid points with known cost
- 5: **for** each adjacent point  $a$  of  $Known$  point **do**
- 6:      $Trial \leftarrow a \cup Trial$
- 7:      $T(a) = costUpdate(a)$  ▷ Using Eq. 6
- 8: **end for**
- 9: **while**  $Trial$  is not empty **do** ▷ Update process
- 10:     $p \leftarrow$  point with the lowest cost in  $Trial$
- 11:    remove  $p$  from  $Trial$
- 12:     $Known \leftarrow p \cup Known$
- 13:    **for** each neighbour point  $a$  of  $p$  **do**
- 14:       $\tilde{T}(a) = costUpdate(a)$  ▷ Using Eq. 6
- 15:      **if**  $\tilde{T}(a) < T(a)$  **then**
- 16:         $T(a) \leftarrow \tilde{T}(a)$
- 17:      **end if**
- 18:      **if**  $a \in Far$  **then**
- 19:        remove  $a$  from  $Far$
- 20:         $Trial \leftarrow a \cup Trial$
- 21:      **end if**
- 22:    **end for**
- 23: **end while**
- 24: **return**  $T$

---

map, where each grid in collision free space has value 1 and grids in obstacle areas have value 0.

The FMM is then applied on such a grid to simulate an interface propagation process. The interface is used to help build up a potential field, whose potential value on each grid point is the local interface arrival time. The interface begins to proceed from the start point on the grid map by taking local grid values to determine propagation speed. When the interface reaches the end point, the potential field ( $\mathbf{T}_{FMM}$ ) is created as shown in Fig. 3b. In the field, the potential value at each point represents local arrival time of the

---

**Algorithm 2** *FMM Path\_Planning* Algorithm

---

**Require:** planning space ( $M$ ), start point ( $p_{start}$ ), end point( $p_{end}$ )

- 1: Calculate speed matrix  $V$  from  $M$
- 2:  $T_{FMM} \leftarrow FMM(V, p_{start})$
- 3:  $path \leftarrow gradientDescent(T_{FMM}, p_{start}, p_{end})$
- 4: **return**  $path$

---

interface, which subsequently indicates local distance to the start point if a constant speed matrix is used. Since the interface begins propagating from the start point, the potential of the start point is therefore the lowest and is equal to zero. Potential values at other points increase as the interface advances and reach the highest value at the end point. Because the interface is not allowed to transmit inside an obstacle area, obstacles' potentials are infinite. Compared with the potential field generated by the APF, the potential field of the FMM has features of global minimum, which avoids local minima problems and increases the completeness of the algorithm. Based on the potential field obtained, the gradient descent method is then applied to find the shortest collision free path by following the gradient of the potential field. The generated path is represented as the red line shown in Fig. 3b.

### 3.3. The fast marching square method

One of the problems associated with path planning by directly using the FMM method is the generated path is too close to obstacles, which can be observed from Fig. 3b. Such a drawback is especially impractical for USVs, because near distance areas around obstacles (mainly islands and coastlines) are usually shallow water, which is not suitable for marine vehicles to navigate. Hence, it is important to keep the planned path a certain distance away from obstacles.

To tackle this problem, Garrido et al. (2007) has proposed a new algorithm named the fast marching square (FMS) method. The basic concept behind the FMS is to apply the conventional FMM algorithm twice but with different purposes. The FMS is represented in Algorithm 3 and explained using the same planning space in Fig. 3a. The FMS first generates a safety potential map ( $\mathbf{M}_s$ ) by applying the FMM to propagate interfaces from all the points in obstacle areas in  $\mathbf{M}$ . The created  $\mathbf{M}_s$  is shown in Fig. 3c, where each

point in the map is assigned with a value, ranging from 0 to 1, representing the shortest distance to obstacles. The further the distance to an obstacle is, the higher the value will be. Such values can be viewed as indices to indicate the safety of local points. Low values represent current locations may be too close to obstacles and consequently may not be safe to proceed; hence USVs should be encouraged to keep travelling in the areas with high index values.

Based on  $M_s$ , the FMM is executed again from the start point to generate the potential field  $T_{FMS}$  (shown in Fig. 3d). It should be noted that the shape of  $T_{FMS}$  is different to that of  $T_{FMM}$  where the potentials near obstacles is always higher than at other places'. Higher potential values around obstacles act as a protecting layer to prevent the path passing too close to obstacles. Finally, based upon the  $T_{FMS}$  the gradient descent method will be used to search for the optimal trajectory, which is shown as the black trajectory in Fig. 3d. It can be observed that compared with the red trajectory (in Fig. 3b) increased safety can be achieved.

---

**Algorithm 3** *Fast\_Marching\_Square* Algorithm

---

**Require:** planning space ( $M$ ), start point ( $p_{start}$ ), end point( $p_{end}$ )

- 1: **for** each point  $a$  in obstacle area in  $M$  **do**
- 2:      $obstaclePoints \leftarrow obstaclePoints + a$
- 3: **end for**
- 4:  $M_s \leftarrow FMM(M, obstaclePoints)$
- 5:  $T_{FMS} \leftarrow FMM(M_s, p_{start})$
- 6:  $path \leftarrow gradientDescent(T_{FMS}, p_{start}, p_{end})$
- 7: **return**  $path$

---

## 4. The angle-guidance FMS method

### 4.1. The kinematic motion of USV

Based on the concept represented in Fig. 1, the FMS algorithm has been improved in this paper to a new method named the ‘angle-guidance FMS’ (AFMS) algorithm with the specific application for USV path planning. Before introducing the AFMS algorithm, the basic motion equations of USV's is explained. Consider  $\langle e \rangle$  is the inertial coordinate frame and  $\langle b \rangle$  is the body fixed coordinate frame. Let the state of the USV relative to  $\langle e \rangle$  is denoted

as  $\boldsymbol{\eta} = [x \ y \ \alpha]^T$ , where  $x$  and  $y$  represent the position coordinates of the USV in the planning space and  $\alpha$  is the USV heading angle. The surge and yaw speed of the USV is expressed with respect to  $\langle b \rangle$  and has the form of  $\boldsymbol{v} = [u \ v \ r]^T$ , where  $u$  and  $v$  are surge and sway speed and  $r$  is the yaw rate. The kinematic motion of the USV can therefore be written as:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}\boldsymbol{v} \quad (7)$$

where

$$\boldsymbol{J} = \boldsymbol{J}(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Normally, the USV is underactuated making the vehicle can only be controlled in the surge and yaw motions. To facilitate the operation, vehicle's surge speed  $u$  remains constant as (Annamalai et al., 2015; Bertaska et al., 2015):

$$u = C \quad (9)$$

where  $C$  is a constant. In addition, the USV has limited turning capabilities. Hence, the yaw rate is subject to the constraints of yaw boundary ( $r_{max}$ ) as:

$$|r| \leq r_{max} \quad (10)$$

#### 4.2. The AFMS path planning algorithm

Based on the kinematic motion of the USV, the AFMS is developed with its pseudocode shown in Algorithm 4. It uses the FMS as the base algorithm, and in order to make the generated path compliant with USV's motion constraints, the core of the AFMS is to create a 'guidance range' (GR) upon the planning space ( $\boldsymbol{M}$ ) (shown as Line 1 in Algorithm 4). The shape of the GR is constructed as graphically shown in Fig. 4. It consists of two different sectors, i.e. the *Turning range sector* in white, and the *Obstacle sector* in shade. The 'Turning range sector' is the same as the turning range shown in

Fig. 1. Its dimension is controlled by three parameters, i.e. range distance ( $d$ ), heading angle ( $\alpha$ ) and turning angle ( $\theta$ ).

The turning angle ( $\theta$ ) parameter is calculated according to vehicle's yaw constraint. Suppose the state of the USV at time step  $t$  is denoted as  $\boldsymbol{\eta}_t = [x_t \ y_t \ \alpha_t]^T$ . When the USV is making turning with its maximum yaw rate, its motion can be expressed through two velocities, i.e. the rotational velocity ( $r_{max}$ ) and the translational velocity ( $u$ ). Therefore, according to Thrun et al. (2005), the state of the USV at next time step  $t + \Delta T$  ( $[x_{t+\Delta T} \ y_{t+\Delta T} \ \alpha_{t+\Delta T}]^T$ ) can be calculated as:

$$\begin{pmatrix} x_{t+\Delta T} \\ y_{t+\Delta T} \\ \alpha_{t+\Delta T} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \alpha_t \end{pmatrix} + \frac{u}{r_{max}} \begin{pmatrix} -\sin \alpha_t + \sin(\alpha_t + r_{max}\Delta T) \\ \cos \alpha_t - \cos(\alpha_t + r_{max}\Delta T) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ r_{max}\Delta T \end{pmatrix} \quad (11)$$

where  $\Delta T$  is the time step between  $t$  and  $t + \Delta T$ . The turning angle ( $\theta$ ) is defined as the difference of the heading angle of the USV between time step  $t$  and  $t + \Delta T$  and can thus be calculated as:

$$\theta = \alpha_{t+\Delta T} - \alpha_t = r_{max}\Delta T \quad (12)$$

The calculated value of  $\theta$  represents the maximum turning angle that can be made by the USV during  $\Delta T$ , e.g.  $\theta = 30^\circ$  represents the USV can only make a starboard or port side turn of up to 30 degrees for one operation. It should be noted that this parameter should be adjusted according to specific USV dynamics in real operation.

The range distance ( $d$ ) is the radius of the cone shape which is able to control the influence range affecting the path and is related to the surge speed of the USV as:

$$d = \begin{cases} d_{min} & \text{if } u < u_{permit}, \\ u * rangeScalar & \text{otherwise.} \end{cases} \quad (13)$$

where  $d_{min}$  is the predefined minimum distance used in the case when the USV has low transverse speed such that enough range space will be created

to facilitate the USV making the required turn. It also can be observed from Eq.13 that the GR dimension is increasing proportionally with the USV speed, which makes the generated path better located in the GR to accommodate a USV's motions when the USV is travelling at high speed. The range distance ( $d$ ) is also controlled by the parameter *rangeScalar*, which is primarily used to regulate the size of the range to prevent the algorithm from generating too large obstacle area such that the target point will be blocked, especially in narrow passages.

Finally, the heading angle ( $\alpha$ ) represents USV's current heading angle and determines the direction of the GR. The weighting values inside the *Turning range sector* are assigned to remain the same as they are in the planning space  $\mathbf{M}$ . In addition, since it is desired that the path should be located within the *Turning range sector* making the *Obstacle sector* act like an obstacle, grid values of the *Obstacle sector* are assigned as 0.

---

**Algorithm 4** *Angle-guidance FMS Algorithm*

---

**Require:** planning space ( $M$ ), start point ( $p_{start}$ ), end point ( $p_{end}$ ), heading angle ( $\alpha$ ), turning angle ( $\theta$ ), range distance ( $d$ )

- 1:  $range \leftarrow guidanceRange(M, d, \alpha, p_{start}, \theta)$
- 2: **for** each point  $p$  in  $range.obstacleSector$  **do**
- 3:      $M(p) = 0$
- 4: **end for**
- 5:  $path \leftarrow FMS(M, p_{start}, p_{end})$
- 6: **return**  $path$

---

However, it should be noted that the process of searching for the points in *Turning range sector* or *Obstacle sector* might be time-consuming as the planning space normally has larger size than the GR, and it is not necessary to consider every points in the  $\mathbf{M}$  but just around the GR area. Therefore, the region of the interest (ROI), which is a common concept in the computer vision research, has been used to fast identify points on the  $\mathbf{M}$ . The ROI is a *binary mask*, which is a binary image having the ROI region equal to 1 and other areas equal to 0. As shown in Fig. 5b, the *binary mask* has the same size as  $\mathbf{M}$ , and a circular ROI has been created with its centre point located at the USV's position. The radius of the ROI is slightly larger than GR's range distance to ensure that each point inside the GR can be identified. The mask will then be applied on  $\mathbf{M}$  to filter out unnecessary areas, and only the

points have the value of 1 ( $\mathbf{M}(p) = 1$ ) will be considered, which reduces the searching area from the whole space to the ROI.

Once the corresponding points in the GR has been identified, by adding the GR to the original planning space, a new planning space ( $\mathbf{M}$ ) can be generated with the consideration of vehicle's heading and yaw rate constraints, upon which the FMS algorithm is employed to search for the path.

In Fig. 6, paths generated by the AFMS and the FMS are compared with the AFMS's trajectory represented in magenta and the FMS's in green. In Fig. 6a, the USV has a heading angle of  $10^\circ$  with the turning angle of  $30^\circ$  and the range distance of 15 m. To better explain the results, the zero degree has been defined in east direction in this case. The trajectory generated by the AFMS follows the USV's heading making it have a slight offset with the path calculated from the FMS. The associated potential field is displayed in Fig. 6b with the GR area plotted at the USV's start point. A more evident case is presented in Figs 6c and 6d, where the USV has a heading of  $180^\circ$  and the range distance of 8 m. The FMS's path, without considering USV's dynamics, avoids the obstacle from right side and forms a large turning angle at the start point, which the USV is incapable of tracking. In contrast, the AFMS's path stays alongside the direction of USV's heading by following the guidance of the GR, and it eventually avoids the obstacle from the left side.

Note that the AFMS algorithm is not the first path planning algorithm developed considering vehicle's dynamics; however, some key features distinguish the AFMS from other work. Although Kim et al. (2014) have integrated vehicle's turning constraint with the *Theta\** algorithm as well as Liu et al. (2014) did for the rapidly exploring random tree (RRT) algorithm, both of these works are done in a 'time-costly manner'. For example, in Kim et al. (2014), vehicle dynamics are treated as an additional cost during each iteration of the algorithm. Despite the generated path being compliant with turning constraint, the algorithm's computational speed has also been potentially decreased. In contrast, the AFMS is developed based on the FMS, which, as earlier stated, is able to generate a continues trajectory which is already dynamics compliant. However, the FMS does not take the initial heading angle into account because it is not an important issue in mobile robot applications but does have influence on USVs, therefore the AFMS provides a solution for this by adding a GR area. The AFMS does not change the algorithm structure internally thereby keeping it as efficient as



the FMS. The path generated by the AFMS can be retained within the GR to ‘completely’ follow the vehicle’s dynamics while largely maintaining the continuity of the trajectory.

## 5. USV formation path planning algorithm based on the AFMS

### 5.1. Path planning algorithms for USV formation forming

To solve the problem of the formation forming (FF) problem for USVs, an algorithm based on the AFMS has been proposed with the algorithm flow chart shown in Fig. 7. Initial heading angles of each USV are first read in as the algorithm inputs. Leader-follower approach is used here to design the formation shape with one USV from the formation or a virtual leader USV been selected to guide the motion of the formation. Then, the next step is to obtain the target position of the leader USV and calculate the target points for follower USVs by referring to the formation parameters. Based on the assigned target points, the AFMS is then applied on each USV to generate the according trajectory. Trajectories are followed individually by each USV until target points are reached.

#### 5.1.1. Tracking control and conflict elimination

Once trajectories have been generated, trajectory-following (or trajectory-tracking) tasks will be performed by the formation. Since the main focus of this work is on path planning research instead of robust controller design, it is assumed that each USV has a trajectory-following controller such that its individual path is able to be perfectly tracked without error. Such an assumption is reasonable and has been largely used in the work of Álvarez et al. (2015) and Gómez et al. (2013) for formation path planning. However, in the FF problem, the formation has not yet been formed, which raises two important issues that need to be specifically addressed, i.e. distance control and conflict elimination.

Since USVs have random starting positions, and the distance between each USV may be much greater than the desired distance; it would be required that each USV should adjust its own speed in order to reduce distance error, and a piecewise speed control strategy has therefore been proposed. To simplify the problem, all USVs are assumed to have the same desired speed

and the leader USV keeps its speed unchanged for the whole duration of process; whereas followers adjust their speeds by referring to the distance error as:

$$V_{follower} = \begin{cases} V_{leader}, & \text{if } dist(leader, follower) = d_{desired}, \\ \alpha_{speed} * V_{leader}, & \text{if } dist(leader, follower) \neq d_{desired} \end{cases} \quad (14)$$

where  $dist(\bullet)$  is the function calculating the instantaneous distance between the follower and the leader.  $\alpha_{speed}$  is the speed adjusting parameter, which will make a follower change the speed such that the distance error can be reduced.

Another important issue is the potential conflict occurring *en route* as shown in Fig. 8, where two USVs have the possibility of collision at the trajectory crossing point. To effectively eliminate the conflict, a priority based speed control algorithm has been designed and employed. It dynamically allocates priority to each USV involved in the conflict by calculating the distance to the closest point of approach (DCPA). As shown in Fig. 9, USV 1 is travelling with the velocity  $V_1$  and USV 2 has the velocity of  $V_2$ .  $V_{12}$  is the relative velocity of the USV 1 with the respect to USV 2,  $D$  is the distance between two vessels and  $\gamma$  is the angle between the relative motion line and the bearing angle of USV 1. The DCPA can then be calculated as:

$$d_{CPA} = D * \sin(\gamma) \quad (15)$$

Based on the DCPA, the priority of the  $i^{th}$  USV can be defined as:

$$P_i \propto \frac{1}{d_{CPAi}}, \quad i \in N \quad (16)$$

where  $i$  is the index of USV and  $N$  is the total number of USVs having conflict. From Eq.16, it can be deduced that the smaller the DCPA is, the higher the priority. In the situation depicted in Fig. 8, USV 2 has smaller DCPA thereby having higher priority than USV 1. The speed of the vessel is then adjusted according to its priority in such a way that the lower the priority, the slower the speed, which allows the fast elimination of the conflict.

It however should be emphasised that the control strategy proposed in this paper is only a framework. To achieve better performance or results, sophisticated controllers such as the PID controller or the model predictive control (MPC) can be adopted according to specific applications, which is beyond the scope of this research.

## 5.2. Path planning algorithm for USV formation in cluttered environment

This section introduces the development and design of a formation path planning algorithm in a cluttered static environment. The static environment is referred to as a time-invariant environment containing multiple static obstacles such as an island and landmass, which forms a typical complicated maritime environment. Again, the algorithm is designed upon leader-follower formation control approach proposed in Garrido et al. (2011). The leader's target point is the formation's target, which remains unchanged; whereas target points for followers are re-planned during each operation time step according to leader-follower strategy to maintain the formation shape. The specific sub-target generation algorithm will be introduced in next section. It should also be noted that similar to the problems introduces previously, there is no consideration of vehicle's dynamics and headings in Garrido et al. (2011) when the algorithm is developed. Therefore, in this paper, the AFMS algorithm has been integrated with the formation path planning architecture to have a specific USV formation path planning algorithm.

### 5.2.1. Sub-target generation based upon the leader-follower scheme

The sub-target generation principle used in this paper is based on Álvarez et al. (2015), which is able to flexibly generate sub-targets according to environment change and makes the formation deformable. Based upon the leader-follower scheme, each follower's sub-target is re-planned by referring to the desired formation angle ( $\beta$ ) and the formation distance ( $d$ ) between the leader and the follower. As shown in Fig. 10, the formation angle ( $\beta$ ) determines the desired angle between the leader and the follower; whereas the formation distance represents the desired distance that the follower USV wants to maintain with the leader.  $\beta$  and  $d$  can be combined to have the desired formation configuration vector  $\vec{D}$  (including  $\beta$  and  $d$ ), which is represented as:

$$\vec{D} = n \vec{i} + m \vec{j} = d \cos(\beta) \vec{i} + d \sin(\beta) \vec{j} \quad (17)$$

where  $\vec{i}$  is the unit vector with the opposite direction to the leader's heading and  $\vec{j}$  is the unit vector perpendicular to the leader's heading. Given that the desired distance is represented by  $d$ , for a formation with triangular shape, sub-target points can be calculated as:

$$x_{follower1} = x_{leader} - d * n, \quad y_{follower1} = y_{leader} + d * m \quad (18)$$

$$x_{follower2} = x_{leader} - d * n, \quad y_{follower2} = y_{leader} - d * m \quad (19)$$

However, by following Eq. 18 and 19, the formation shape cannot be flexible. Fortunately, the safety map ( $\mathbf{M}_s$ ) generated by the AFMS has good capability for indicating local information. In  $\mathbf{M}_s$  values at each points range from 0 to 1 with 0 representing obstacle space and 1 representing collision-free space. Such a value can be used as a deformation parameter ( $B$ ) to indicate how much the formation needs to change. Hence, the new sub-targets generation equations are:

$$B_1 = \mathbf{M}_s(x_{follower1}, y_{follower1}), \quad B_2 = \mathbf{M}_s(x_{follower2}, y_{follower2}) \quad (20)$$

$$x_{follower1} = x_{leader} - d * n, \quad y_{follower1} = y_{leader} + d * m * B_1 \quad (21)$$

$$x_{follower2} = x_{leader} - (k - B_2)d * n, \quad y_{follower2} = y_{leader} - d * m * B_2 \quad (22)$$

where function  $\mathbf{M}_s(x, y)$  returns the value at point  $(x, y)$  in  $\mathbf{M}_s$ .  $B_1$  and  $B_2$  are two deformation parameters controlling the coordinates of sub-target points. Values of deformation parameters are equal to values in the map  $\mathbf{M}_s$  at  $(x_{follower1}, y_{follower1})$  and  $(x_{follower2}, y_{follower2})$ . It could be observed that as distances between followers and obstacles becomes smaller, deformation parameter ( $B_1$  or  $B_2$ ) values will be closer to 0, which decreases the distance values in the y axis attracting followers to stay closer to each other. Parameter  $k$  is a parameter indicating the priority of each follower to deal with the situation of conflict when more than one follower is required to adjust positions.

### 5.2.2. USV formation path planning algorithm

Before describing the specific algorithm structure, it first should be noted that the key feature distinguishing the single vehicle path planning and the formation path planning is the internal collision avoidance within the formation. To address it, each vehicle of the formation should be treated as an additional obstacle called the ‘formation obstacle’ to others during each iteration. In this paper, the ‘formation obstacle’ is represented by a two-dimensional Gaussian function, which could generate sufficient repulsive force to prevent the collisions.

Fig. 11 represents the algorithm flow chart. The algorithm is developed based on the AFMS algorithm and uses the navigation environment map as

the input. The map is first converted into a binary map ( $W_o$ ). Difference between the algorithm in Garrido et al. (2011) and this paper is that based on  $W_o$ , the leader USV detects its instantaneous heading angle and employs the AFMS algorithm to generate an trajectory according to its heading and turning constraints. After the leader's path is determined, the algorithm then computes sub-target point for each follower by referring to predefined formation configuration. Using the sub-targets as the temporary end points, the algorithm generates trajectories for followers in sequence. Note that all the other vehicles except the USV being currently planned are treated as 'formation obstacles' at this stage to ensuring the safety of the path. Now, each USV of the formation has been calculated with a trajectory. By following trajectories, the formation will move to the next point and start the next iteration until the final target point is reached.

## 6. Algorithm validations

This section discusses and analyses the proposed algorithm outputs by using two main sets of simulations. The first simulation is undertaken to validate the formation forming algorithm with two different test scenarios that have been designed as: 1) USVs start mission with uniform heading, and 2) USVs start mission with different headings and potential conflicts.

The second simulation has been carried out with the aim of testing the formation path planning algorithm in a static environment. In this simulation, it is assumed that a proper formation shape has already been formed and the algorithm has been mainly tested to validate: 1) if the formation can plan the path according to the initial heading, 2) if the formation shape can be maintained in a safe area and 3) if the formation is able to adapt and adjust shape when collision risks exist.

All of the simulation results are presented by time-dependent motion sequences. The algorithm has been coded in Matlab and simulations are run on the computer with a Pentium i7 3.4 GHz processor and 4GB of RAM.

### 6.1. Verification of USV formation forming

In this test, three identical USVs with the same dynamic characteristics are employed. USVs start from random locations without forming any expected

formation shape. The test area has been designed as a static environment with two landmasses located as shown in Fig. 12. In order to simulate an area having enough space for USVs to complete the formation forming mission, the dimensions of area constructed are  $100 \text{ m} \times 100 \text{ m}$ , which is further represented as a bitmap of  $500 \text{ pixels} \times 500 \text{ pixels}$  (1 pixel = 0.2 m).

#### 6.1.1. Formation forming test case 1 (FF1)

The aim of the FF1 tests the scenario that USVs have the same initial heading but with different starting points. USVs are expected to form a triangle shape when they arrive at their target points. The specific testing configurations are listed in Table 1. It should be noted that when considering the formation angle, the leader vehicle's instantaneous heading is 0 degree, and the angle is increasing anti-clockwise. In addition, both the acceleration and deceleration parameters can be used to reduce the distance error between the leader and follower based upon their actual relative distance. Meanwhile, the deceleration parameter can also help the vehicle eliminate the conflict by slowing down the speed according to the priority scheme proposed in Section 5.1.1.

Simulation results are represented in Fig. 12. USVs are identified by different colours, i.e. the leader USV is plotted in red with follower 1 in blue and follower 2 in magenta. Target points are displayed with blue circle markers. The planned trajectories for the three USVs are plotted in red; whereas, the travelled routes are represented by the blue dotted lines. In addition, the green dotted lines are drawn to display the instantaneous formation shape while USVs are tracking, and the instantaneous heading angle of each USV is plotted by the black arrow.

In Fig. 12a, it is shown that three trajectories have been calculated by the algorithm at the time step 1. Since they have the same initial heading angle ( $90^\circ$ ), the three paths have similar form. Motion sequences at time steps 80, 184 and 280 are presented in Fig. 12b - 12d. It is evident that the desired formation shape has been gradually generated while the USVs track their trajectories, and eventually form the shape at the target points.

The quantitative analysis of the FF1 is shown in Fig. 13, where the heading angles and velocities of the three USVs are displayed and compared. In Fig. 13a, it can be observed that three USVs start on same headings which varies

smoothly through the whole period, which avoids severe heading change along the route. In terms of the speed information shown in Fig. 13b, the leader USV remains at the desired speed throughout. However, higher speeds are observed both for follower 1 and follower 2 as both of them are farther away from the desired distance to the leader at the beginning, therefore their speeds are higher to reduce the distance error. Also, small oscillations occur for the two followers as they approach the desired distance. This is mainly caused by the controller design, which adopts a piecewise function strategy. Such a problem can be compensated by other controllers such as the PID controller, which is beyond the scope of this work.

### *6.1.2. Formation forming test case 2 (FF2)*

In this test (FF2), a more complex scenario has been tested. The USVs are now starting from different locations each with different initial headings, which may cause potential conflict. Test configuration parameters are listed in Table 2.

From the movements records illustrated in Fig. 14, it is evident that in order to fulfil the requirements imposed by the initial headings as well as the target points, the two followers' trajectories are generated towards each other in the beginning then again before reaching the target points, forming two crossing points. By following these two trajectories, conflict may occur at each crossing point shown in Fig. 14a. In Fig. 14b, the first conflict has emerged, and follower 1 (plotted in blue) is closer to the conflict point thereby is being given a higher priority. Follower 2 has lower priority and becomes the 'give-way' vessel, the speed of which is therefore reduced to avoid the conflict. The avoidance time period for conflict 1 is marked as 'Elimination of conflict 1' in Fig. 15b. Once the first conflict has been resolved, follower 2 speeds up to continue to track its trajectory while follower 1 is travelling at its desired velocity.

At time step 147 (shown in Fig. 14c), the second conflict can occur while the two followers are travelling towards their target points. Again, follower 1 has higher priority as it is closer to the conflict point, which causes the follower 2 to decelerate and wait for the conflict to be resolved (shown in Fig. 14d. Fig. 14e and Fig. 14f represents the movements of all USVs once there are no more conflicts with the targets being reached at time step 298. The smooth

heading variations of the three USVs are guaranteed during whole operation time period, which is displayed and demonstrated in Fig. 15a.

### 6.2. Verification of USV formation path planning

In this validation, the USV formation application has been tested with different initial headings. The dimension of the simulation environment is  $755 \text{ m} \times 377 \text{ m}$  with multiple static obstacles randomly located in the area. The USV formation has been assigned with a common formation shape, the triangle shape, with three identical vehicles. One of the vehicle has been designated as the leader vehicle, and it is assumed that during the operation there is no communication delay within the formation and no malfunctions happened to any vehicle of the formation. The formation parameters are listed out in Table. 3. For the first test, the initial formation heading is  $30^\circ$ . In order to form a regular triangle formation shape, which has been adopted in most formation operations, desired angles of two followers to the leader USV are  $45^\circ$  and  $-45^\circ$  with the same desired distance of 45 m. In the second test, all parameters are the same except the initial heading being set up as  $150^\circ$ .

When the USV formation has an initial heading of  $30^\circ$ , the movement sequences of the formation are displayed in Fig. 16. The leader USV is represented in red and two followers are in blue and magenta colour. To better illustrate the formation behaviour, the instantaneous formation shape is plotted with green line. The generated trajectory for the leader at each time step by using the proposed AFMS algorithm is plotted as red line. It can be observed that the trajectory stays close to the heading angle direction at each time step to satisfy the heading angle as well as turning constraint requirements of the USV. Fig. 16a shows the formation movement situation at time step 1. Because the formation is located at a relatively ‘safe’ area without need to avoid any obstacles, a desired triangle formation shape has been formed. However, at time step 38 (Fig. 16b), the follower2 USV (plotted in magenta) is close to obstacles thus the formation takes an appropriate manoeuvre by ‘shrinking’ its shape. The ‘shrinking’ manoeuvre is the result of the leader-follower sub-target generation scheme, where the followers are able to adaptively choose their individual deformation parameters ( $B$ ) based upon their specific locations and change the formation shape accordingly. Such a collision avoidance manoeuvre has been repeatedly adopted by the USV formation for the whole operation (Fig. 16c - Fig. 16h) until the end



point has been reached.

When the initial heading is changed to  $150^\circ$ , results are displayed in Fig. 17. It is evident that the generated path is now different from the previous one (Fig. 16). To be compliant with the heading angle constraint, the trajectory first makes a port side turn to form a ‘turning circle’ for the formation to adjust its heading. Once the formation has been on its adequate direction, the rest of the trajectory is located at the bottom half of the environment. By following such a trajectory, the formation has to pass through three ‘narrow channel’ areas by deforming its shape more similar to line shape (See Fig. 17c, 17e and 17g).

## 7. Conclusions and future works

This paper has presented and discussed path planning algorithms for USV formations. The algorithms have been designed with two different sets to solve the formation forming and path planning problems, respectively. The core of the algorithm is based on the ‘angle-guidance fast marching square’ (AFMS) method, which has been proposed to solve the USV's heading constraint problem encountered in the real experiments. The AFMS inherits the advantages of the FMS's but takes special considerations of USV's heading angle. By using a ‘guidance range’ (GR) area, the generated path is compliant with USV's turning constraint and is capable of assisting the USV adjusting its heading towards the desired orientation, which considerably increases the trajectory’s practicability. By using the AFMS, simulation results show that the newly designed algorithms can effectively and efficiently form and navigate USV formations while adaptively modifying the paths according to different headings.

In terms of the future work, the implementation of the designed algorithms on a practical USV platform, *Springer* USV, will be first carried out. *Springer* is a low-cost research vessel designed and developed by Plymouth University with the primary aim of being undertaking pollutant tracking and environmental monitoring operations (Naeem et al., 2008). Some path planning experiments have already been undertaken on *Springer* with the initial test results reported in (Liu et al., 2015b). The next step is to use the similar experiment method proposed in (Liu et al., 2015b) to verify and analyse the

performance of the AFMS with other algorithms such as A\* and FMS. Currently, *Springer* can only perform basic operations by following predefined waypoints. It is desired that by integrating the path planning algorithms in to the navigation guidance and control (NGC) system of the vehicle, *Springer* is capable of intelligently planning the trajectories and making decisions on board, which will enormously increase the autonomy level of the vehicle.

Besides the practical vehicle verification, attentions should also be placed on the dynamic environment, which includes various moving ships associated with different encounter situations. To address this, it is not only necessary to integrate COLREGs regulations into the algorithm, but also to develop a priority strategy to facilitate the USV in taking proper collision avoidance decisions to avoid multiple vessels. In addition, the ‘dynamic environment’ can also be referred to the varying environment which is possibly affected by ocean conditions such as currents and tidal. ‘Smart’ actions are required for USV to avoid hazard areas when it is experiencing severe ocean conditions.

## 8. Acknowledgements

This work is supported by the ACCeSS group. The Atlantic Centre for the innovative design and Control of Small Ships (ACCeSS) is an ONR-NNRNE programme with Grant no. N0014-10-1-0652, the group consists of universities and industry partners conducting small ships related researches. The first author would like to thank the China Scholarship Council (CSC) for supporting his studies at the University College London, UK.

## References

- Álvarez, D., Gómez, J.V., Garrido, S., Moreno, L., 2015. 3d robot formations path planning with fast marching square. *Journal of Intelligent & Robotic Systems* , 1–17.
- Annamalai, A., Sutton, R., Yang, C., Culverhouse, P., Sharma, S., 2015. Robust adaptive control of an uninhabited surface vehicle. *Journal of Intelligent & Robotic Systems* 78, 319–38.
- Balch, T., Arkin, R.C., 1998. Behavior-based formation control for multi-robot teams. *Robotics and Automation, IEEE Transactions on* 14, 926–39.

- Bertaska, I.R., Shah, B., von Ellenrieder, K., Švec, P., Klinger, W., Sinisterra, A.J., Dhanak, M., Gupta, S.K., 2015. Experimental evaluation of automatically-generated behaviors for usv operations. *Ocean Engineering* 106, 496–514.
- Campbell, S., Naeem, W., Irwin, G.W., 2012. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control* 36, 267–83.
- Chen, J., Sun, D., Yang, J., Chen, H., 2009. A leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme. *The International Journal of Robotics Research* .
- Clark, C.M., Bretl, T., Rock, S., 2002. Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems, in: *Aerospace Conference Proceedings, 2002*. IEEE, IEEE. pp. 7–3621.
- Cormen, T.H., 2009. *Introduction to algorithms*. MIT press.
- Garrido, S., Malfaz, M., Blanco, D., 2013. Application of the fast marching method for outdoor motion planning in robotics. *Robotics and Autonomous Systems* 61, 106–14.
- Garrido, S., Moreno, L., Blanco, D., 2008. Exploration of a cluttered environment using voronoi transform and fast marching. *Robotics and Autonomous Systems*. 56, 1069 –81.
- Garrido, S., Moreno, L., Blanco, D., Martin, F., 2007. FM2: A real-time fast marching sensor-based motion planner, in: *2007 IEEE/ASME international conference on Advanced intelligent mechatronics.*, pp. 1–6.
- Garrido, S., Moreno, L., Lima, P.U., 2011. Robot formation motion planning using fast marching. *Robotics and Autonomous Systems*. 59, 675 –83.
- Gómez, J.V., Lumbier, A., Garrido, S., Moreno, L., 2013. Planning robot formations with fast marching square including uncertainty conditions. *Robotics and Autonomous Systems* 61, 137–52.
- Hao, Y., Agrawal, S.K., 2005. Planning and control of ugv formations in a dynamic environment: A practical framework with experiments. *Robotics and Autonomous systems* 51, 101–10.

- Kim, H., Kim, D., Shin, J.U., Kim, H., Myung, H., 2014. Angular rate-constrained path planning algorithm for unmanned surface vehicles. *Ocean Engineering*. 84, 37 – 44.
- Kitts, C., Mahacek, P., Adamek, T., Rasal, K., Howard, V., Li, S., Badaoui, A., Kirkwood, W., Wheat, G., Hulme, S., 2012. Field operation of a robotic small waterplane area twin hull boat for shallow-water bathymetric characterization. *Journal of Field Robotics* 29, 924–38.
- Lee, T., Kim, H., Chung, H., Bang, Y., Myung, H., 2015. Energy efficient path planning for a marine surface vehicle considering heading angle. *Ocean Engineering* 107, 118–31.
- Lewis, M.A., Tan, K.H., 1997. High precision formation control of mobile robots using virtual structures. *Autonomous Robots* 4, 387–403.
- Liu, S., Sun, D., Zhu, C., 2014. A dynamic priority based path planning for cooperation of multiple mobile robots in formation forming. *Robotics and Computer-Integrated Manufacturing* 30, 589–96.
- Liu, Y., Bucknall, R., 2015. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering* 97, 126–44.
- Liu, Y., Liu, W., Song, R., Bucknall, R., 2015a. Predictive navigation of unmanned surface vehicles in a dynamic maritime environment when using the fast marching method. *International Journal of Adaptive Control and Signal Processing* .
- Liu, Y., Song, R., Bucknall, R., 2015b. A practical path planning and navigation algorithm for an unmanned surface vehicle using the fast marching algorithm, in: *OCEANS 2015-Genova, IEEE*. pp. 1–7.
- Mahacek, P., Kitts, C., Mas, I., et al., 2012. Dynamic guarding of marine assets through cluster control of automated surface vessel fleets. *Mechanics, IEEE/ASME Transactions on* 17, 65–75.
- Manley, J.E., 2008. Unmanned surface vehicles, 15 years of development, in: *OCEANS 2008, IEEE*. pp. 1–4.

- Mastellone, S., Stipanović, D.M., Graunke, C.R., Intlekofer, K.A., Spong, M.W., 2008. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *The International Journal of Robotics Research* 27, 107–26.
- Mehrjerdi, H., Ghomman, J., Saad, M., 2011. Nonlinear coordination control for a group of mobile robots using a virtual structure. *Mechatronics* 21, 1147–55.
- Monteiro, S., Bicho, E., 2002. A dynamical systems approach to behavior-based formation control, in: *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, IEEE. pp. 2606–11.
- Naeem, W., Irwin, G.W., Yang, A., 2012. Colregs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics*. 22, 669 –78.
- Naeem, W., Xu, T., Sutton, R., Tiano, A., 2008. The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 222, 67–79.
- Peng, Z., Wang, D., Chen, Z., Hu, X., Lan, W., 2013. Adaptive dynamic surface control for formations of autonomous surface vehicles with uncertain dynamics. *Control Systems Technology, IEEE Transactions on* 21, 513–20.
- Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J., Lane, D., 2007. Path planning for autonomous underwater vehicles. *Robotics, IEEE Transactions on* 23, 331–41.
- Rui, S., 2014. Path Planning and Bi-directional Communication for Unmanned Surface Vehicle. Technical Report. Marine and Industrial Dynamic Analysis Research Group, University of Plymouth.
- Sethian, J.A., 1996. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences* 93, 1591–5.
- Sharma, S.K., Sutton, R., Motwani, A., Annamalai, A., 2014. Non-linear control algorithms for an unmanned surface vehicle. *Proceedings of the*

- Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 228, 146–55.
- Shojaei, K., 2015. Leader–follower formation control of underactuated autonomous marine surface vehicles with limited torque. *Ocean Engineering* 105, 196–205.
- Smierzchalski, R., 1999. Evolutionary trajectory planning of ships in navigation traffic areas. *Journal of marine science and technology* 4, 1–6.
- Tam, C., Bucknall, R., 2010a. Collision risk assessment for ships. *Journal of marine science and technology* 15, 257–70.
- Tam, C., Bucknall, R., 2010b. Path-planning algorithm for ships in close-range encounters. *Journal of marine science and technology* 15, 395–407.
- Tam, C., Bucknall, R., 2013. Cooperative path planning algorithm for marine surface vessels. *Ocean Engineering* 57, 25–33.
- Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic robotics. MIT press.
- Tsou, M.C., Hsueh, C.K., 2010. The study of ship collision avoidance route planning by ant colony algorithm. *Journal of Marine Science and Technology* 18, 746–56.
- Xu, B., Stilwell, D.J., Kurdila, A.J., 2013. Fast path re-planning based on fast marching and level sets. *Journal of Intelligent & Robotic Systems* 71, 303–17.
- Xue, Y., Clelland, D., Lee, B., Han, D., 2011. Automatic simulation of ship navigation. *Ocean Engineering*. 38, 2290 –305.

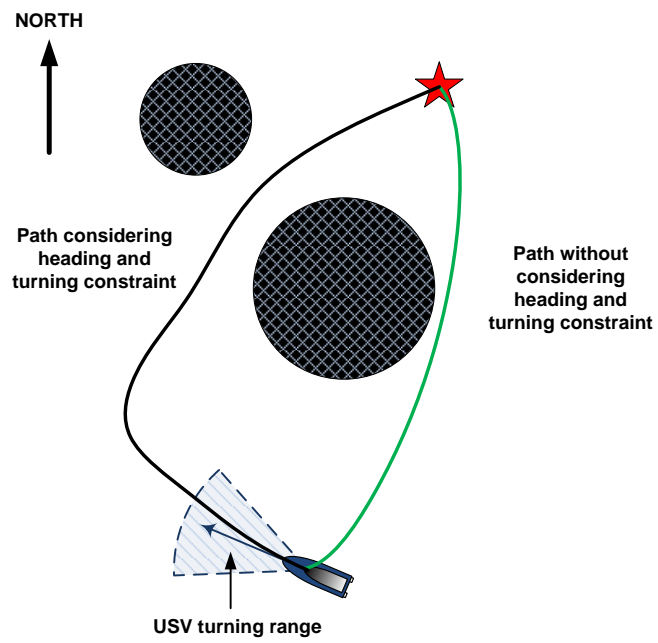


Figure 1: Turning angle guided path planning concept.

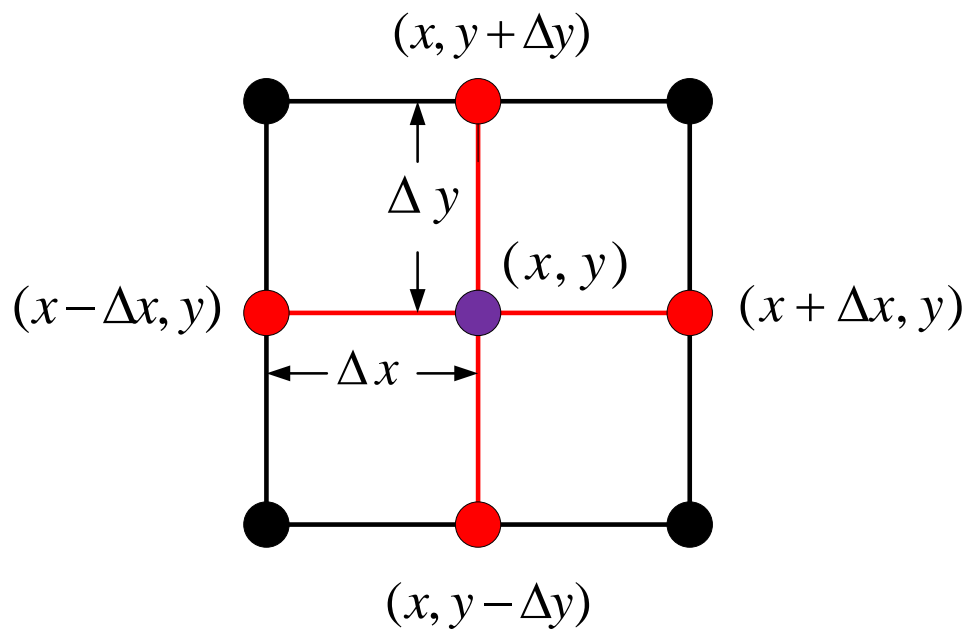
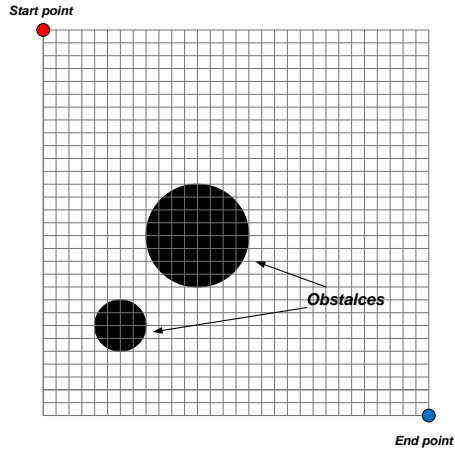
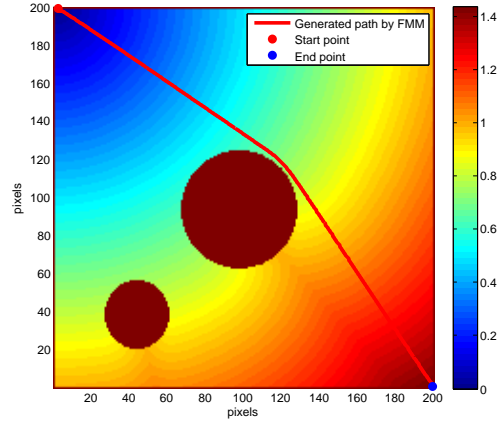


Figure 2: Grid point  $(x, y)$  and its neighbours. The grid map has four connectivity, the grid point therefore has four neighbour points. The grid size is  $\Delta x$  and  $\Delta y$  in the  $x$  and  $y$  direction respectively.

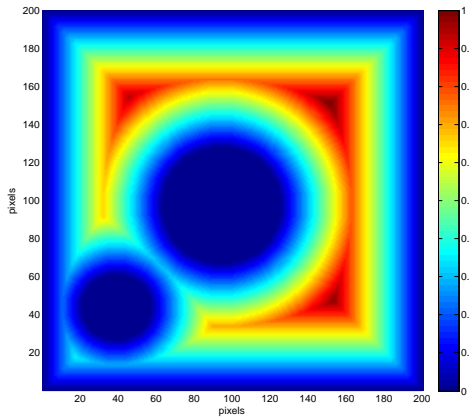




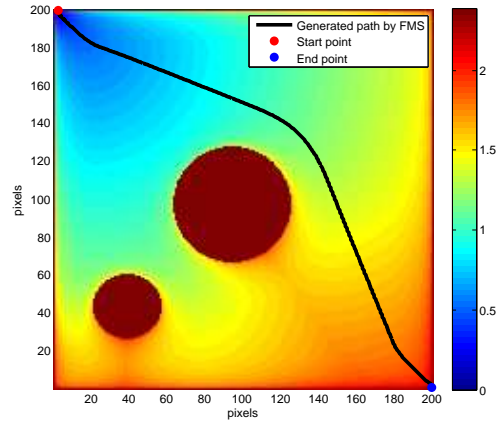
(a)



(b)



(c)



(d)

Figure 3: (a) The grid map. (b) Potential field ( $T_{FMM}$ ) generated by using the FMM. Potential value at each point represents the local distance to the start point. The red line is the path searched by the gradient descent method based upon  $T_{FMM}$ . (c) The safety potential map ( $M_s$ ) generated by the first step of the FMS. Values in the map is the safety indices to indicate the safety of local points. (d) Potential field ( $T_{FMS}$ ) generated by using the second step of the FMS. The black line is the calculated trajectory using the FMS.

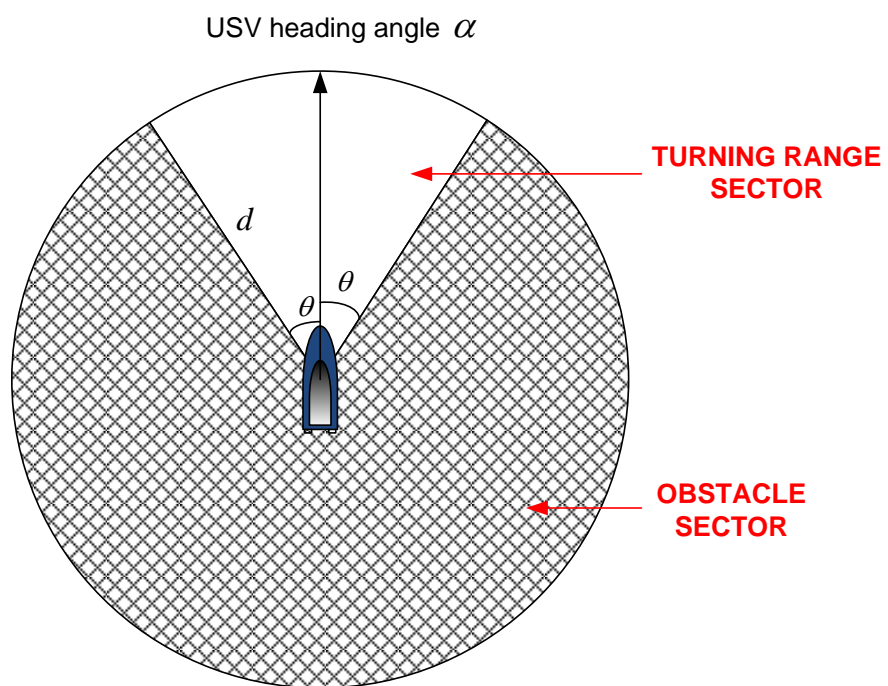


Figure 4: The illustration of guidance range (GR).

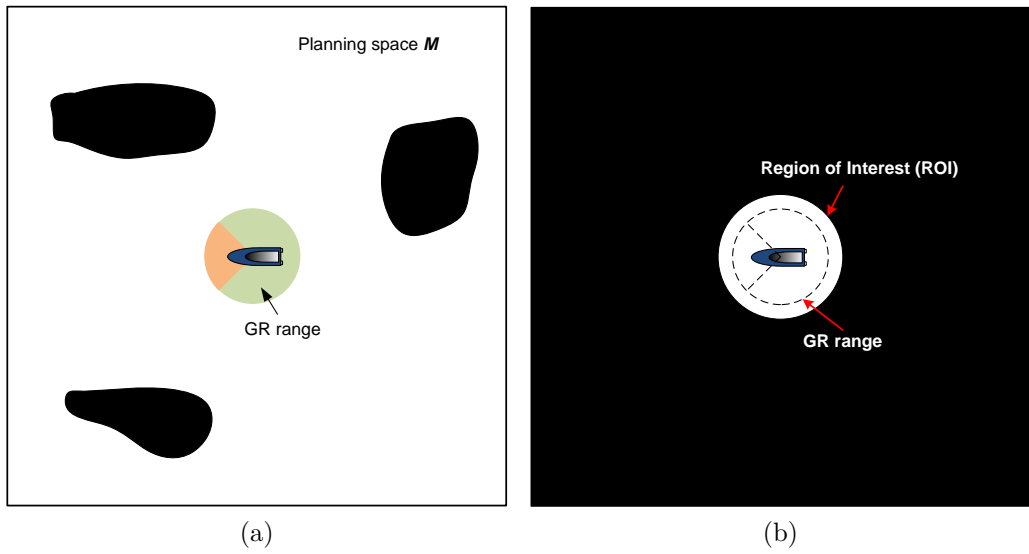


Figure 5: The region of interest (ROI) over the original planning space. (a) The original planning space with all the information. (b) The ROI, which is a binary image having the ROI region (white area) equal to 1 and other areas (black area) equal to 0. The GR range has been plotted with the dash line to indicate the difference between the GR and the ROI.

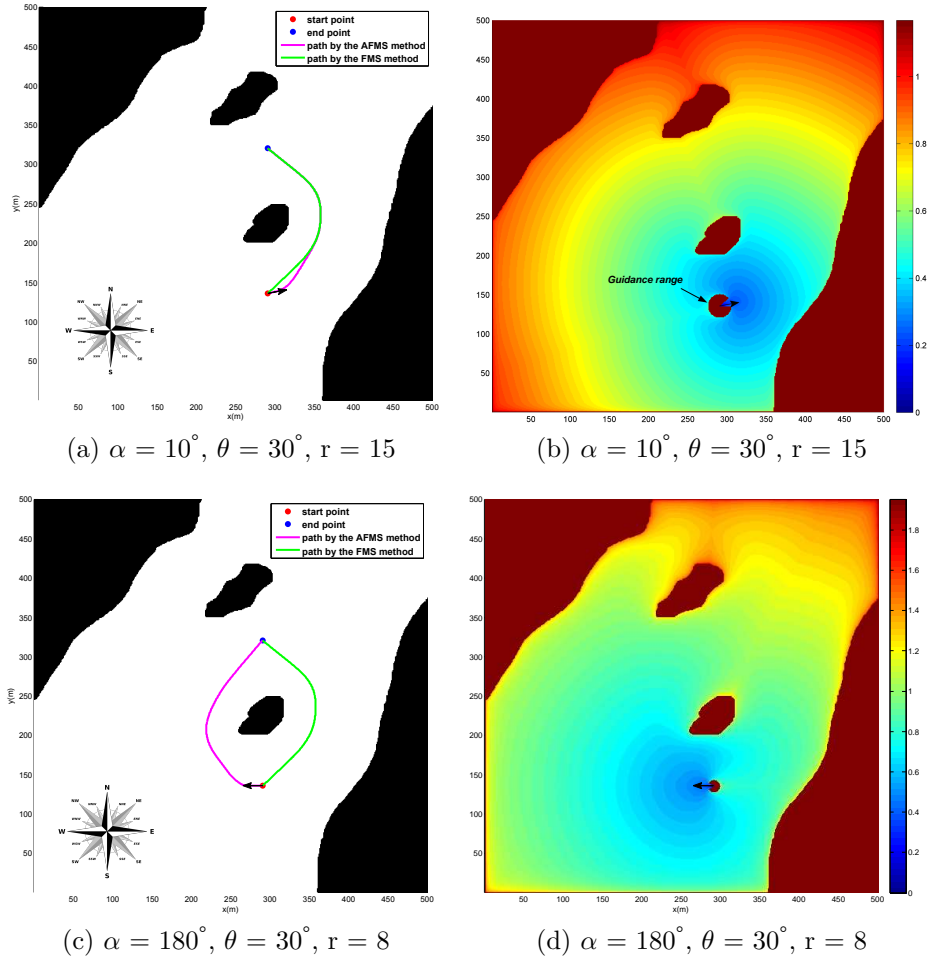


Figure 6: Comparison of trajectories generated by the AFMS and the FMS. (a) Case 1: The USV has a heading angle of  $10^\circ$  with the turning angle of  $30^\circ$  and the range distance of 15 m. The trajectory generated by the AFMS follows the USV's heading making it have a slight offset with the path calculated from the FMS. (b) The associated potential field of Case 1. (c) Case 2: The USV has a heading of  $180^\circ$  with the turning angle of  $30^\circ$  and the range distance of 8 m. The AFMS's path stays alongside the direction of USV's heading by following the guidance of the GR, and it eventually avoids the obstacle from the left side. (d) The associated potential field of Case 2.

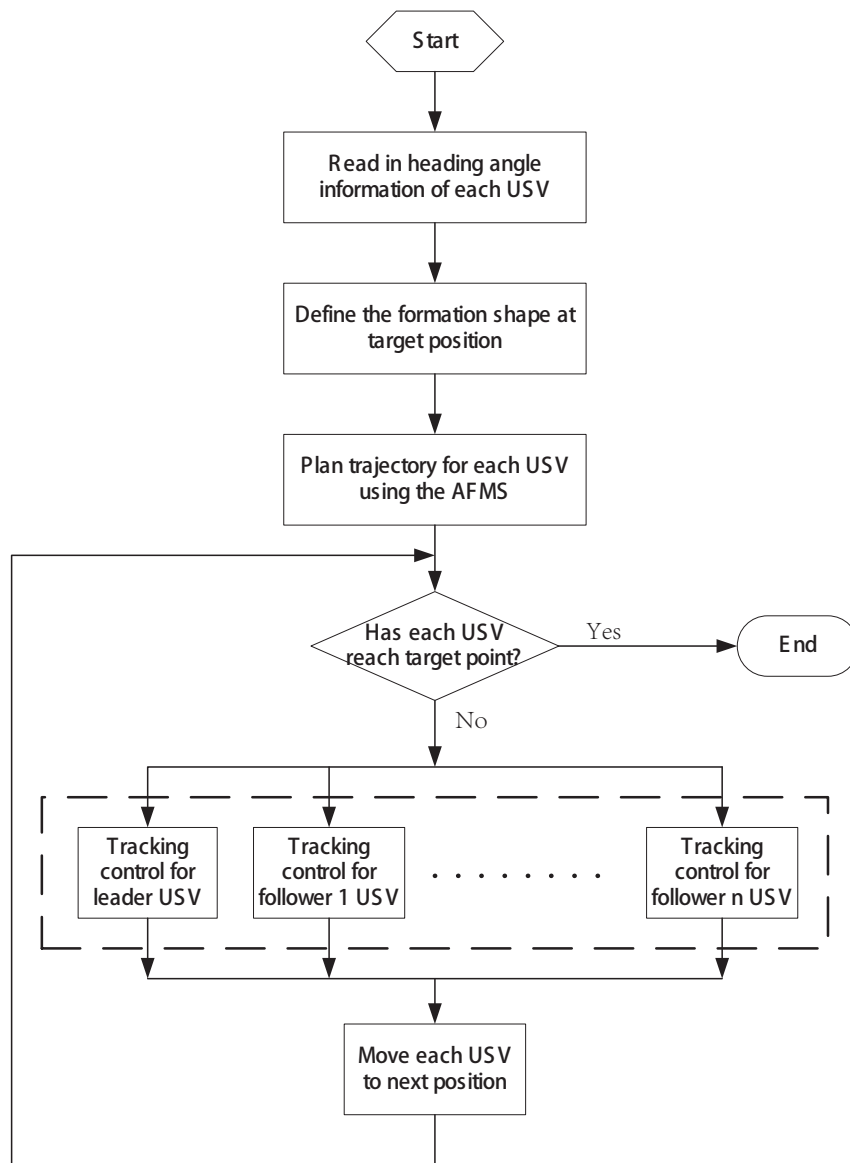


Figure 7: Flow chart of the USV formation forming algorithm.

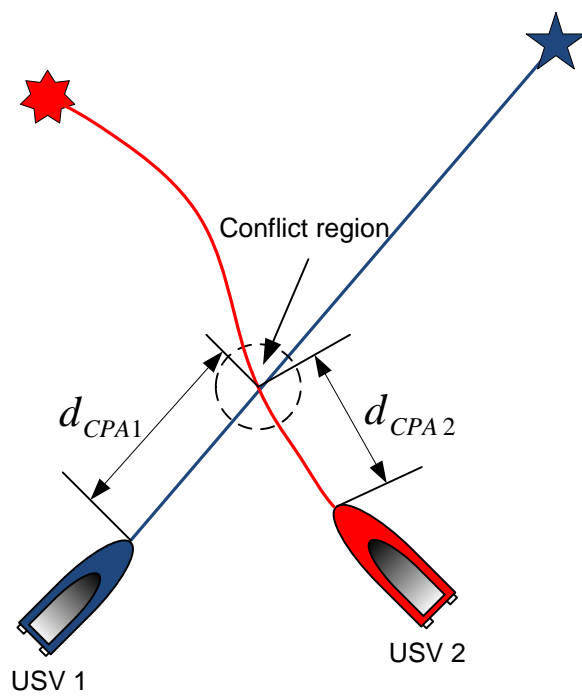


Figure 8: The conflict occurring between two USVs. USV 2 has smaller DCPA thereby having higher priority than USV 1.

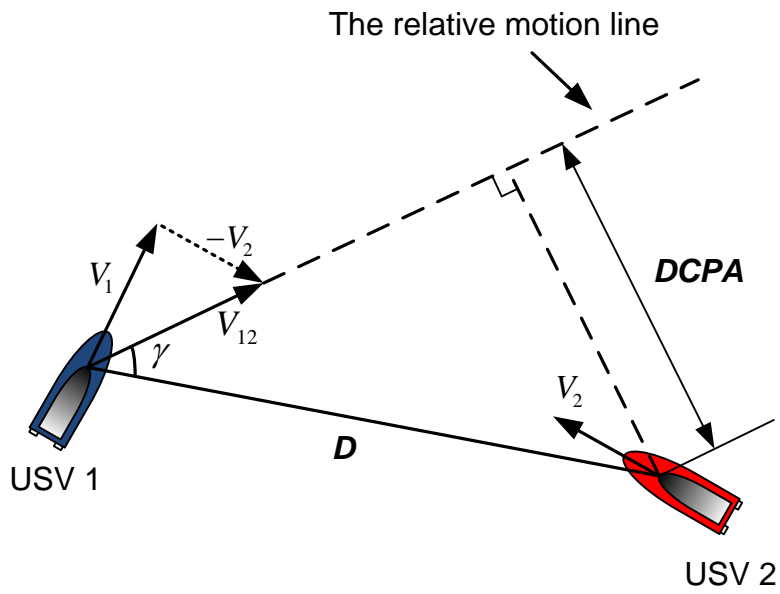


Figure 9: The collision risk assessment between two USVs based upon the distance to the closest point of approach (DCPA).

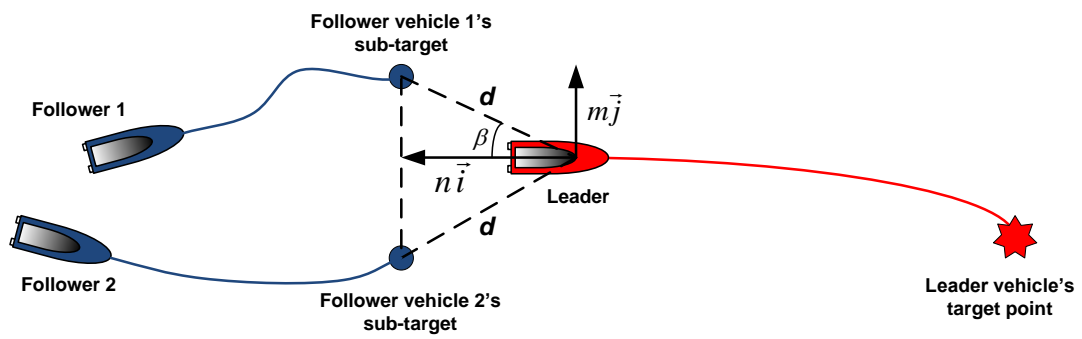


Figure 10: Sub-target points generation illustration. Sub-targets are generated by referring to the desired direction and distance between the leader and the follower.



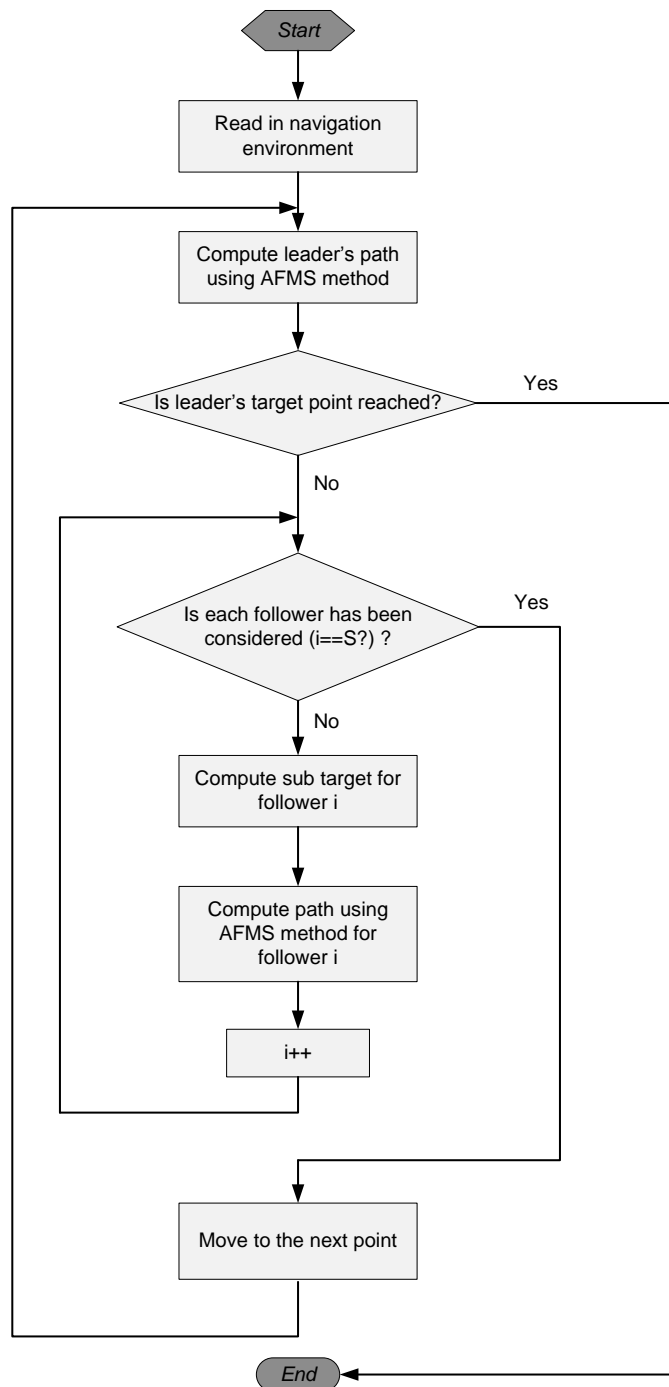


Figure 11: Algorithm flow chart of USV formation path planning in static environment.

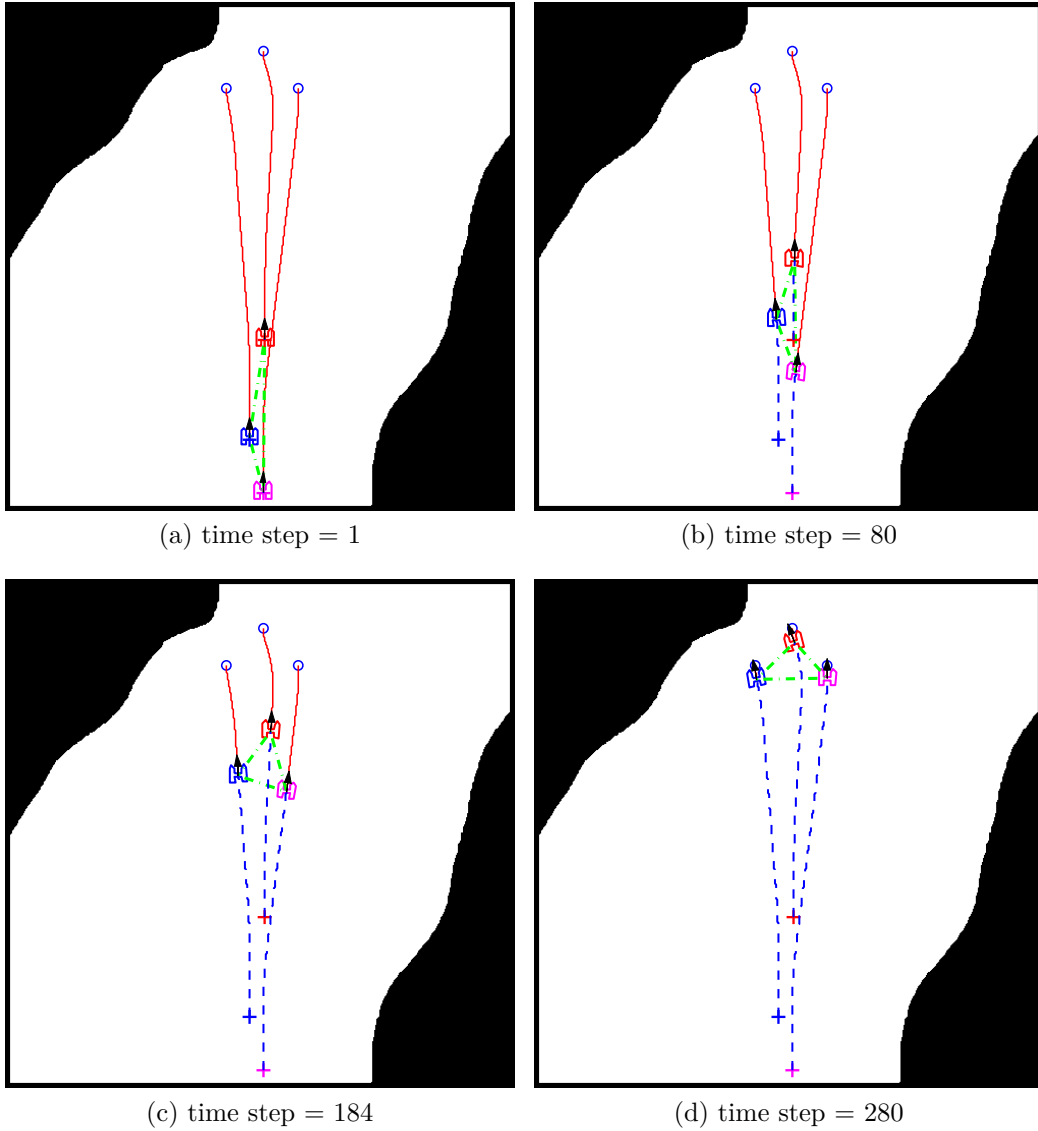
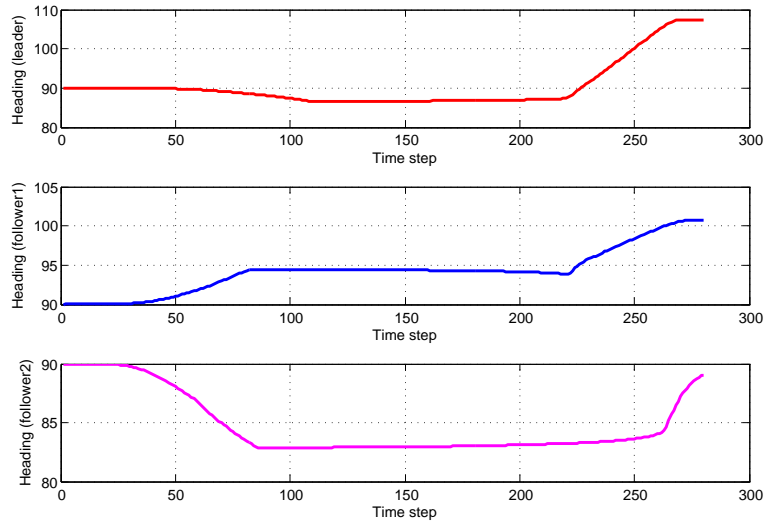
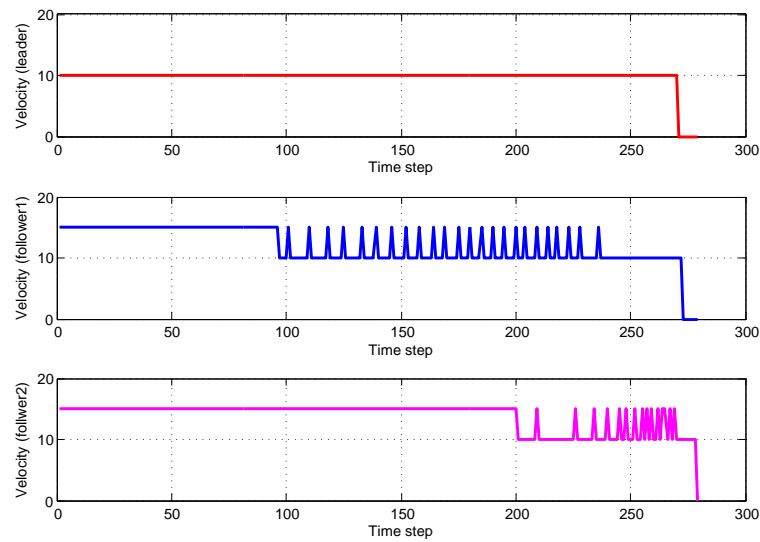


Figure 12: Formation movement sequences of FF1. USVs have the same initial heading but different starting points. USVs are expected to form a triangular shape when they arrive at target points. Similar trajectories are calculated for each USV in the formation as they have the same initial heading.



(a)



(b)

Figure 13: Quantitative evaluation of FF1. (a) Heading information of FF1. USVs start with the same heading and headings smoothly vary though the whole period. (b) Velocity information of FF1.

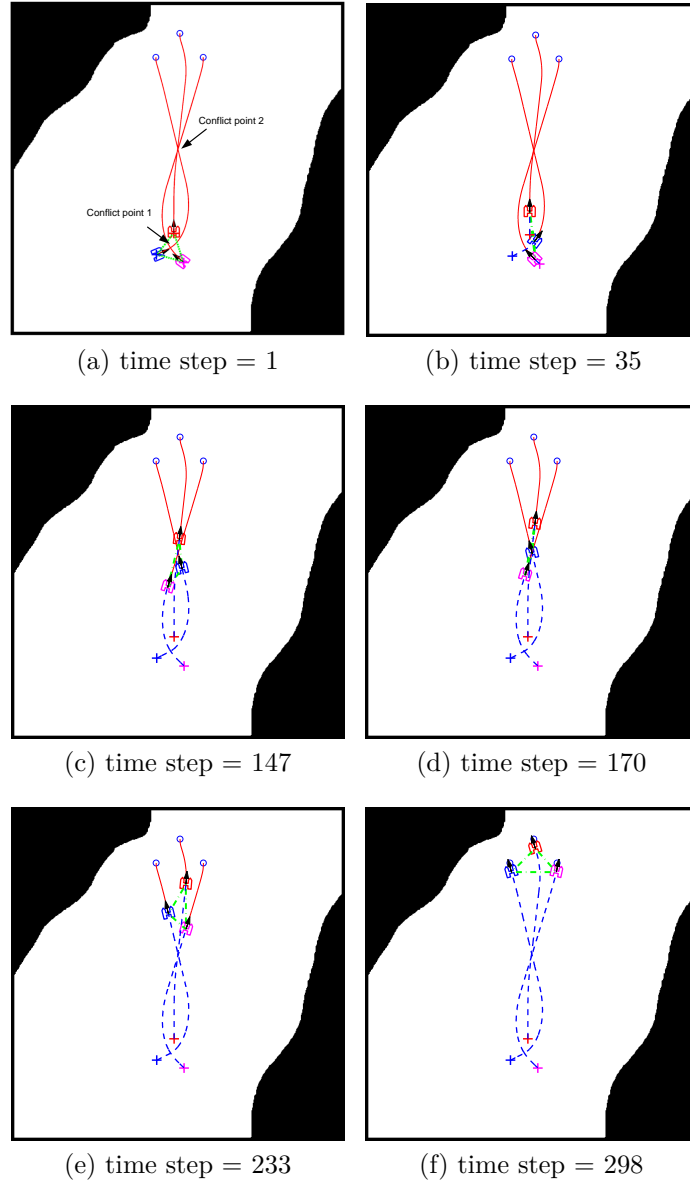
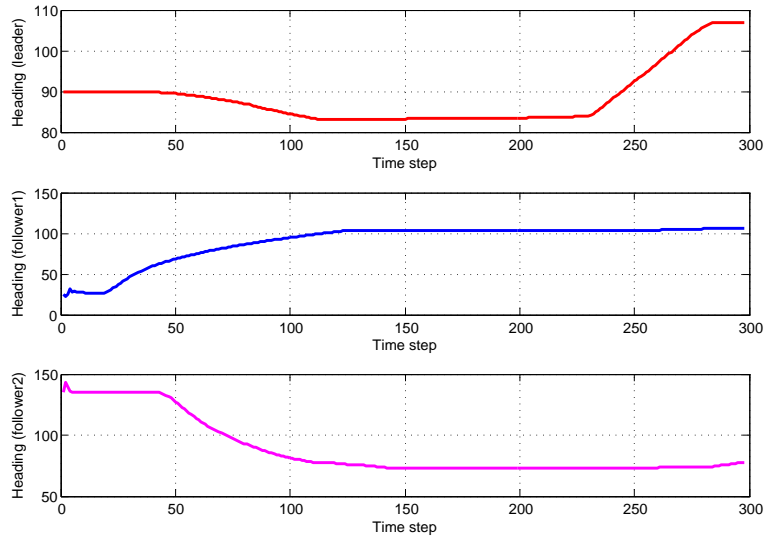
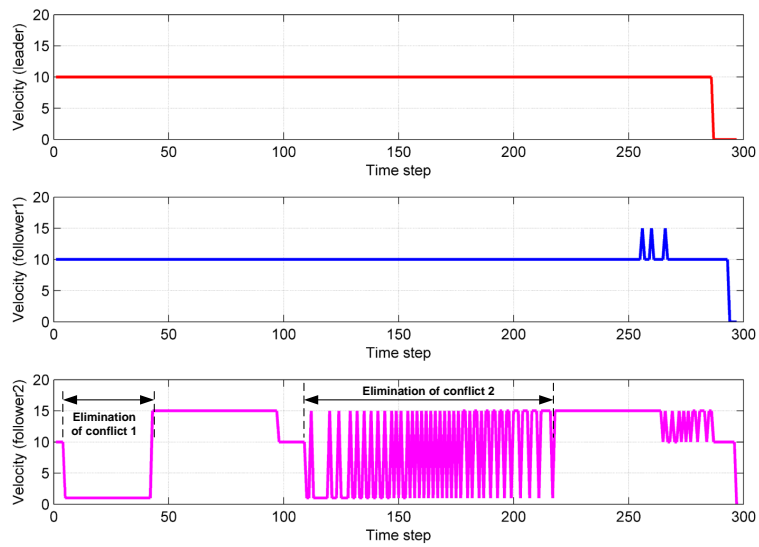


Figure 14: Formation movement sequences of FF2. Trajectories have been generated by considering the dynamics of USV but with conflicts *en route*. (a) and (b) eliminate the first conflict. (c) and (d) eliminate the second one. (e) and (f) show the formation is generated when USVs approach the target points.



(a)



(b)

Figure 15: Quantitative evaluation of FF2. (a) Heading information of FF2. (b) Velocity information of FF2.

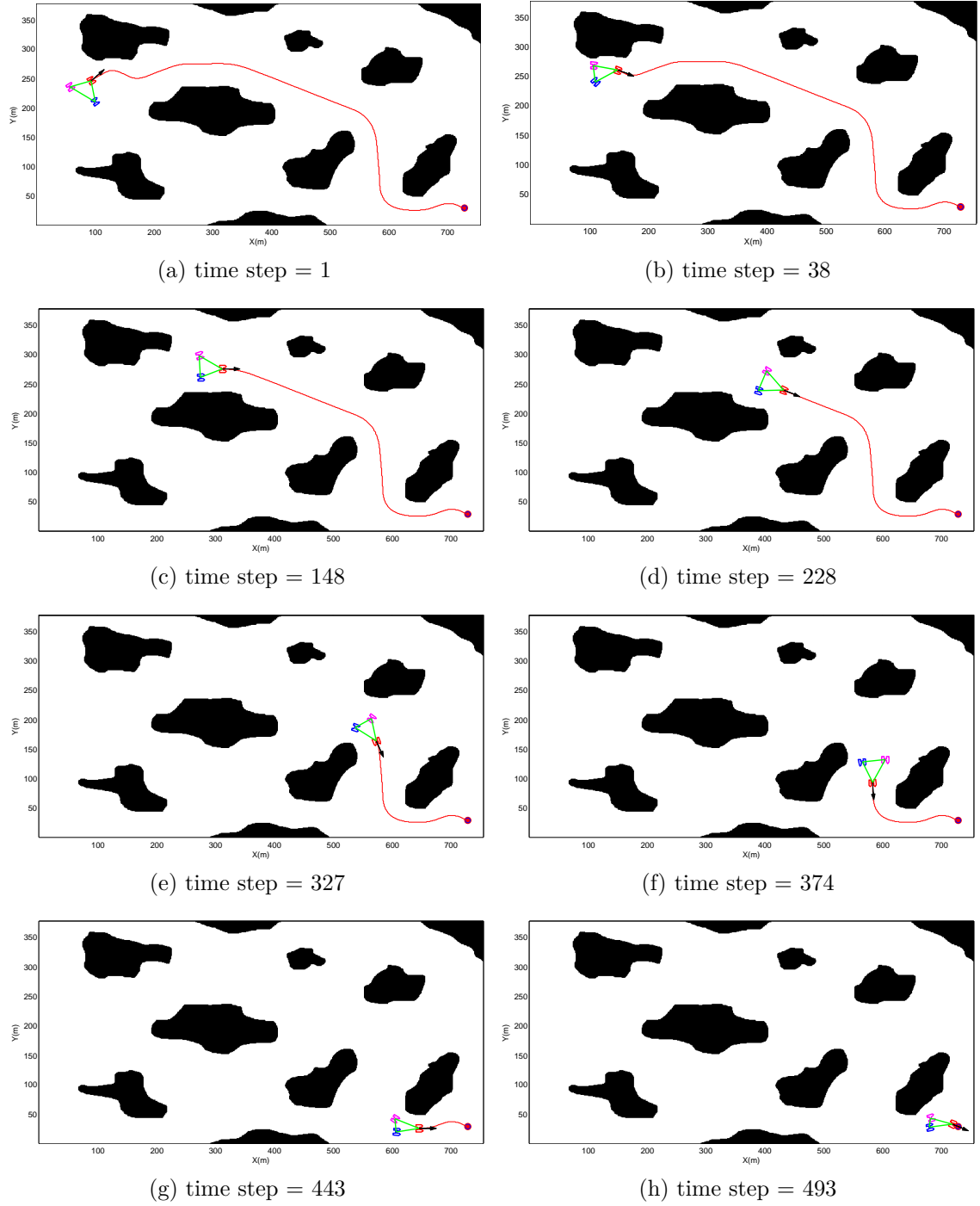


Figure 16: Formation movement sequences for USV formation with initial heading of  $30^\circ$ .

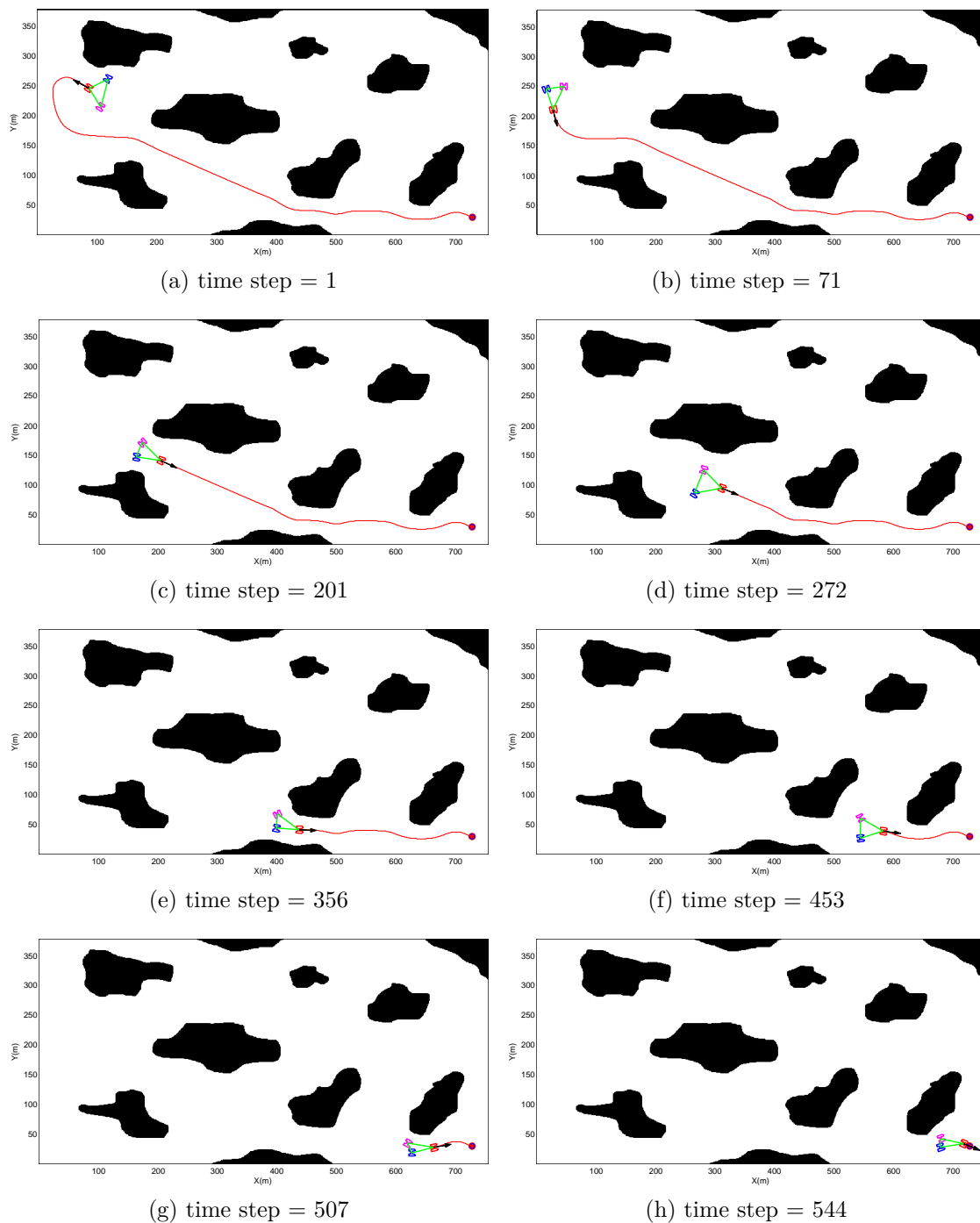


Figure 17: Formation movement sequences for USV formation with initial heading of  $150^\circ$ .

Table 1: Testing configurations for FF1.

	<b>Leader USV</b>	<b>Follower 1 USV</b>	<b>Follower 2 USV</b>
Start point (pixels)	(256, 168)	(241, 70)	(254, 17)
Initial heading (degrees)	90	90	90
Formation distance (pixels)	50	50	50
Formation angle (degrees)	0	45	-45
Desired speed (pixels/time step)	10	10	10
Acceleration parameter	N/A	1.5	1.5
Deceleration parameter	N/A	0.1	0.1



Table 2: Testing configurations for FF2.

	<b>Leader USV</b>	<b>Follower 1 USV</b>	<b>Follower 2 USV</b>
Start point (pixels)	(245, 153)	(219, 121)	(260, 109)
Initial heading (degrees)	90	25	140
Formation distance (pixels)	50	50	50
Formation angle (degrees)	0	45	-45
Desired speed (pixels/time step)	10	10	10
Acceleration parameter	N/A	1.5	1.5
Deceleration parameter	N/A	0.1	0.1

Table 3: Configuration parameters USV formaion.

	Initial head- ing1	Initial head- ing2	Turning con- straint	Formation dis- tance	Formation angle
Leader USV	$30^\circ$	$150^\circ$	$30^\circ$	N/A	N/A
Follower1 USV	$30^\circ$	$150^\circ$	$30^\circ$	45 m	$45^\circ$
Follower2 USV	$30^\circ$	$150^\circ$	$30^\circ$	45 m	$-45^\circ$