Predictive navigation of unmanned surface vehicles in a dynamic maritime environment when using the fast marching method

Yuanchang Liu, Wenwen Liu, Rui Song and

Richard Bucknall[*][†]

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

ABSTRACT

Effective and intelligent path planning algorithms designed for operation in a dynamic marine environment are essential for the safe operation of unmanned surface vehicles (USVs). Most of the current research deals with the "dynamic problem" by basing solutions on the non-practical assumption that each USV has a robust communication channel to obtain essential information such as position and velocity of marine vehicles. In this paper, a Kalman Filter based predictive path planning algorithm is proposed. The algorithm has been designed to predict the trajectories of moving ships as well as the USV's own position in real time and accordingly assesses collision risk. For path planning a weighted fast marching square method is proposed and developed to search for the optimal path. The path can be optimised for mission requirements, such as minimum distance to travel and the most safety path by adjusting weighting parameters. The proposed algorithm has been validated using a number of simulations that include practical environmental aspects. The results show that the algorithms can sufficiently deal with complex traffic environments and that the generated practical path is suited for both unmanned and manned vessels.

KEY WORDS: unmanned surface vehicle, USV, predictive path planning, dynamic environment, fast marching method

1. INTRODUCTION

Unmanned surface vehicles (USVs) are receiving increasing attention due to their wide deployment in both military and civilian applications. As there is no human operator on board, a USV's autonomous navigation system plays a critical role. Figure 1 depicts the system structure of a typical USV navigation system. It consists of three different modules, i.e. the data acquisition module (DAM), the path planning module (PPM) and the advanced control module (ACM). A USV perceives its surrounding environment using the DAM, which acquires navigation information using a range of different sensors such as GPS, Inertial Measurement

[*] Correspondence to: Richard Bucknall, Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK

[†] Email: r.bucknall@ucl.ac.uk

Unit (IMU), Automatic Identification System (AIS) and marine radar. Using the data obtained, the algorithm in the PPM determines a safe path for the USV. The safe path includes a set of waypoints, which are used by the controller in the ACM as reference points to safely navigate the USV.

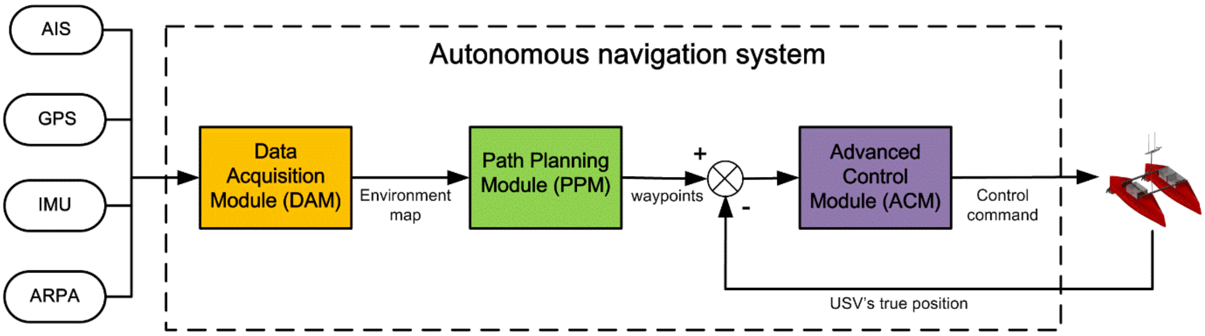

AIS
GPS
IMU
ARPA
Autonomous navigation system
Data Acquisition Module (DAM)
Environment map
Path Planning Module (PPM)
waypoints
+
-
Advanced Control Module (ACM)
Control command
USV's true position

Figure 1. The system structure of the autonomous navigation system for a USV.

Of the three modules, the PPM plays the most important role in the autonomous navigation system. The aim of path planning is to plan an optimised trajectory connecting mission start and end points without colliding with any obstacle *en route*. It should be noted that collision avoidance, especially for USV navigation, is one of the most important requirements since it ensures the safe operation of USV. In maritime environment, both static obstacles (buoys and rocks) and dynamic obstacles (moving ships) present potential collision risk for the USV. Any path planning algorithm should be capable of intelligently and autonomously avoiding these obstacles by maintaining a safe distance.

A number of different approaches have been proposed by many researchers in the past decade for autonomous marine path planning. Smierzchalski [1] first used the genetic algorithms to search navigational paths in maritime traffic areas. The proposed algorithm was able to steer a ship to avoid both static and dynamic obstacles. Tam and Bucknall [2,3] also implemented the evolutionary algorithm for ship path planning with specific emphasis on improving the algorithm's consistency and a vector field was constructed to direct the search routine towards the regions of interest. However, both of these research efforts suffered the problem of output incompleteness, a disadvantage of using this type of algorithm. To overcome this, the deterministic path planning algorithm has dominated USV path planning research in recent years. Xue et al [4] proposed an algorithm using the artificial potential field (APF) method. The algorithm created attractive potential fields over the planning space referring to the mission end point. For collision avoidance, repulsive fields surrounding each obstacle were generated. The final path was calculated by following the gradient of the total potential field. Another study undertaken by Naeem et al [5] solved searching incompleteness by adopting an improved A* algorithm and the line-of-sight guidance theory. Kim et al [6] reported work of USV path planning in a real environment by using the Theta* algorithm. This algorithm was an improved version of the A*, which specifically took the turning rates of the USV into consideration and improved the practicability of the output trajectory. However, there are also two main disadvantages for the deterministic path planning algorithms which prevent its wide adoption for applications in practical USV navigation systems. For the APF, the local minima problem is the most common issue, which could lead to the failure of finding a path in a complex environment. In terms of the A*, a grid based searching algorithm, the non-smoothness of generated path also restricts implementation for USVs.

In recent years, the Fast Marching Method (FMM) has been intensively used in robot path planning research. The FMM is a method that overcomes the shortcomings of the APF and the A* and is able to search for path with fast computational speed. A number of different improvements have been made to the FMM to increase its performance in various applications. Garrido et al [7, 8] first proposed a Voronoi Fast Marching (VFM) method, which combined the Voronoi graph based searching method with the FMM, to improve the safety of the path in cluttered environment for robot navigation. Gomez et al [9] then expanded the VFM method to the Fast Marching Square (FMS) in robot formation navigation application. Garrido et al [10] again explored the possibility of using the FMM in outdoor path planning problems. A weighted cost matrix was calculated before running the algorithm, and the generated path could be optimised for different optimisation requirements. Besides the application in robot navigation, the FMM has also been studied in other path planning problems. Petres et al [11] used the anisotropic fast marching to solve the problem of autonomous underwater vehicle in ocean environment with current. Xu et al [12] investigated the application of the FMM in inaccurate partial static environment for USVs. The algorithm had a path re-planning capability and can effectively plan a new path by largely maintaining the existing path.

It is worth mentioning that although the above mentioned research effort has successfully achieved using the FMM in various applications with improved performance, they considered only a static environment or a simple dynamic environment. To implement the FMM for USV navigation, a dynamic path planning problem, which is mainly focused on solving the collision avoidance with moving obstacles, it is necessary to address the possibility of a large number of moving vessels in the area of interest. In [9] and [10], even though the algorithm is capable of avoiding obstacles collision risk was not assessed.

Another important issue associated with the USV when dealing with the dynamic path problem is the uncertainty of the position of the moving obstacles [13]. Detecting other ships' movements AIS, RADAR, LIDAR, etc. can be used but these systems are not infallible. For instance, it has been reported that AIS information is sometimes partially or fully interrupted meaning it is not received properly [14], making dependence impractical. It is therefore necessary to make the navigation system capable of predicting the trajectory of the moving ship (dynamic obstacle) based on current information when data is lost or delayed, and to keep searching for the path according to the predicted trajectory until the data connection can be re-established.

In this paper, a new predictive path planning algorithm is proposed to specifically address the dynamic path planning of USVs using a predictive methodology. The algorithm consists of two functionalities, i.e. the collision risk assessment (CRA) function and the path planning function. It is assumed that the dynamic information of moving ships, such as velocities and instantaneous positions, can be obtained by using on-board sensors or navigation devices. Based on such information, the CRA first employs the Kalman Filter (KF) algorithm to predict the movements of moving ships in defined time steps, and assesses the collision risks. If it is required to avoid the moving ship, a safe area around the ship will be generated to assist with collision avoidance. When planning the trajectory, a new method named weighted FMS algorithm is used which follows the FMS method but with modifications allowing the generated path to be optimised.

The organisation structure of this paper is as follows. Section 2 first introduces the fundamentals of the FMM and explains its implementation for the path planning problem. Then, the proposed weighted FMS method is described and compared against the conventional FMS. Section 3 explains the KF based position updating and estimation and the uncertainty modelling of moving ships. The detailed predictive path planning algorithm is described in Section 4 with simulation results presented in Section 5. Section 6 concludes the paper with discussion of future work.

2. PATH PLANNING AND THE FAST MARCHING METHOD

2.1. Problem formulation

Consider a USV navigating in a 2D configuration space *C*, which contains two different spaces, i.e. the free space (C_{free}) and the obstacle space (C_{obs}). The start point and end point are denoted as $\boldsymbol{s} \in C_{free}$ and $\boldsymbol{z} \in C_{free}$. The USV path planning problem can be formalised as to find a continuous collision free path with optimal cost:

$$\tau : [0,1] \rightarrow C_{free}, \quad \tau(0) = \boldsymbol{s} \text{ and } \tau(1) = \boldsymbol{z} \tag{1}$$

The cost here can be any measurements such as distance, safety or energy consumption. To calculate the path cost, a local cost function representing the cost of point $\boldsymbol{x}$ is first defined as $w(\boldsymbol{x})$. Hence, the optimal path cost $\mathrm{D}(\boldsymbol{z})$ indicating the path cost from point $\boldsymbol{s}$ to point $\boldsymbol{z}$ is defined as:

$$\mathrm{D}(\boldsymbol{z}) = \min_{\tau} \int_0^1 w(\tau(t)) d(w(\tau(t))) \tag{2}$$

2.2. Eikonal equation and the fast marching method (FMM)

The analytical solution of (2) is difficult to find however it is proved in [15] that $\mathrm{D}(\boldsymbol{z})$ can be approximated by using the viscosity solution of the Eikonal equation:

$$||\nabla(D(\boldsymbol{z}))|| = w(\boldsymbol{z}) \tag{3}$$

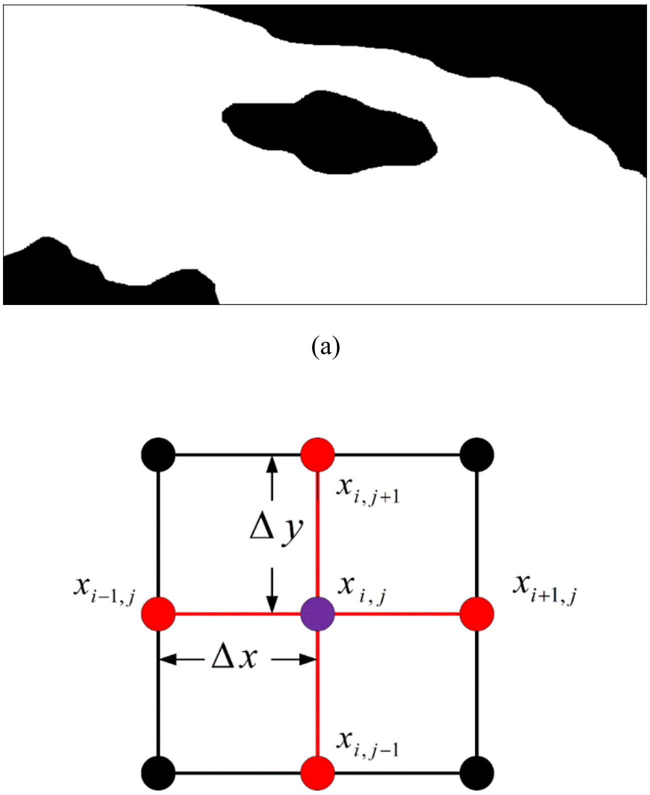
To solve the Eikonal equation, the FMM method can be used. The details of using the FMM in path planning are illustrated as follows. Assume the original environment map ($\boldsymbol{W}_o$) is represented in Figure 2(a). For path planning, the map is first discretised into a 2D Cartesian grid $\boldsymbol{W}_g$. The grid is a binary distance map where each grid in collision free space has a value 1 and grids in an obstacle area have a value 0.

Given $x_{i,j}$ is one of the grid points, the cost $D(x_{i,j})$ needs to be solved. The neighbour of $x_{i,j}$ is the point set containing four elements $x_{i+1,j}$, $x_{i-1,j}$, $x_{i,j+1}$ and $x_{i,j-1}$ (shown in Figure 2(b)). By using the Upwind Finite Difference approximation scheme, (3) can become:

$$\max\left(\frac{D(x_{i,j})-\min\left(D(x_{i+1,j}),D(x_{i-1,j})\right)}{\Delta x},0\right)^2 + \max\left(\frac{D(x_{i,j})-\min\left(D(x_{i,j+1}),D(x_{i,j-1})\right)}{\Delta y},0\right)^2 = w(x_{i,j}) \tag{4}$$

where Δx and Δy are the grid sizes in x and y directions respectively.

Because the cost function $w(x_{i,j})$ represents the distance between the current point and its neighbouring point, the total cost $D(x_{i,j})$ only represents the total distance so the final generated path has the minimum distance cost. When running the algorithm, the cost of start point is first assigned with an initial value (0 in this example). Then the FMM iteratively solves (4) to update the distance cost for each neighbouring point to the start point. Once all neighbour points are updated, the point with smallest cost will be selected as the new start point and the FMM will continue to calculate the cost associated with the new neighbouring points. Such an updating scheme is the same as the Dijkstra's updating method and will terminate until all points have been assigned with costs.

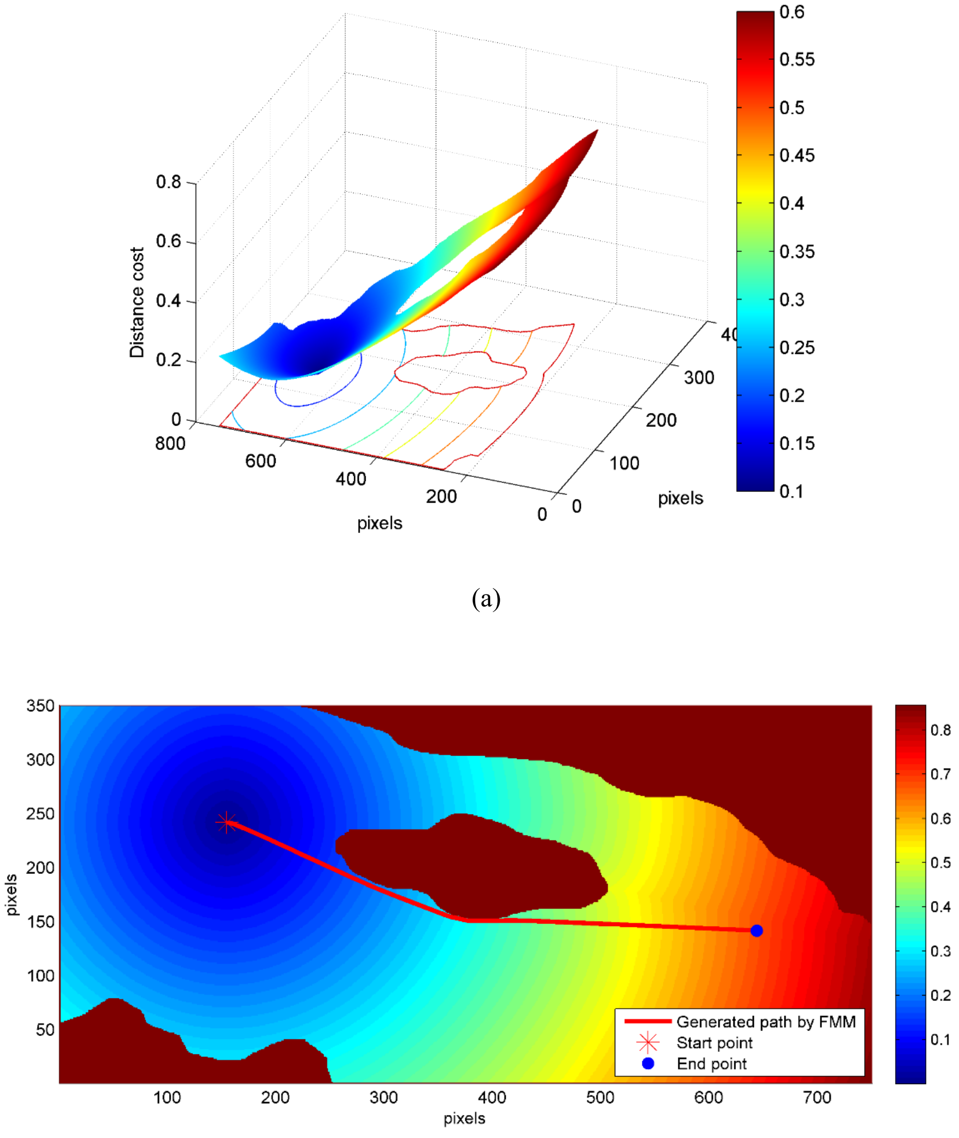


(a)

(b)

Figure 2. (a) The original environment map $\boldsymbol{W}_o$, where black area stands for obstacles and has binary value of 0, white area stands for collision free area and has binary value of 1. (b) Grid point $x_{i,j}$ and its neighbours. The grid map has four connectivity; hence, each point has four neighbour points. Δx and Δy are the grid sizes in x and y directions respectively.

The updating result is shown in Figure 3(a), where a potential field has been created. The potential value at each point indicates local distance to the start point with the lowest potential located at the starting point with cost value 0. Potential values at other points increase with distance, with the exception of obstacles for which the potentials are infinite. The highest potential value is set at the end point, which indicates that the end point has the farthest distance to the start point. Note that potential values are only indications of distance instead of actual distance values. Compared with the potential field generated by other methods such as the APF, the potential field of the FMM has features of global minimum, which avoids local minima problems and increases the completeness of algorithm. Based on the potential field obtained, the gradient descent method is then applied to find the shortest collision-free path (shown as a red line in Figure 3(b)) by following the gradient of the potential field.



(a)

(b)

Figure 3. (a) The potential field generated by the FMM. Potential value represents distance cost. (b) The path generated by following the gradient of the potential field.

2.3. Weighted fast marching square method

One of the problems associated with path planning when directly using the FMM is that the generated path gets too close to obstacles. 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 distant away from obstacles.

To tackle this problem the FMS method proposed by [9] for indoor mobile robots is used and modified in this paper. A difference between the FMM and the FMS is that the FMS takes the safety cost of path into consideration by running the FMM twice. When running the FMM for the first time, the algorithm chooses all the points representing obstacles as start points and generates a safety map $\boldsymbol{W}_s$ (Figure 4(a)). In the $\boldsymbol{W}_s$, each point has been given a value ranging from 0 to 1, and the further the distance to the obstacle is, the higher the value. 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 value. After the generation of $\boldsymbol{W}_s$, the FMM will be used again on the $\boldsymbol{W}_s$ to search for the safest path. However, such method is done at the sacrifice of total distance cost since it can only maximise the safety of path.

Hence, in this paper, a weighted synthetic map $\boldsymbol{W}$ is proposed to replace the $\boldsymbol{W}_s$ in the FMS to improve the optimisation result. The $\boldsymbol{W}$ is calculated as the weighted combination of distance map $\boldsymbol{W}_d$ and safety map $\boldsymbol{W}_s$:

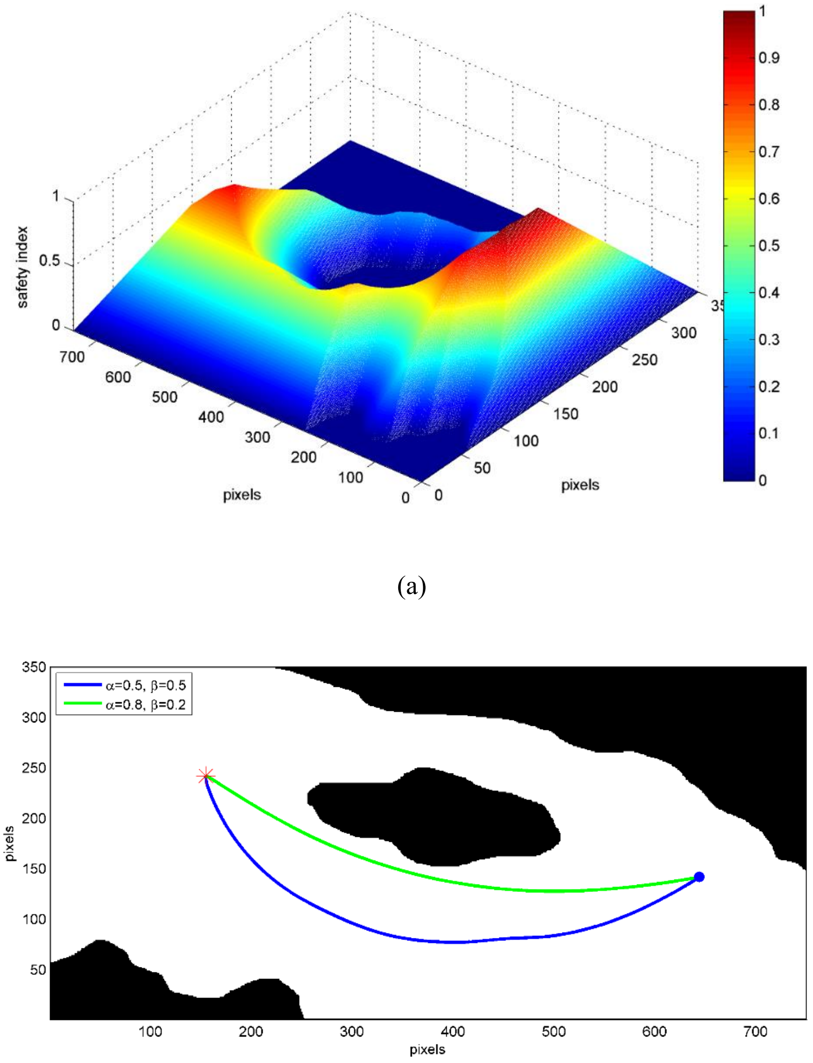
$$\boldsymbol{W} = \alpha * \boldsymbol{W}_s + \beta * \boldsymbol{W}_d \tag{5}$$

and

$$\alpha + \beta = 1 \tag{6}$$

where α and β are two weighting values called distance map weighting and safety map weighting. By adjusting these two weighting values, the algorithm can generate the path according to different task requirements. For example, in the case that USV is equipped with limited energy, minimum total distance is the primary requirement; hence, distance map weighing α is larger than safety map weighting β when synthetic map $\boldsymbol{W}$ is generated. It should be noted that apart from these two maps, other maps specifying task requirements can also be added into the synthetic map $\boldsymbol{W}$ to increase the algorithm's capability.

Then, over the synthetic map $\boldsymbol{W}$, the FMM is run for the second time to search for the final trajectory. Figure 4(b) shows the path generated by selecting different weightings. The path in blue has two equal weightings, which makes the trajectory have both a balanced distance and safety cost and it therefore chooses to stay further away from the central island; whereas the path in green has more emphasis on distance cost in achieving the shorter distance with the cost of being closer to the obstacle.



(a)

(b)

Figure 4. (a) Safety map $\boldsymbol{W}_S$. In the map, the higher the value is, the safer the area will be. (b) Two different paths by using different weighting values.

3. STATES PREDICTION AND COLLISION RISK ASSESSMENT OF MOVING SHIPS

It is assumed in this paper that moving ship's information is obtained via AIS, which is one of the most effective and widely used devices in maritime navigation. The broadcasting mechanism of AIS is illustrated in Figure 5. In practice, AIS automatically broadcast information at regular intervals with the transmission consisting of two sections, i.e. the data receiving point and the data waiting period. At the data receiving point, information including velocity, position and navigational status can be obtained; whereas during the data waiting period, no information is exchanged. The length of the data waiting period is various based on different AIS classes, and for a class B AIS device default waiting time is 30 seconds.

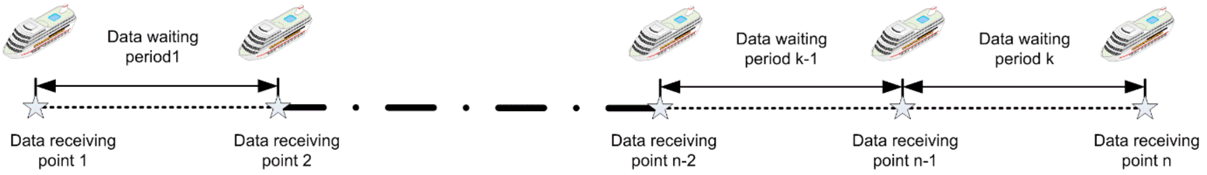


Figure 5. AIS broadcasting mechanism. The transmission is consisting of two different periods. At the data receiving point, AIS information is received and updated. During data waiting period, AIS information is in absence.

It should be noted that a certain degree of information uncertainty is associated with AIS transmission. This is because at each data receiving point, information is inaccurate as signal noise is apparent, which is especially evident in harsh environment. Also, during the data waiting period, such uncertainty is more serve as there is no awareness of moving ships. Therefore, to increase the accuracy of moving ship's information, a Kalman Filter (KF) based predictor is proposed in this paper. There are two processes composing the predictor, i.e. the updating process and the estimating process (Figure 6). The updating process works at each data receiving point by using the KF algorithm to filter information noise. When AIS data is not received, the estimating process based on the KF's prediction function will be employed to continue provide an estimation of moving ships. Therefore, by using the proposed position predictor, it is possible to be aware of the moving ship's situation in anytime, which facilitates the path planning algorithm to determine an optimal trajectory. Specific explanations of such predictor will be provided in following sections.

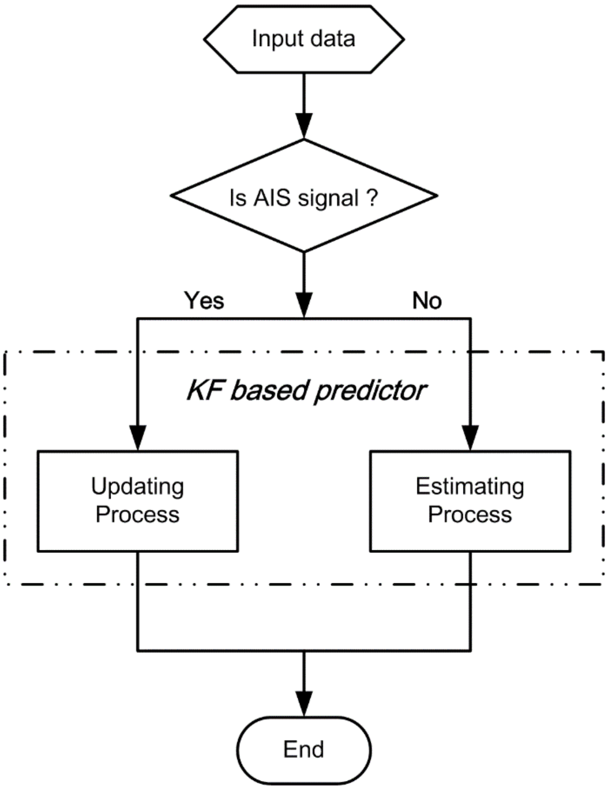


Figure 6. KF based predictor algorithm.

3.1. Kalman filter algorithm

As a linear recursion algorithm, the KF is able to estimate the state of a dynamic system by following the least mean square error principle. Compared with other filters such as the Bayesian estimation, the KF has faster computational speed, which makes it more suitable to be implemented for USV path planning, where a quick decision making system is always preferred. In addition, the KF employs a recursive principle only requiring the current and previous time step states to make the estimation, which makes it as a resource efficient algorithm suitable for online real time application. The recursive estimation process of the KF mainly involves two different equations, i.e. the state transition equation and the observation equation:

$$X_{k+1} = A_k X_k + B_k u_k + w_k, \tag{7}$$

$$Z_k = H_k X_k + v_k \tag{8}$$

where in (7) X_k is the system state vector at time k, A_k is the state transition matrix and B_k is the control input matrix applied on the control input u_k. In (8), Z_k is the system observation vector at time k and H_k is the observation matrix. w_k and v_k are the transition noise and observation noise respectively.

The KF algorithm is summarised in Figure 7, which consists of two main processes, i.e. the Prediction Process and the Update Process. The algorithm starts by taking the inputs of the initial estimation of the system $\hat{x}_0$ and the associated system uncertainty P_0. Based on these two values, the Prediction Process is first employed to predict the system state ($\hat{x}_{k|k-1}$) and the according system uncertainty ($P_{k|k-1}$) for next time step. Then, a KF gain (K_k) can be calculated and used as the input for the Update Process together with system measurement Z_k. The Update Process updates and corrects $\hat{x}_{k|k-1}$and $P_{k|k-1}$ to $\hat{x}_{k|k}$ and $P_{k|k}$ with improved accuracy and passes them back to Prediction Process as new inputs for next time step.

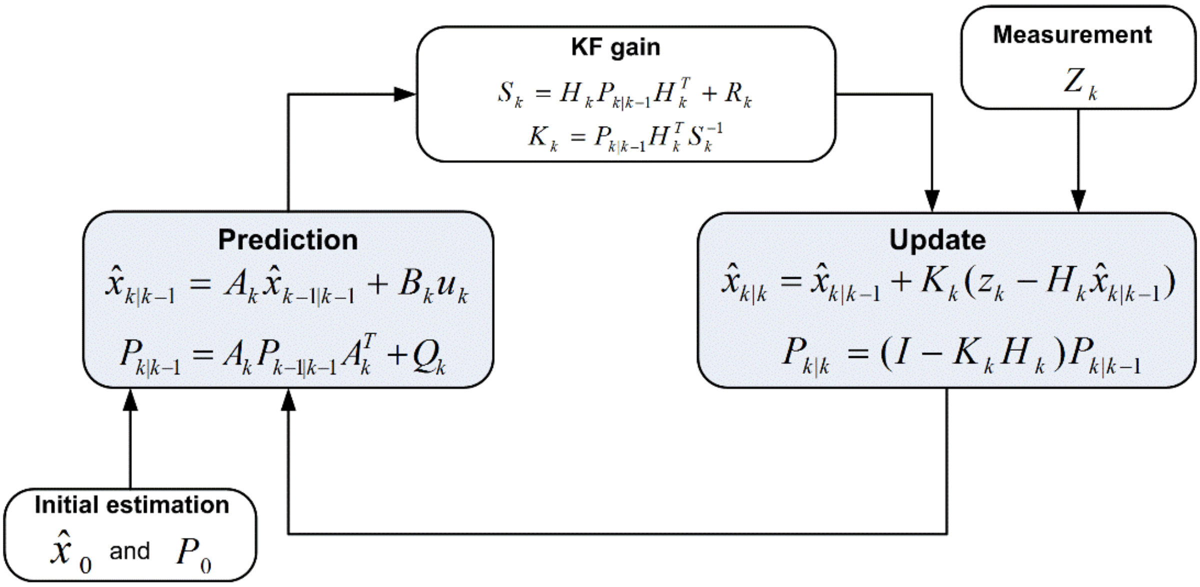

KF gain
$S_k = H_k P_{k|k-1} H_k^T + R_k$
$K_k = P_{k|k-1} H_k^T S_k^{-1}$
Measurement
Z_k
Prediction
$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + B_k u_k$
$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k$
Update
$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1})$
$P_{k|k} = (I - K_k H_k) P_{k|k-1}$
Initial estimation
$\hat{x}_0$ and P_0

Figure 7. Kalman filter recursive process.

3.2. Position update of moving ships

When implementing the KF for a position update in the maritime environment, the system state vector can be expressed as:

$$X = [x, y, v_x, v_y]^{\,T} \tag{9}$$

where x and y represent the position in x direction and y direction, v_x and v_y are velocities in x and y direction. The movement of the ship is assumed to belong to the constant velocity model (CVM), which is one of the most common models used for marine vessel. The movement equation is:

$$X_{k+1} = A_k X_k + G_k a_k \tag{10}$$

with

$$A_k = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & \Delta T & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

and

$$G_k = \begin{bmatrix} \frac{\Delta T^2}{2} & 0 \\ 0 & \frac{\Delta T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \tag{12}$$

where ΔT is the sampling period and a_k is defined as:

$$a_k \sim N(0, \sigma^2) \tag{13}$$

which is a zero-mean white noise to model the uncertain accelerations in x and y directions. Note that there is no B_k term in (10) as the control input of the moving ship is unknown. Instead α_x and α_y with small values are put into the system to model the effect of the unknown input, which cause small deviation for the velocities in according direction; the ship modelled by (10) will have nearly constant velocity and is hence called the CVM. Now, (7) can be rewritten as:

$$X_{k+1} = A_k X_k + w_k, \tag{14}$$

with

$$w_k \sim N(0, \boldsymbol{Q}) \tag{15}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} \frac{\Delta T^4}{4} & 0 & \frac{\Delta T^3}{2} & 0 \\ 0 & \frac{\Delta T^4}{4} & 0 & \frac{\Delta T^3}{2} \\ \frac{\Delta T^3}{2} & 0 & \Delta T^2 & 0 \\ 0 & \frac{\Delta T^3}{2} & 0 & \Delta T^2 \end{bmatrix} \sigma^2 \tag{16}$$

Because only the position coordinates of the moving ship need to be estimated, the observation vector Z_k is now in the form of $[x\ ,y]^T$ and (8) is written as:

$$\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix} + v_k \tag{17}$$

where v_k is used to simulate the measurement noises as a Gussian noise with zero mean and standard deviation σ_z. Based on (14) and (17), the KF can be used at each position update point to track moving ships.

3.3. Position prediction of moving ships

During the transmission interval (the data waiting period), the perception of the moving ship's positions is still of importance to ensure the safe navigation of USV. However, since there is no accurate information established during this period, the USV can only make a prediction of ship's travel information, and therefore only the Prediction Process of the KF algorithm loop (See Figure 7) is repeatedly executed to calculate the possible positions of moving ship based on latest known information. It should be noted that such prediction process has no corrections and will possibly accumulate large errors if the prediction time is too long. Therefore, it is important to carefully select the prediction time. Additionally, as new moving ship's information is received the KF loop continues to be employed to compensate for the generated errors.

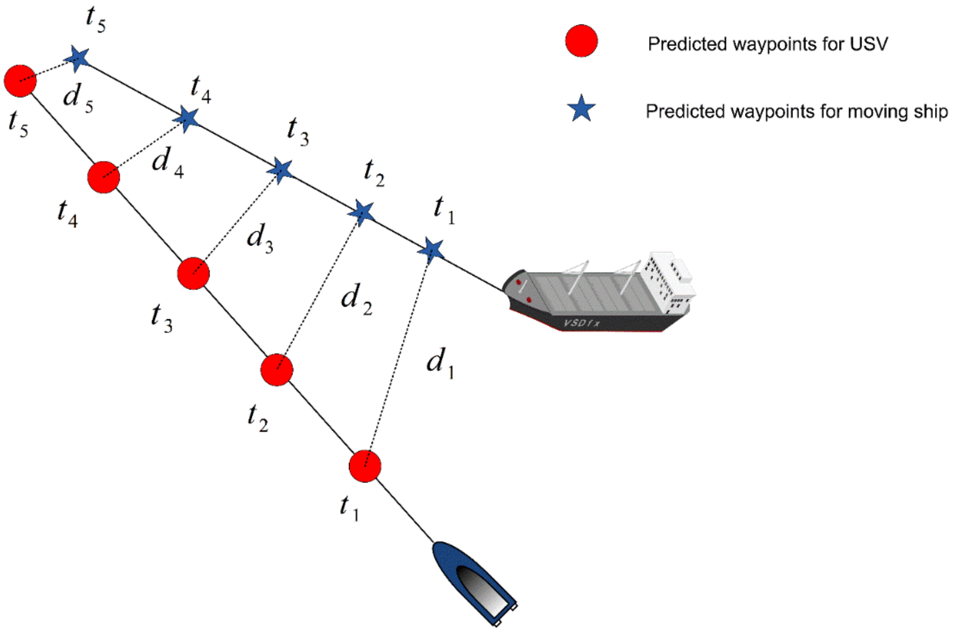
3.4. Collision risk assessment and modelling of moving ships

Once the path of the moving ship in next time periods have been predicted, to assess the collision risk the trajectory of USV itself will need to be estimated. Assume that USV is navigating in a 2D space and has access to its own travel information such as current position, velocity, heading angle and turning rate; according to the kinematic equations of USV, the nonlinear estimation model of USV itself is established as:

$$\begin{cases} x(k+1) = x(k) + \Delta t * v(k)\cos\varphi(k) \\ y(k+1) = y(k) + \Delta t * v(k)\sin\varphi(k) \end{cases} \tag{18}$$

$$\varphi(k+1) = \varphi(k) + \Delta t * \alpha(k) \tag{19}$$

where $x(k)$ and $y(k)$ represent the position in x and y directions at time step k, $v(k)$ is the velocity magnitude at time step k with heading angle as $\varphi(k)$ and $\alpha(k)$ is the heading angle turning rate. Then, as shown in Figure 8 based on these two predicted paths, the smallest distance between them can be calculated. If this distance is less than the predefined safety distance two ships will have the possibility of clashing, hence appropriate collision avoidance manoeuvres are needed.



Predicted waypoints for USV
Predicted waypoints for moving ship
t_5
d_5
t_4
d_4
t_3
d_3
t_2
d_2
t_1
d_1

Figure 8. Collision risk assessment. Minima distance between two predicted paths are calculated. Such distance will be compared with predefined safety distance to determine the possibility of collision.

To prevent collision with dynamic obstacles, most studies in path planning research have adopted the concept of a ‘safety area’ (SA) (‘ship domain’ in marine vessels collision avoidance) to model the area from which all other vehicles are prohibited. The shape of such an area is normally circular and the centre of the area is located on the obstacle’s instantaneous position. However, in USV path planning, a circular shape area is not always the most practical, especially when a ship is travelling at high speed, which tends to hold greater risk at the bow. It is more realistic to assign the shape of safety area of a moving ship according to its velocity.

Another important feature of SA is the area which should be consistent with the static obstacles’ representation in the planning space so that a reasonable map $\boldsymbol{W}$ can be constructed and used by the algorithm to search for the path. When dealing with the dynamic obstacle avoidance problem, the whole planning time period is discretised into several infinitesimal time periods, Δt, and during each time step the moving ship is regarded as a ‘static’ obstacle. Hence, the SA should be formed to have the dynamic safety map $\boldsymbol{W}_d$ indicating the degree of collision risk within the area i.e. in same way the safety map ($\boldsymbol{W}_s$) maps representing static obstacles, such that dynamic obstacles could be integrated with the static obstacles to generate a synthetic safety map.

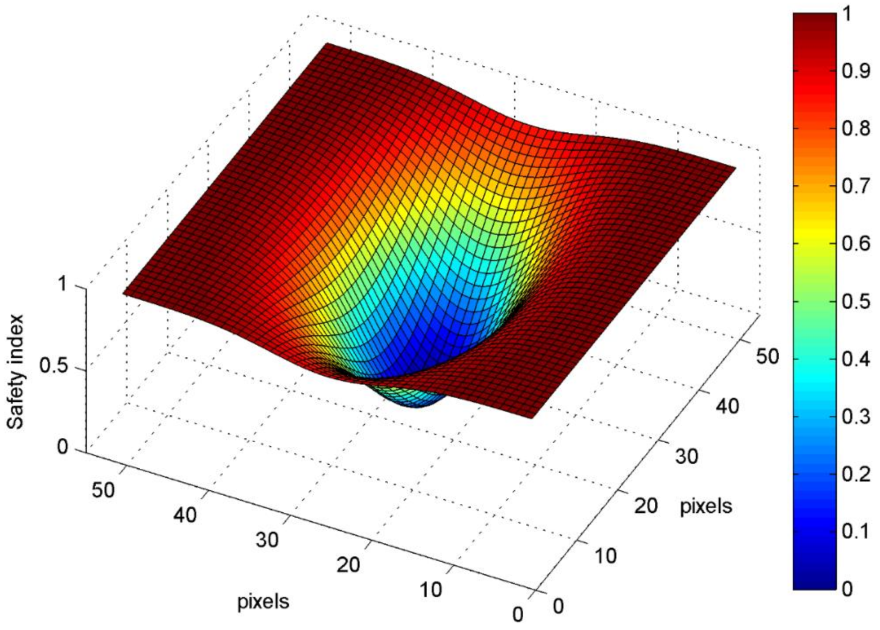
Therefore, the two-dimensional Gaussian distribution is selected to represent SA with the form $\mathbf{p} \sim \mathcal{N}(\mu, \Sigma)$. The mean vector μ is set to be zero so that the Gaussian distribution can be centred at the position of the moving ship. The covariance matrix Σ is the diagonal matrix with non-diagonal elements to be zero. The shape of Gaussian distribution is determined by two diagonal elements $\sigma_{11}$and σ_{22}, and they will change according to the ship’s velocity as:

$$\sigma_{11} = \begin{cases} velocity * AreaScalar, & if\ velocity\ \geq V_{min}, \\ V_{min} * AreaScalar, & if\ velocity\ < V_{min} \end{cases} \tag{20}$$

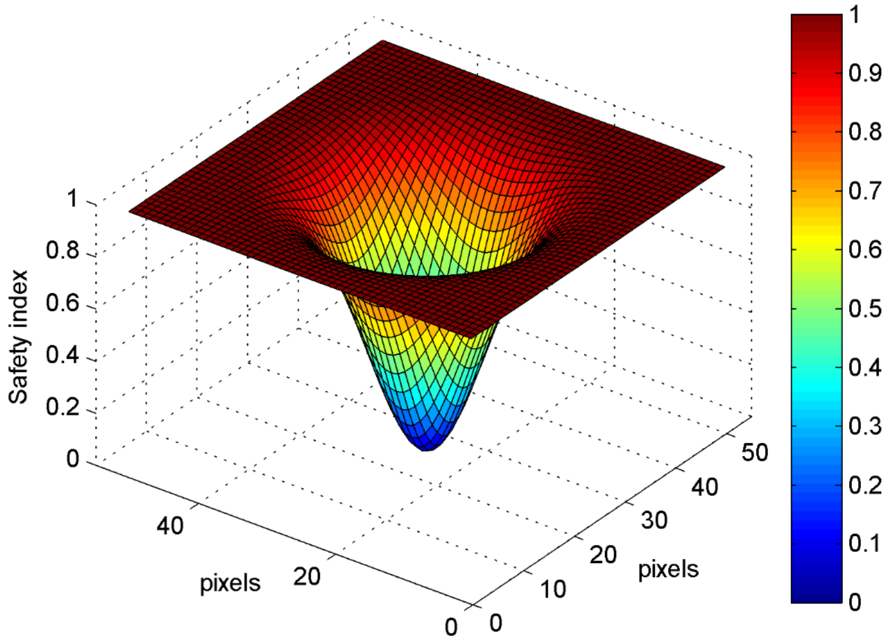
$$\sigma_{22} = \begin{cases} V_{max} * AreaScalar, & if\ V_{max} < velocity \\ velocity * AreaScalar, & if\ V_{min} \leq velocity\ \leq V_{max} \\ V_{min} * AreaScalar, & if\ velocity\ < V_{min} \end{cases} \tag{21}$$

where $AreaScalar$ is a scalar factor to control area's size, V_{min} and V_{max} are two thresholds to regulate the shape of the area. It can be deduced that when the ship is moving with low velocity, a near circular shape will be assigned to it. For example, if the ship is travelling with a velocity less than V_{max}, then σ_{11} and σ_{22}belong to the same equation according to (20) and (21). The values of these two elements increase according to the velocity, which further makes the radius of the generated SA increase relatively to the velocity's magnitude and direction. If the ship's velocity is higher than V_{max}, an elliptical shape is generated to model the safety area. In this time the major axis (σ_{11}) is increasing proportionally to the velocity; whereas the minor axis (σ_{22}) is settled. Once the area has been determined, the centre of it will be placed at the moving ship's position(x_i, y_i), and if the elliptical is in use, the long axis will be in the direction of velocity with the short axis orthogonal to it.

It should be noted that the $\boldsymbol{W}_d$ constructed by conventional two-dimensional Gaussian distribution is still different from $\boldsymbol{W}_s$. This is because in two-dimensional Gaussian distribution, the highest probability density is located at the centre point. This means the points close to the moving ship have higher safety values, which is the opposite to the safety map $\boldsymbol{W}_s$, where high safety index values are located away from obstacles. Therefore, to address it, a reverse smoother function $\boldsymbol{W}_d = -\boldsymbol{W}_d + 1$ is used to scale the $\boldsymbol{W}_d$ to the same range as $\boldsymbol{W}_s$'s. In Figure 9(a) and 9(b), two different safety areas for high speed ship and low speed ship are represented respectively. It can be observed that the safety index within the area is the same to $\boldsymbol{W}_s$'s, i.e. the centre area (points closer to ships) has a lower index value than the edge.



(a)



(b)

Figure 9. Two different safety areas for (a) high speed ship and (b) low speed ship. Low value means the area is more dangerous.

4. PREDICTIVE PATH PLANNING FOR USV

Based on the weighted FMS method and the collision risk assessment algorithm introduced in the previous two sections, the predictive path planning algorithm for USV can now be developed. Compared with other maritime path planning algorithms, the algorithm in this work has the following features:

- **Completeness**: As mentioned in the introduction section, some path planning algorithms developed by using the evolutionary searching algorithm suffer from the problem of searching incompleteness, which means the algorithm may fail to find a path in complex environment. The algorithm developed in this paper is based the FMM, which adopts a deterministic searching scheme ensuring a path can always be found as long as it exists.
- **Fast computational time**: the FMM, as the base method for this algorithm, is fast in dealing with searching problem due to its low computation complexity ($\mathcal{O}\big(\mathrm{N}\log(\mathrm{N})\big)$ for a grid map having total grid number of N). Such a feature is especially ideal for practical path planning, where a fast decision making process is preferred to ensure safety.
- **On-line planning scheme**: The algorithm in this paper implements a true on-line reactive planning scheme. An optimal path is first calculated as the guidance trajectory, and USV will keep tracking it as long as no new collision risk is determined by the prediction algorithm. Such schemes will, to a large extent, maintain navigation route and reduce unnecessary manoeuvres.

The detail of the algorithm is described here with flow chart shown in Figure 10:

1. The algorithm first takes in the navigation map, where static obstacles have been clearly represented and stored as an original binary map $\boldsymbol{W}_o$. Such a map can be obtained through commercial charts, such as marine navigation charts. Also, advanced sensor technology, such as the Simultaneous Localisation and Mapping (SLAM), can be used to construct the map of the unknown environment while the USV is navigating.
2. Based on the map received, the safety map $\boldsymbol{W}_s$ will be first generated and combined with $\boldsymbol{W}_o$ to have an initial map $\boldsymbol{W}_{ini}$. A collision free path $\boldsymbol{\tau}_{ini}$ in such environment will be sought by using the FMM and stored as the guidance route. By following $\boldsymbol{\tau}_{ini}$, the USV will start to proceed towards the end point.
3. While following the path, the USV will simultaneously monitor the positions and velocities of itself as well as other moving ships. The prediction algorithm will now be called to estimate the trajectory of the USV and other ships in next few time steps, and determine if there will be a collision within such time period.
4. If the collision risk exits, a new path should be generated. The dynamic safety map $\boldsymbol{W}_d$ will be constructed around each moving obstacle.
5. The $\boldsymbol{W}_d$ will then be merged with the safety map $\boldsymbol{W}_s$ as well as the original binary map $\boldsymbol{W}_o$ to generate a new synthetic map $\boldsymbol{W}$. Based on $\boldsymbol{W}$, a new path $\boldsymbol{\tau}_{new}$ will be sought by applying FMM again. $\boldsymbol{\tau}_{new}$ is the optimised trajectory without colliding with both static and dynamic obstacles in time step t, and the USV will follow it until the next waypoint has been reached.
6. When the new waypoint is reached, the algorithm will determine if it is the final target point. If it is not, the algorithm will jump back to step 3 and move towards the next waypoint.

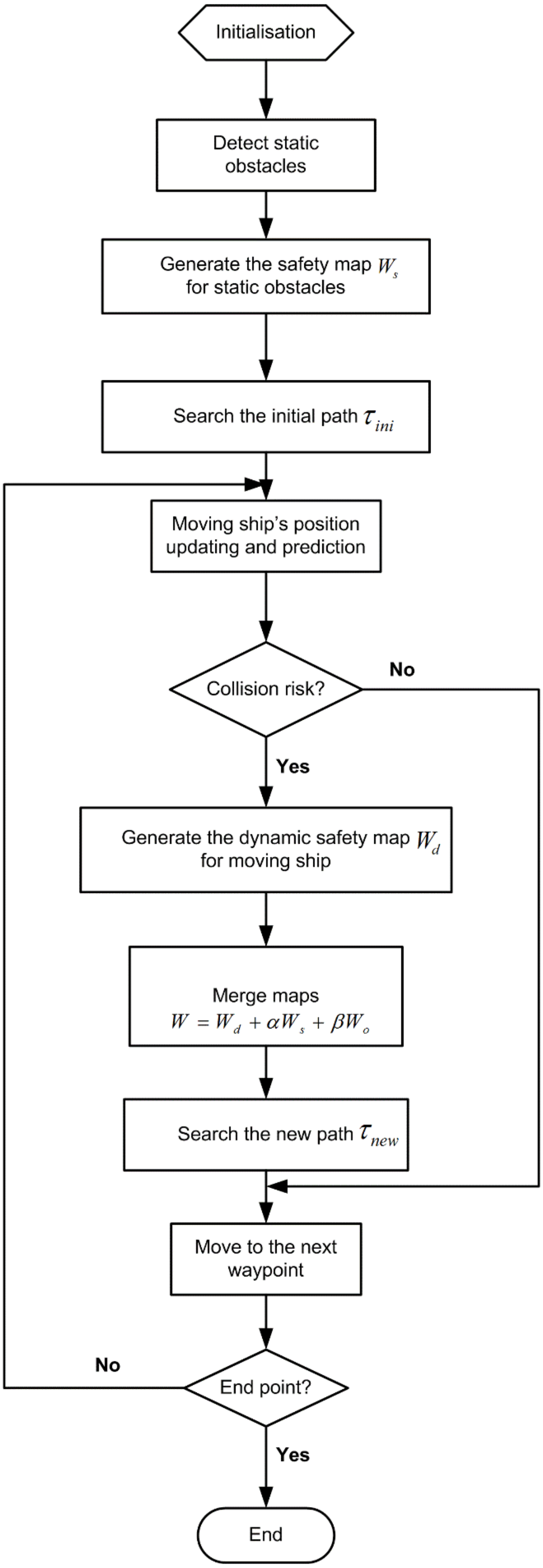

Initialisation
Detect static obstacles
Generate the safety map W_s for static obstacles
Search the initial path τ_{ini}
Moving ship's position updating and prediction
Collision risk?
No
Yes
Generate the dynamic safety map W_d for moving ship
Merge maps $W = W_d + \alpha W_s + \beta W_o$
Search the new path τ_{new}
Move to the next waypoint
No
End point?
Yes
End

Figure 10. Algorithm flow chart of the predictive path planning.

5. SIMULATIONS AND DISCUSSIONS

In this section, simulations have been carried out in three different cases to validate the algorithm proposed. First, the proposed predictor is used to process a series of inaccurate position information based on real AIS data from Marine Traffic service available at http://www.marinetraffic.com/. Then, to illustrate the capability of the predictive path planning algorithm, simulations have been carried out in two different testing environments. The aim of the first test is to validate the fundamentals of the predictive path planning algorithm; hence, a self-constructing area is used as the simulation environment with one moving obstacle involved. To further test the capability of the algorithm dealing with a practical navigation problem, the second test is run in a real maritime environment. Multiple moving obstacles with different transverse velocities have been added into the environment to construct a complex traffic situation. The algorithm has been coded in Matlab 2013(b) and simulations are run on the computer with a Pentium i7 3.4 Ghz processor and 4Gb of RAM.

5.1. Simulation of the proposed predictor by using AIS information

The aim of this simulation is to test the capability of the proposed predictor processing AIS information. Real AIS information from *Marine Traffic* online service has been used as the input data. In Figure 11(a), a vessel is travelling to Portsmouth harbour through a channel with the speed of 9.2 knots and the course of 334°.

5.1.1. Coordinate transformation of AIS position information

The AIS's position information is coded in longitude and latitude based on WGS84 (World Geodesic System 1984); whereas the speed of the vessel is determined in knots. To facilitate the implementation of the KF, it is important to first convert the longitude and latitude position onto the Cartesian coordinate. Assume that the vessel's longitude and latitude coordinate is (λ, φ), the origin point of Cartesian coordinate is (λ_o, φ_o) and the vessel's Cartesian coordinate is (x, y), the coordinate transformation can be achieved by using the Gauss-Kruger Projection as:

$$x = X + Nsin\varphi cos\varphi\Delta\lambda^2[\frac{1}{2\rho^2} + \frac{cos\varphi \Delta \lambda^2}{24\rho^4} \cdot (5 - tan^2\varphi + 9\eta^2 + 4\eta^4) + \frac{cos^4\varphi \Delta \lambda^4}{720\rho^6} \cdot (61 - 58tan^2\varphi + tan^4\varphi)] \tag{22}$$

$$y = Ncos\varphi[\frac{\Delta\lambda}{\rho} + \frac{cos\varphi^2}{6\rho^3} \cdot (1 - tan^2\varphi + \eta^2) \cdot \Delta\lambda^3 + \frac{cos^4\varphi}{120\rho^5} \cdot (5 - 18tan^2\varphi + tan^4\varphi + 14\eta^2 - 58\eta^2 tan^2\varphi) \cdot \Delta\lambda^3] \tag{23}$$

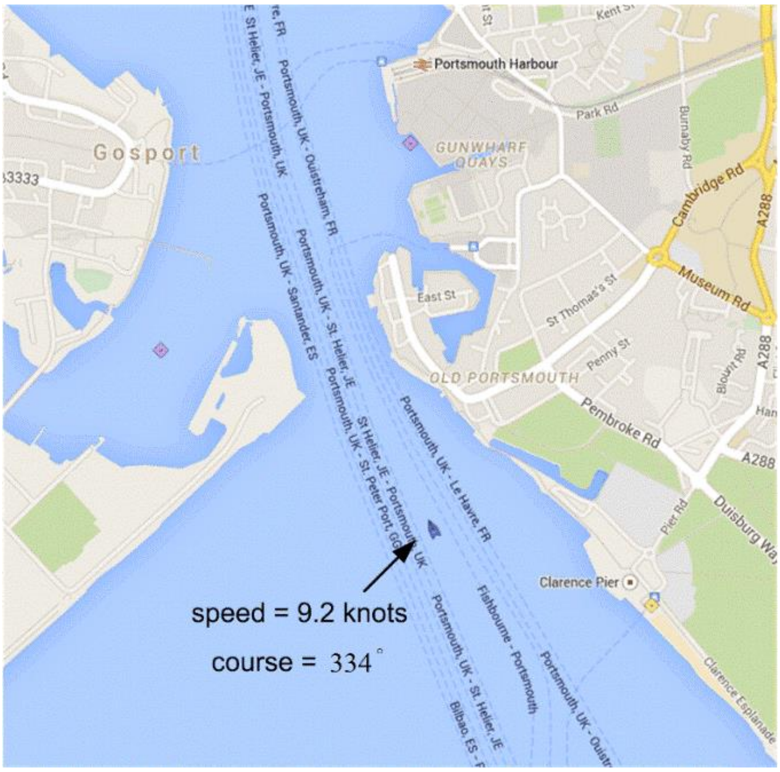
where $\Delta\lambda = \lambda - \lambda_o, X = X(\varphi) - X(\varphi_o)$ and other parameters can be referred to [16].

5.1.2. Simulation results

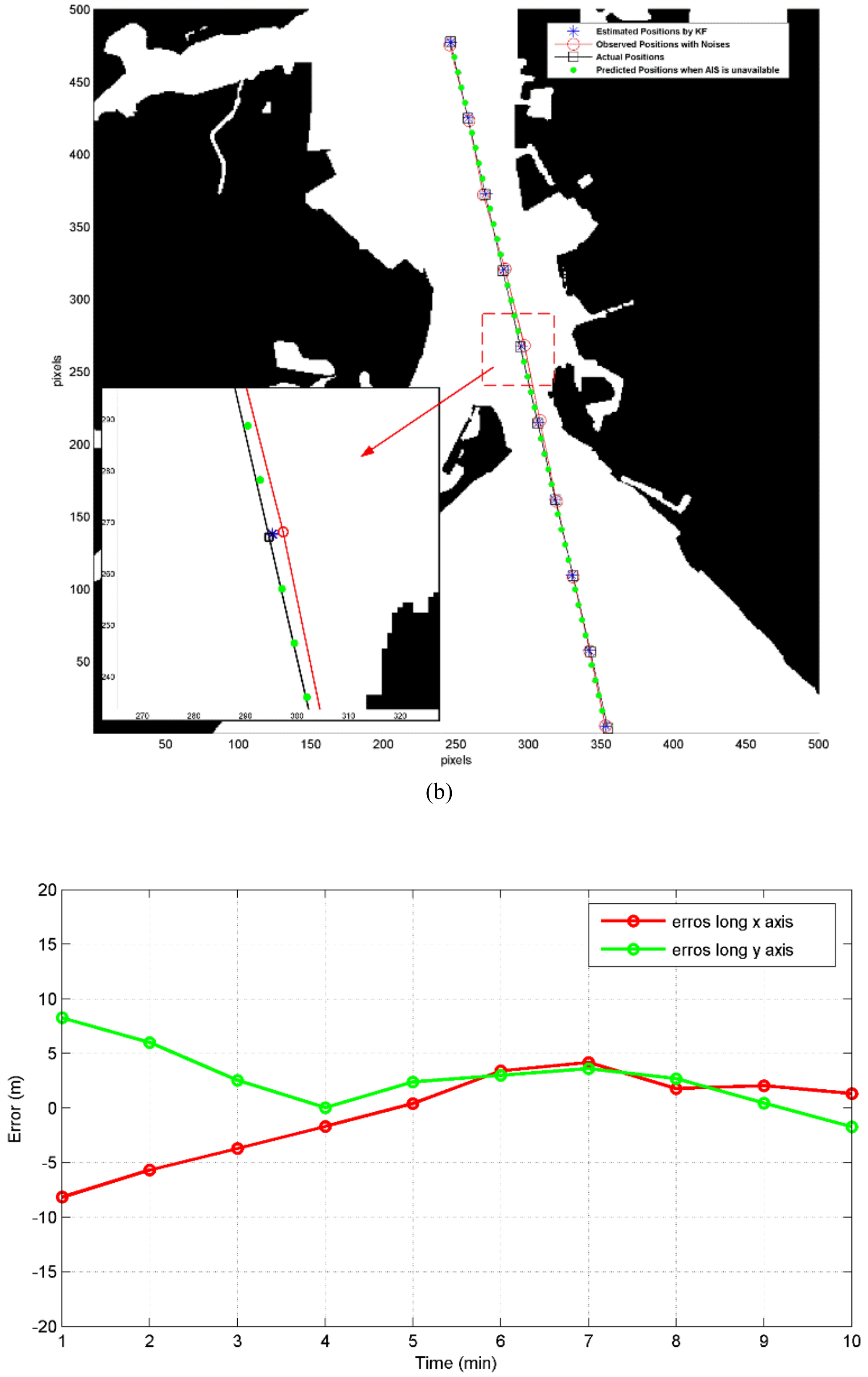
The simulation area has first been converted into a binary map, which has the dimension of 500 pixels * 500 pixels representing a 2.5 km * 2.5 km area (1 pixels = 5 m). The vessel keeps to an unaltered course thereby forming a straight line trajectory. The AIS transmission interval is 1 minute, during which the vessel's navigation information is unknown. To simulate the fact that AIS is inaccurate due to transmission disturbance, the observation noise has been modelled as a white Gaussian noise sequence with a covariance matrix $\boldsymbol{R} = 1.5^2\boldsymbol{I}_{2\times2}$. In addition, the system state disturbance is also modelled as a zero-mean white Gaussian noise with a covariance matrix $\boldsymbol{Q} = 0.01^2\boldsymbol{G_k}\boldsymbol{G_k^T}$.

Simulation results showing how vessel's position information is updated and predicted in 10 minutes are presented in Figure 11(b). Since the simulation covers a time period of 10 minutes, 10 consecutive data receiving points are plotted with the vessel's true position drawn as black square markers and the inaccurate observed position as red circle makers. Positions obtained by using the KF are plotted with blue star makers. From an enlarged observation in Figure 11(b), it can be observed that the KF can evidently increase the accuracy by providing a position closer to the true position point. Such observation can further be verified in Figure 11(c), which shows the position errors by using the KF in x and y axis. In the beginning, errors in both x and y axis are as big as 9 metres. However, after the KF is iteratively employed, the errors are decreasing, and after 3 minutes they become stable under 4 metres, which demonstrates that accurate positions can be obtained by using the proposed algorithm.

During each AIS transmission interval, the vessel's information is unknown, and the predictor uses previous updated position information to calculate the possible positions of the vessel. It is assumed in this paper that the predictor makes the predictions every 12 seconds, and the predicted results are plotted with green dot makers in Figure 11(b). It can be observed that the predicted positions stay close to the true trajectory of the vessel, which means that the predictor is able to continue to provide the vessel's positions even though the AIS data is absent.



(a)



(b)

(c)

Figure 11. Simulation results of the proposed predictor processing AIS information. (a) AIS information with the vessel having the speed of 9.2 knots and the course of 334°. (b) Simulation results with true position plotted with black square markers, observed position plotted with red circle markers, position provided by the KF plotted with start markers and predicted positions with green dot markers. (c) Position errors in x and y axis by using the KF.

5.2. Simulation in self-constructing map

The self-constructing environment is a binary map shown in Figure 12(a) with one ‘island’ in the middle and landmass located at the top and bottom corner. The area has been built to simulate a close range encounter situation with the dimension being 500 m * 500 m. The simulation configurations of the USV and the moving ship (MS) are listed in Table 1. Algorithm’s prediction time period is set as 10, which means that the USV is able to estimate its own the movements as well as the MS for the next 10 time steps. Also, it is assumed the AIS’s transmission interval is every 5 time steps, which makes the USV unable to continuously perceive the MS’s position information thereby requiring position estimation.

Table 1 Simulation configurations for the USV and the moving ship (MS)

	USV	Moving ship (MS)
Start point (m)	(26, 241)	(478,123)
End point (m)	(476,241)	(403,123)
Speed (knots)	10	9
Course	Depends on the path	180 degrees

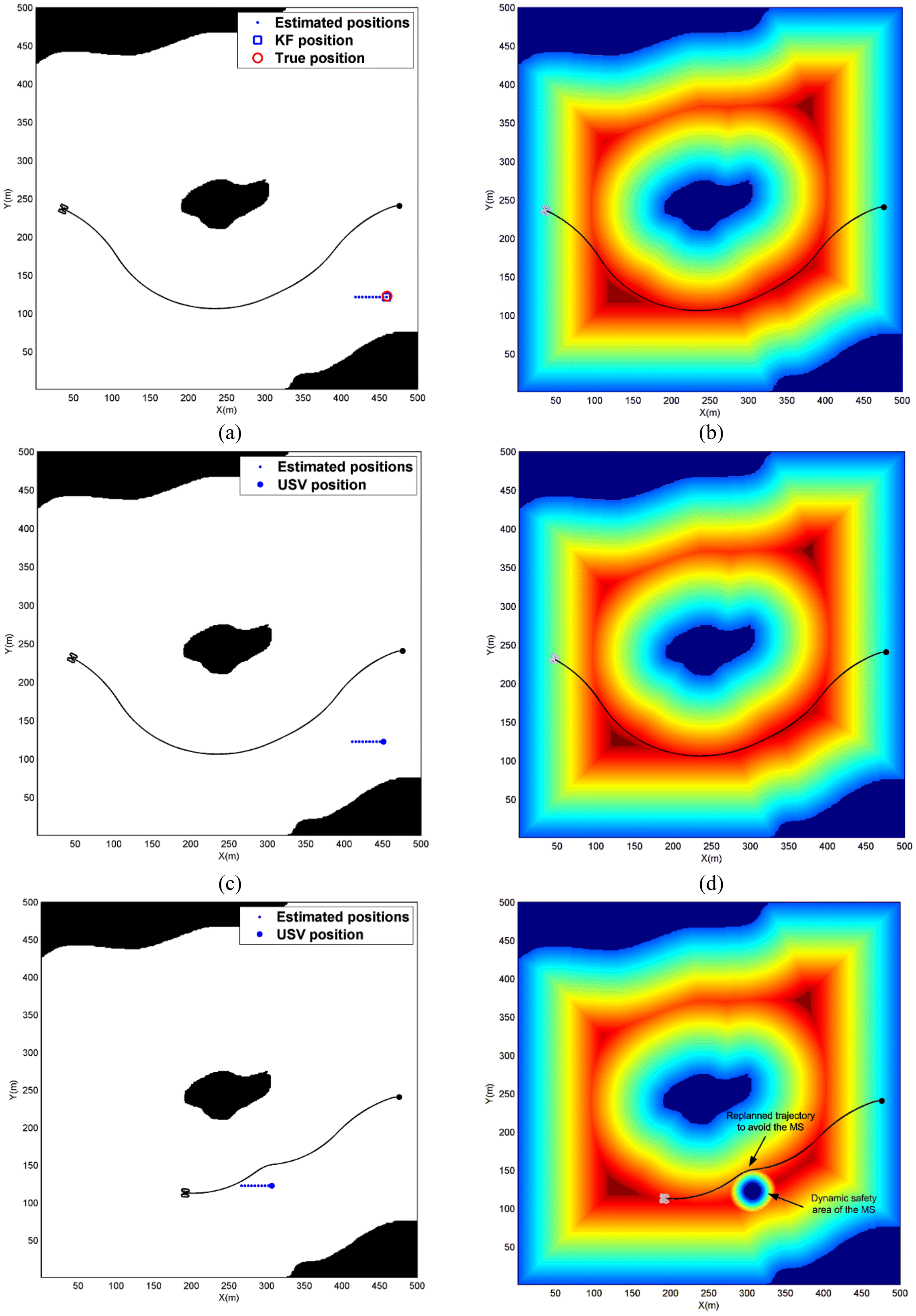
When the USV is launched from the start point, it first reads in the environment map $\boldsymbol{W}_o$ and calculates the safety map $\boldsymbol{W}_s$. Equal weightings ($\alpha = 0.5, \beta = 0.5$) for these two maps are used to generate the synthetic map $\boldsymbol{W}_{ini}$ resulting a ‘balanced’ optimised path. As represented by the black line in Figure 12(a), the initial path generated by the algorithm is able to avoid obstacles and reach the target point. Figure 12(b) illustrates the synthetic map $\boldsymbol{W}_{ini}$, where it is shown that the path tends to stay in the ‘brighter’ area, which is much safer. At initial time step, the instantaneous navigation information is assumed to be available to the USV via AIS. In Figure 12(a), the position produced by the KF is plotted with the blue square marker. It stays close to the red circle marker representing the MS’s true position, which proves that the algorithm provides accurate positional information. Also, possible positions of the MS in next 10 time steps are estimated by the algorithm as small blue markers plotted in Figure 12(a). Based on this, the algorithm determines that the MS currently has no collision risk to the USV, hence the dynamic safety map $\boldsymbol{W}_d$ is not integrated with $\boldsymbol{W}_{ini}$ at this point.

Figure 12(c) and Figure 12(d) show how the algorithm works during the data waiting period when the AIS signal is unavailable. As presented in Figure 12(c), since there is no MS’s position information, the algorithm makes an estimation and therefore only estimated positions are plotted. Collision risk has yet to be presented hence the path is unchanged. Figure 12(d) illustrates the synthetic map.

Figure 12(e) represents the navigation situation at time step 40. The algorithm now determines that if the USV keeps tracking the current path, it will possibly collide with the MS sometime within the next 10 time steps. Hence, the current path needs must change to avoid the collision and in Figure 12(f) a small dynamic safety map $\boldsymbol{W}_{d\_40}$ is generated by the method described in Section 3 and has been integrated with $\boldsymbol{W}_{40}$. The path is now re-calculated by the algorithm to avoid the MS as plotted with black line in Figure 12(e).

Figure 12(g) and Figure 12(h) make a record of how the USV passes by the MS at time step 52. In Figure 12(h), it shows that the USV is able to find the safest path by choosing to pass through the edge of the dynamic safety area, which maintains a satisfactory distance from the

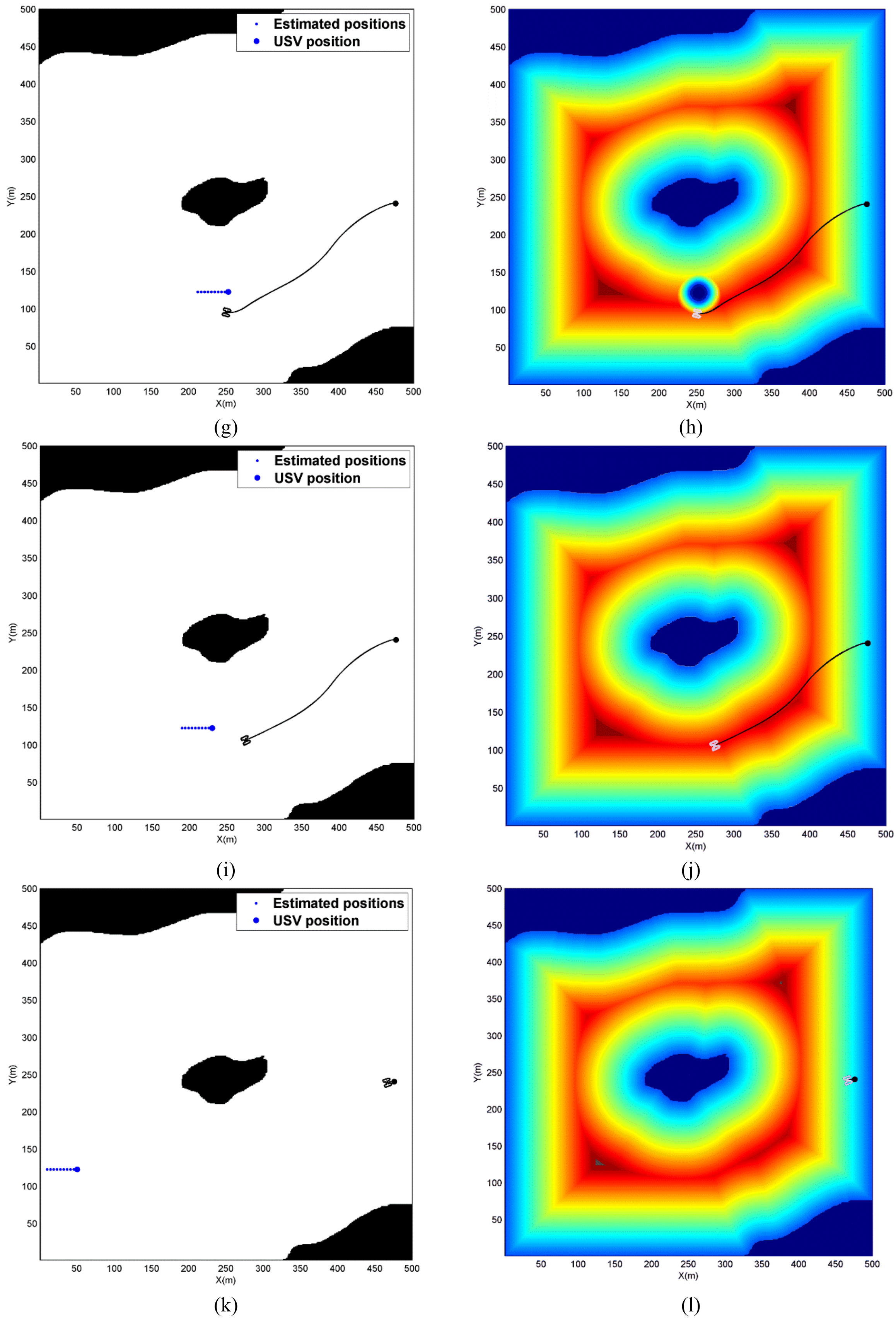
MS. Once the USV clears all collision risks, the path is altered again to seek an optimal path towards the target point with no dynamic safety map adding onto the synthetic map any more (Figure 8(i), Figure 8(j)). The USV can eventually arrive at the target point at time step 97 (See Figure 8(k) and Figure 8(l)).



(a) (b)

(c) (d)

(e) (f)



(g)

(h)

(i)

(j)

(k)

(l)

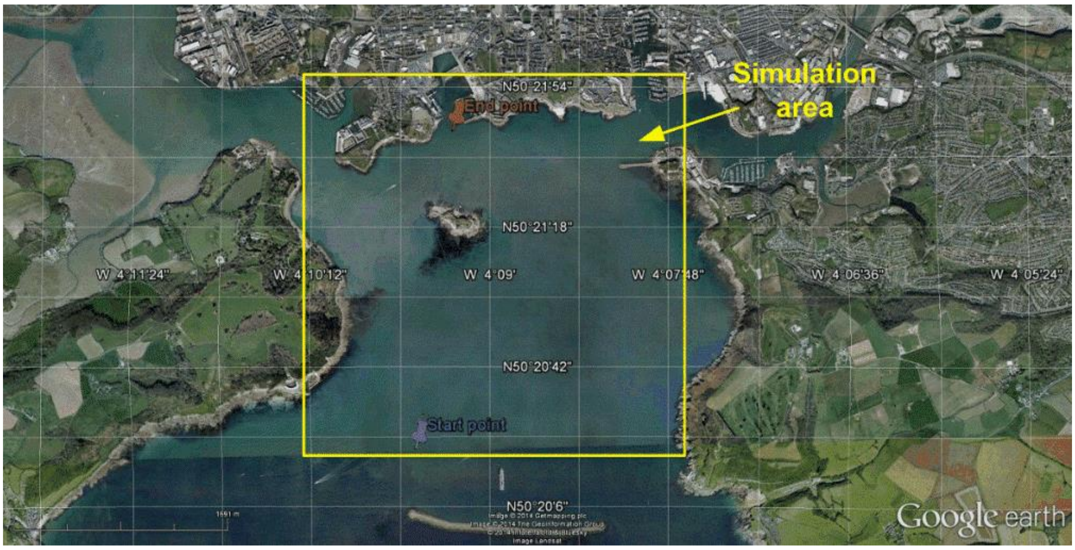
Figure 12. Simulation results in the self-constructing area. Results are represented in both binary maps and according synthetic maps. (a) – (b) Time step = 5. (c) – (d) Time step = 8. (e) – (f) Time step = 40. (g) – (h) Time step = 52. (i) – (j) Time step = 57. (k) – (l) Time step = 97.

5.3. Simulation in practical environment

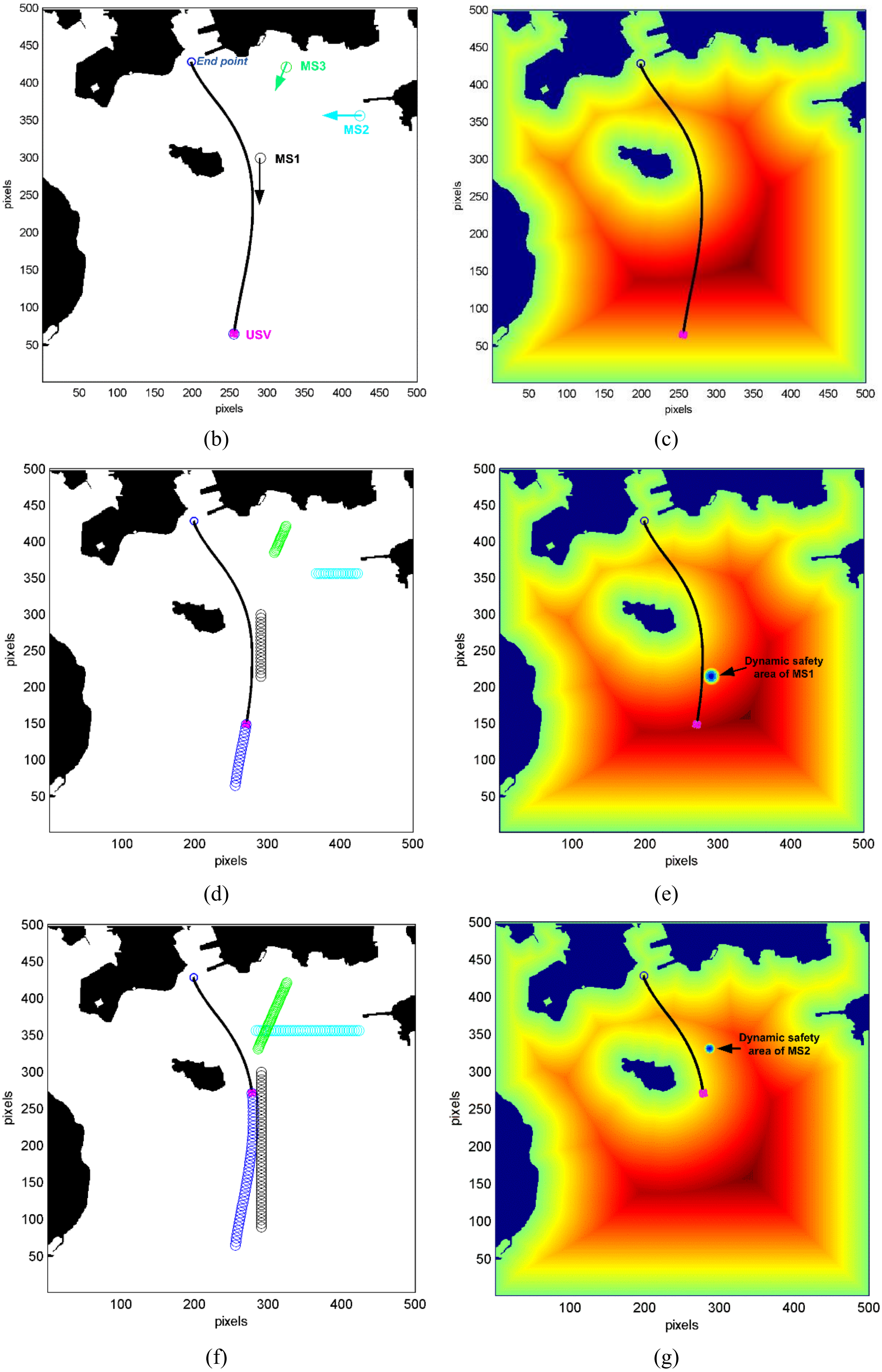
The second test considers a practical maritime environment. The area near Plymouth harbour shown in Figure 13(a) is selected as the testing area, which has 2.5km*2.5km dimension. The selected area is first converted into a binary map shown in Figure 13(b) with 500*500 pixels dimension. To validate the capability of the algorithm dealing with complex traffic situations, three moving ships are added into the environment.

Similar to the first case, an initial guidance path is first generated by the algorithm and shown in Figure 13(b) as the black line. At time step 18 (Figure 13(d)), the collision risk with MS1 is identified by the USV; hence the dynamic safety area of MS1 is created and added into the synthetic map as $\boldsymbol{W}_{18}$(Figure 13(e)).The USV now re-plans its path to avoid the MS1. At time step 43, collision risk with MS1 no longer exists, but there is now a new possible collision risk with MS2. Therefore, only the dynamic safety area of MS2 emerges in the map as $\boldsymbol{W}_{43}$ (shown in Figure 13(g)). As the USV proceeds, the traffic becomes more complicated, and at time step 52, MS3 starts to present a collision threat to the USV while MS2 is still collision risk, which makes the USV need to take actions to avoid both of these two ships. As shown in Figure 13(i), dynamic safety areas for both MS2 and MS3 are integrated with $\boldsymbol{W}_{52}$. Based on $\boldsymbol{W}_{52}$, a collision free path avoiding both static and dynamic threats can be sought, which is shown as a black line in Figure 13(i). Figure 13(j) - Figure 13(m) show how the USV avoids the MS3 and reaches the final target point.

The simulation results show that our algorithm is able to safely navigate the USV in a practical environment. When avoiding the moving ships, the algorithm can 'intelligently' identify the collision risks in real-time and appropriately adjust the path to guarantee safe passage. In terms of the computational time, due to the 'reactive' avoidance scheme mentioned previously, the whole process can be finished in 11.61s. However, it should be noted that such computation time only refers to the simulation time not the actual journey time.



(a)



End point
MS3
MS2
MS1
USV
Dynamic safety area of MS1
Dynamic safety area of MS2
pixels
50
100
150
200
250
300
350
400
450
500

(b)

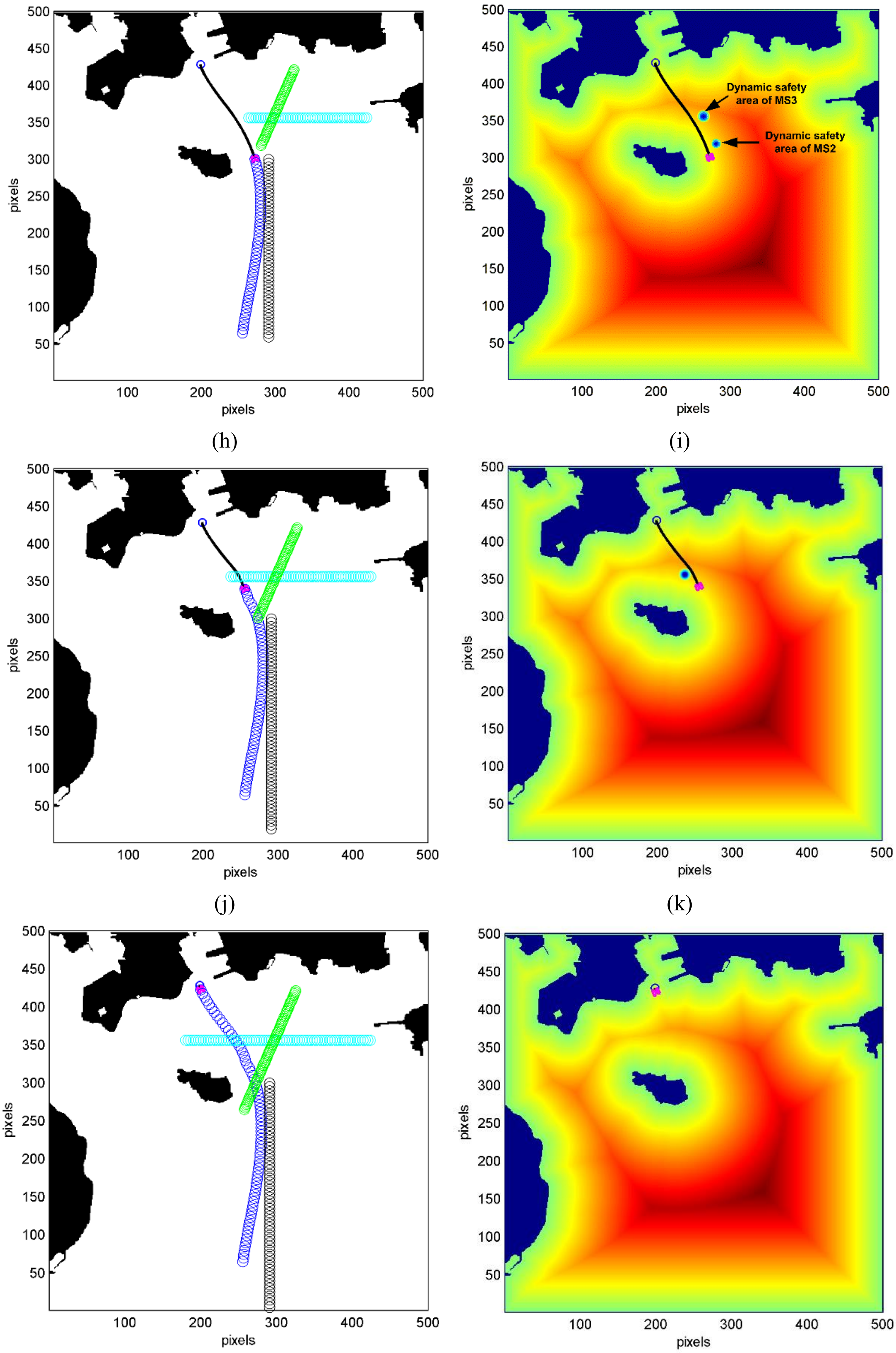
(c)

(d)

(e)

(f)

(g)



500
450
400
350
300
250
200
150
100
50
pixels
100
200
300
400
500
pixels
Dynamic safety area of MS3
Dynamic safety area of MS2

(h)

(i)

(j)

(k)

(l)

(m)

Figure 13. Simulation results in the practical area. (a) The simulation area near Plymouth harbour. (b) - (m) The sequence of movements of USV and its synthetic map at according time step.

6. CONCLUSTIONS AND FUTURE WORK

6.1. Conclusions

This paper represents a new predictive path planning algorithm for USV navigation. To achieve fast computational time and to produce a smooth trajectory, the FMM is used as the base algorithm. A weighted FMS algorithm has been proposed to improve the optimisation performance. By using the algorithm, the generated path can be optimised in terms of both total distance and path safety by adjusting corresponding weightings. Such method makes the path more adaptive in various applications, where the minimum distance has higher priority, and in other cases safety requirement is more important.

Another feature of the algorithm is its capability for dealing with dynamic obstacles. Fully reactive dynamic path planning has been achieved in this paper by using the KF algorithm. Based on the path predicted, the algorithm is able to alert the coming collision risk immediately and generate a new path in shorter time. In order to deal with different dynamics of the moving ship, a variable two-dimensional Gaussian distribution is used to model the dynamic safety area ($\boldsymbol{W}_d$).

The algorithm developed in this paper furthers the development of the research effort for autonomous navigation of USVs. Due to the integration of the KF and the path planning algorithm, the newly developed algorithm improves robustness of the received information and predicts the movement of dynamic obstacles for the case of loss of data or disturbed communications hence overcoming AIS signal blockage and excessive signal noise problems.

6.2. Future work

In terms of future work, several improvements can be done as:

1. **Multiple optimisations**: Currently, only two optimisation disciplines (total distance and path safety) are used. In the next work, more disciplines such as energy consumptions, average speed and ocean current, can be employed to make the path capable of dealing with different requirements. Fortunately, this is not difficult to achieve. Based on the idea in [9], different optimisation requirements can be converted into a sort of 'weighted map' and added above the existing synthetic map $\boldsymbol{W}$, which is the input of the FMM. The possible problem is how to use reasonable values to represent different optimisation requirements.
2. **Algorithm's practicability**: Even though the algorithm has been tested in a practical environment in this paper, the output path can further be improved in terms of the practicability. COLREGS (International collision avoidance rules for marine vehicles) should be used to guide the direction of the path. In the COLREGS, collision avoidance manoeuvres have been specified according to different encounter situations. Since most vessels still adopt rules in COLREGS when actions are needed, it is better for the USV's path planning algorithm implement these rules.
3. **Dynamic obstacles**: In this paper, only CVM has been used to model the movement of moving ship. This is based on the situation that vessels normally keep a constant speed *en route* due to the manoeuvrability issue. However, in reality, more complicated

movement could also be taken by vessels, course change being one of the most common. A coordinated turning model can be used to model such movement. It should be noted that with the change of the movement model, different dynamic models could be used in state transition equation of the KF.

ACKNOWLEDGEMENT

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. The authors are also indebted to Konrad Yearwood for his valuable critique of this paper.

REFERENCES

1. Smierzchalski R. Evolutionary trajectory planning of ships in navigating traffic areas. *Journal of Marine Science and Technology* 1999; **4**: 1–6.
2. Tam C, Bucknall R. Collision risk assessment for ships. *Journal of Marine Science and Technology* 2010a; **15** (3): 257-270. DOI: 10.1007/s00773-010-0089-7.
3. Tam C, Bucknall R. Path planning algorithm for ships in close range encounters. *Journal of Marine Science and Technology* 2010b; **15** (4): 395-407. DOI: 10.1007/s00773-010-0094-x.
4. Xue Y, Clelland D, Lee B.S, Han D. Automatic simulation of ship navigation. *Ocean Engineering* 2011; **38** (17-18): 2290–2305.
5. Naeem W, Irwin GW, Yang A. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* 2012; **22** (6): 669-678. DOI: http://dx.doi.org/10.1016/j.mechatronics.2011.09.012.
6. Kim H, Kim D, Shin JU, Kim H, Myung H. Angular rate-constrained path planning algorithm for unmanned surface vehicles. *Ocean Engineering* 2014; **84**: 37-44. DOI: http://dx.doi.org/10.1016/j.oceaneng.2014.03.034.
7. Garrido S, Moreno L, Blanco D. Exploration of a cluttered environment using Voronoi Transform and Fast Marching. *Robotics and Autonomous Systems* 2008; **56** (12): 1069-1081. DOI: http://dx.doi.org/10.1016/j.robot.2008.02.003.
8. S. Garrido, L. Moreno, D. Blanco and M. L. Munoz. Sensor-based global planning for mobile robot navigation. *Robotica* 2007; **25**: 189-199. DOI:10.1017/S0263574707003384.
9. Gómez JV, Lumbier A, Garrido S, Moreno L. Planning robot formations with fast marching square including uncertainty conditions. *Robotics and Autonomous Systems* 2013; **61** (2): 137-152. DOI: http://dx.doi.org/10.1016/j.robot.2012.10.009.
10. Garrido S, Malfaz M, Blanco D. Application of the fast marching method for outdoor motion planning in robotics. *Robotics and Autonomous Systems* 2013; **61** (2): 106-114, ISSN 0921-8890. DOI: http://dx.doi.org/10.1016/j.robot.2012.10.012.

11. Petres, Clement; Pailhas, Y.; Patron, P.; Petillot, Y.; Evans, J.; Lane, D. Path Planning for Autonomous Underwater Vehicles. *IEEE Transactions on Robotics* 2007; **23** (2): 331-341. doi: 10.1109/TRO.2007.895057
12. Xu B, Stilwell DJ, Kurdila AJ. Fast Path Re-planning Based on Fast Marching and Level Sets. *Journal of Intelligent and Robotic Systems* 2013; **71** (3-4): 303-317.
13. Liu YC, Song R, Liu WW, Bucknall R. Autonomous navigation system for unmanned surface vehicles, in *Proceedings of 13th International Conference on Computer and IT Application in the Maritime Industries*, May 12 -14 2014, Redworth, UK. 123-135.
14. Last P, Bahlke C, Hering-Bertram M, Linsen L. Comprehensive Analysis of Automatic Identification System (AIS) Data in Regard to Vessel Movement Prediction. *Journal of Navigation* 2014; **67** (5): 791-809. DOI: http://dx.doi.org/10.1017/S0373463314000253
15. Sethian JA. A fast marching level set method for monotonically advancing fronts. *Proc Natl Acad Sci USA* 1996; **93** (4): 1591-1595.
16. Liu J, Liu G. Algorithm of Coordinates Conversion in Gauss -Kruger Projection. *Journal of Computer Simulation* 2005; **22** (10): 119-121.