Cooperative motion control of multiple autonomous robotic vehicles
Collision Avoidance in Dynamic Environments

Sergio Carvalhosa*, A. Pedro Aguiar (Advisor)*, A. Pascoal (CoAdvisor)*,  

* Instituto Superior Tecnico - Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal

Abstract: This paper addresses the cooperative motion control (CMC) problem of multiple autonomous robotic vehicles, taking explicitly into account the collision avoidance problem in dynamic environments. The first part of the paper describes the general architecture for CMC, that includes cooperative path following, where multiple vehicles are required to follow pre-specified spatial paths while keeping a desired inter-vehicle formation pattern. In the second part of the paper we propose a Collision Avoidance system (CAS) that is composed by two subsystems: the collision prediction, and the collision avoidance module. For collision prediction we combine a bank of Kalman filters running in parallel, each using a different model for target motion, to derive an unknown object’s velocity and estimate its probable trajectory, the vehicle pre-determined path is then checked for possible interactions with the obstacle. Collision avoidance is achieved either by controlling the speed of the vehicle along its assigned mission path, or through path re-planning using harmonic potential fields. Because group coordination must be taken into consideration, collision avoidance is implemented as part of a Behavior-based system, that is decentralized and can be used with groups of heterogeneous vehicles. The proposed collision avoidance system is then applied to a group of marine surface crafts, where simulation results are presented through the use of a Cooperative Motion Control Simulator developed to model the key different aspects of cooperative multiple vehicle systems.

Keywords: Cooperative Motion Control, Collision Avoidance, Collision Prediction, Autonomous Surface Crafts, Potential field theory, Kalman filter

1. INTRODUCTION

The role of autonomous vehicles and robotics in aiding man in harsh environments has become an increasing focus of interest over the last decade, as the technology that enables this kind of systems becomes available. A particular area that has received special attention, has been the study of coordination and control of various classes of unmanned autonomous vehicles. The main motivation for this trend is the wide range of military and civilian applications where teams of these vehicles working together exhibit better performance in terms of flexibility, robustness and efficiency compared to the single heavily equipped vehicle approach. For tasks such as space exploration, automated transport convoys, security patrols or large object transportation, having a cooperative team of vehicles provides for better area coverage as well has robustness to systems malfunctions. The completion of missions of this nature often requires holding a desired geometrical formation pattern, that is at the same time reactive and adaptive to an unknown environment, as for example to prevent imminent collisions. There are several aspects to bare in mind when designing a cooperative multi-vehicle solution, going from the mechanics of the vehicles to the communication channel between them or even the control strategy followed, all of them represent challenges of their own.

Fig. 1. Collision avoidance typical scenario for a group of autonomous marine vehicles on a data gathering mission at sea.

In an attempt to improve the efficiency and robustness of control methods designed for these types of architectures, a Cooperative Motion Control Simulator was developed and is presented in this paper. Furthermore the development of a Collision Avoidance system is discussed, affording the automated vehicles in the formation the capability to
overcome a given obstacle while taking other formation members into consideration.

In the literature one can find several approaches for collision avoidance in environments with static obstacles, either by applying geometrical methods like the road map and cell decomposition, or based on potential field theory just to name a few. However, in environments with moving obstacles the problem becomes more complex. In terms of moving obstacles it is not enough to know an obstacle position, for efficient collision avoidance it is fundamental to be able to predict the obstacles behavior, in order to lead the vehicle away from Imminent Collisions Situations (ICS).

A typical collision avoidance scenario is shown in Fig. 1, where a convoy of three autonomous surface craft (ASC) performing a path following mission while maintaining a predefined formation, face a number of static and moving obstacles in their trajectory.

This paper presents a solution for this kind of scenario, supported by a behavior-based architecture, where vehicles commute from a mission execution state to a collision avoidance state. A collision prediction module will act as the trigger between states. In a typical situation, during the execution of a mission this architecture will work has follows:

1. Vehicle will detect the obstacle, track it and then derive a model for the obstacle movement. Tracking of the obstacle will continue until its out of sensor range.
2. Within a given time window into the future, the mission path will be checked for possible interactions with the obstacle.
3. In the presence of ICS, a Collision avoidance behavior will then be set according to the type of interaction detected. This can either be velocity correction, or path re-planing based on an inner-map representation of the world.
4. Mission execution behavior will resume once obstacle is labeled as harmless and vehicle is back on mission path.

The outline of this paper is structured as follows. The architecture for a Cooperative Motion Control system and its main characteristics are presented in section 2. In section 3, an overview is given on the Collision Avoidance system, target tracking and collision prediction are addressed. Section 4 describes the path planning method used for collision avoidance as well as the control law for the speed profile of the vehicle. In Section 5, simulation results are presented. Concluding comments are found in section 6.

2. COOPERATIVE MOTION CONTROL ARCHITECTURE

In this section the architecture for a Cooperative multi-vehicle system taken from Vanni F. and M. (2008) and Aguiar and Pascoal (2007) is described. From a theoretical viewpoint, the problems that must be solved to achieve coordination of multiple vehicles cover a vast number of fields, that include navigation, guidance and control. When designing a multi-vehicle solution there are several aspects that should be taken into account. Fig. 2 illustrates the systems functional architecture. It comprises the three main systems around which the problem is formulated:

- the *mission planner*, where parameters and goals are set, and the missions formalized
- the *agent*, comprised of all the subsystems required for a vehicle to be able to follow the mission parameters and coordinate with the rest of the team. These include
  - i) The dynamic model of the vehicle,
  - ii) Mission Controllers and
  - iii) Communications module
- and the *environment*, the vehicles are to be deployed in an environment with its own set of characteristics, that should be taken in consideration and modeled when designing a Cooperative multi-vehicle solution.

2.1 Agents

An elaboration on the architecture described for the agent system is illustrated in Fig. 2.1, where the interaction between the different subsystems is depicted. Let's examine two of these subsystems in more detail:

**Path-following controller** - a dynamical system whose inputs are a path $P_{vi}$, a desired speed profile $v_{ri}$ that is common to all agents, and the agents output $y_i$. Its output is the vehicles input $u_i$, computed so as to make it follow the path at the assigned speed, and a generalized path-variable $\gamma_i$. Further, it accepts corrective speed action from the coordination controller via the signal $\delta_{vi}$. Notice that the dynamics of the parameterizing variable $\gamma_i$ are defined internally at this stage and play the role of an extra design knob to tune the performance of the PF control law.

**Coordination Controller** - the dynamical system that enforces coordination with other team members, receiving has inputs the generalized path-variable $\gamma_{ri}$, and estimates $\gamma_{ri}$ of the generalized coordination states $\gamma_{ij}$ of the $n$ agents it communicates with. It passes on to Path-Following the planned path $P_{vi}$ and its associated speed profile $v_{ri}$.

![Fig. 2. Functional architecture for Coordinated Mission Control](image-url)
Fig. 3. Architecture defined for a vehicle participating in a cooperative mission coupled with the correction speed signal $\tilde{v}_{ri}$ which is used to synchronize agent $i$ with its neighbors.

The proposed Collision Avoidance system (CAS) is to be implemented at this Path-Following motion control level, together with the Coordination algorithm.

3. COLLISION AVOIDANCE OVERVIEW

This section describes the Collision Avoidance module and its integration into the cooperative mission control. In this regard, the scenario in Fig. 1 is once again taken into consideration. In the figure, three unmanned marine vehicles undertake the mission of following an ‘L’ shaped path, while maintaining a triangular formation between them. Static and moving obstacles intersect the vehicles trajectories, and it is therefore necessary to take preemptive measures to avoid collision.

Two main motivations can be used to reflect this problem: i) the main mission task that consists of the necessity to achieve a set of goals, going from offline path planning with coordination control to a target following mission; ii) the self preservation behavior in which collision avoidance plays its role.

As both tasks can at times share some common goals, it is important to define a hierarchical relationship between them to account for those moments when conflicting outcomes from both navigation strategies occur. It is assumed that for the majority of missions, self preservation is the priority, and therefore if during a mission any command for collision avoidance is issued it will overwrite any command for mission execution control. This behavior can be interpreted as the transition between two states, a mission state, and a collision avoidance state.

In Fig. 3 system architecture is represented through a Petri net, in the presence of an obstacle in sensing range, target tracking is launched and kept alive until the obstacle is out of range. Target tracking provides for a model from which to derive the probable trajectory of the obstacle, Collision Prediction can then determine if collision is imminent, and trigger the transaction to Collision Avoidance state. Collision avoidance module then takes the necessary measures, that will be presented further ahead, and ensure collision is avoided. Once the maneuver is completed, and if no more imminent collisions are present, the mission execution state can resume. In this manner, CA module can alter both the path and the velocity profile planned for the mission, and feed it to the path following algorithms specific to that type of vehicle.

3.1 Prediction Module

The ability to successfully avoid collision with an obstacle, is closely related with how much in advance one can be aware of it, even more so when the obstacles are moving. Being able to predict a collision ahead of time, will not only give the vehicle time to react to dynamic obstacles, but the ability to do so in a robust and smooth manner.

**Target Tracking**

Target tracking commences once an obstacle is in sensor range, it is assumed that its position can be determined at every time step with a zero mean Gaussian error. For the time window in which the trajectory is to be predicted, obstacles are assumed to have bounded linear $v$ and angular $w$ velocities, and their behavior can be described by the following kinematic model:

\begin{align*}
\dot{x} &= v \cos(\theta) \\
\dot{y} &= v \sin(\theta) \\
\dot{\theta} &= w
\end{align*}

(1)
To estimate both \( v \) and \( w \) and derive a targets future position, an Interactive Multiple Model Kalman filter (IMM-KF) is used. The IMM-KF is a nonlinear filter that combines a bank of Kalman filters running in parallel, each one using a different model for target motion, with a dynamic system that computes the conditional probability of each KF. The output of the IMM-KF is the state estimate given by a weighted sum of the state estimations produced by each Kalman filter. Further details on the IMM-KF can be found in the book by Bar-Shalom Y. and X.R. (2002). For the implementation of the IMM-KF in the collision prediction module the solution developed in M. Bayat and Aguiar (2009) is applied. In it each Kalman filter \( j \) is designed according to the following discrete process model with a constant sampling time \( T_s \):

\[
\begin{align*}
    x^j_{k+1} &= x^j_k + T_s V^j_k \cos \theta^j_k \\
    y^j_{k+1} &= y^j_k + T_s V^j_k \sin \theta^j_k \\
    \theta^j_{k+1} &= \theta^j_k + T_s w^j + w^j_{bk} \sqrt{T_s} \\
    V^j_{k+1} &= V^j_k + w^j_{vk} \sqrt{T_s}
\end{align*}
\]

(2)

where \( w^j \) is the angular velocity and is set to be constant with a different value in each model, ranging from \(-w_{\text{max}}\) to \(w_{\text{max}}\) including the origin(straight line translation).

Collision Prediction

Once the variables for the obstacle model \( \{X, v, w\} \) have been derived trough target tracking, it is now possible to make a reasonable estimate of its position in a given time window into the future. Knowing that our automated vehicle is performing a predefined mission path, we can test if the local configurations and velocities of the vehicle and the moving obstacle may lead to a collision.

Its important to bare in mind that although a wider time window can result in greater anticipation to collisions, it will also add to uncertainty on the obstacle estimated position. For each prediction a sample Time Varying Dynamic Window (TVDW) can be used, taking in consideration factors like cruise velocity, channel uncertainty and system kinematic and dynamic constraints. The model adopted for the TVDW used in simulations takes into account the time needed to reach a complete halt when traveling at a certain speed:

\[
W^{\tau_0} = [t_0, t_0 + \delta], \quad \delta = \frac{v_0}{a} + \zeta,
\]

(3)

where \( v_0 \) is the velocity of the vehicle at time instant \( t_0 \), \( a \) is the maximal breakage deceleration of the vehicle under horizontal translation, and \( \zeta \) is a desired safe margin.

Trajectory estimation for the vehicle and a moving obstacle is shown in Fig. 5, where two different scenarios are addressed. Let \( A^{t_{vh}}_{} \) and \( A^{t_{ob}}_{} \) denote the current region occupied by the vehicle and obstacle respectively, change in position and orientation of these regions in 2D space can be represented by:

\[ A_{vh} = \Phi_v(x_v(t), y_v(t), A^{t_{vh}}_{}), \quad \forall t \in W^{t'} \]

\[ A_{ob} = \Phi_o(x_o(t), y_o(t), A^{t_{ob}}_{}), \quad \forall t \in W^{t'} \]

here \( \Phi \) denotes the operator that takes as arguments the position \( x(t), y(t) \) of the trajectory at time instant \( t \) and a given region \( A^t \), and provides the region \( A^t \) at time \( t \). We can then define collision has the situation where both regions \( A^{t_{vh}}_{vh} \) and \( A^{t_{ob}}_{ob} \), share points in common,

\[ A_{vh} \cap A_{ob} \neq \emptyset \]

(4)

For collision prediction we are interested in a spatial and temporal correlation analysis of these regions. Even thought knowing the intersection of the swept areas of \( A_{vh} \) and \( A_{ob} \) can be useful data for collision avoidance, given that \( \Phi_v \) and \( \Phi_o \) are functions of time, it will only be relevant if some points in the intersection are time correlated.

Thus temporal correlation is defined has the first step to predict a possible collision. For this we take the trajectory’s for the center of mass derived for both the vehicle and the obstacle, \( P_v(t) \) and \( P_o(t) \), and compute the local minima of their euclidean distance in function of time, in the time window \( W^{t'} \), which we will denote by the pairs \( \{t^*_i, d^i(t^*_i)\} \). We then define

\[ t_c = \min t^*_i \]

(5)

as the time instant for the first minimum in which a collision might occur. Assuming that the regions occupied by the intervenients are inside two tight circles of radius \( R_{vh} \) and \( R_{ob} \), and let \( \epsilon = R_{vh} + R_{ob} \), confirming the inequality

\[ d(t_c) = \|P_v(t_c) - P_o(t_c)\| \leq \epsilon \]

(6)
Fig. 6. Collision scenario in which the obstacle behavior is interpreted has a static virtual object (SVO) has true, will verify collision. In the case where condition (6) is not verified, no more calculations are needed, and so Collision Prediction module is done for the current time step.

Once collision is verified, calculations to find the intersection of the swept areas of both intervenients can then be narrowed to a window expanding from $t_c$. With this knowledge, we can extrapolate more data relating the obstacle to later be used in an efficient Collision Avoidance maneuver. Consider $B_{vh}$ as the swept area of the vehicle, let $t_i$ and $t_f$ be the initial and final time instants for which $A_{ob}$ overlaps $B_{vh}$ as illustrated in Fig.6. The obstacle will be labeled dynamic or a static virtual (SVO) object depending on $\Delta t = t_f - t_i$, the time it interferes with the mission path. When labeled as a SVO, a representation of the obstacle will be passed on to the Collision Avoidance inner map based on the obstacle swept area. Collision Avoidance state is enabled.

4. COLLISION AVOIDANCE

Up to this stage no actual collision avoidance maneuvers have yet been issued, Collision Prediction Module acts has a trigger and has a data gatherer for the Collision Avoidance state. The main bulk of the work done towards the development of actual Collision avoidance is now presented.

In a dynamic environment there are two actions a vehicle can take in the effort of avoiding a collision, one is to temporally deconflict the trajectory’s which relates to acting on the velocity of the vehicle throughout the mission path, and the other is to attempt a spacial deconfliction of the paths involving the planing of an alternative path. The solution adopted takes from both approaches. A threshold $\Delta t_{\text{threshold}}$ is set, that defines the maximum time period an obstacle can overlap the vehicles path, whenever a long interference with the path is detected the safest approach is taken and an alternative path is computed.

4.1 Path planning using Harmonic Potential fields

The Potential field method is used for the online planning of the new path, in this method an artificial potential field is assigned to the area where a robot works, obstacles in the area are assigned repulsive potentials while an attractive potential describes the goal position. The new path can be derived from the gradient of the total artificial potential.

It is common for potential field methods to suffer from local minima. A local minimum can attract and trap the robot, preventing it from reaching its final goal. For this reason Harmonic potential fields are utilized by employing the panel method known in fluid mechanics, harmonic function completely eliminates local minima, which eliminates the possibility of generating a stationary point in the velocity field except the goal point. Therefore all the potential functions used in the creation of the artificial field obey Laplace equation $\nabla^2 \phi = 0$, and can generate either attractive or repulsive potentials.

\[
\phi_g = \frac{\lambda_g}{2\pi} \ln(\sqrt{(x-x_g)^2 + (y-y_g)^2}) \quad (7)
\]
\[
\phi_u = -U(x \cos \alpha + y \sin \alpha) \quad (8)
\]

Equations (7) and (8) refer to the goal and uniform flow potential respectively, where $(x_g, y_g)$ are the goal coordinates, $\alpha$ is the angle between the $x - \text{axis}$ and the direction of the uniform flow, and $\lambda_g$ and $U$ are the potentials strengths. Both potentials will drive the vehicle to its desired goal position.

Panel method

The panel method that has been used to solve the potential flow of a fluid around bodies of arbitrary shape, is in this case used to realize obstacle avoidance. The boundary of the obstacle in 2D space will be approximated by line segments(panels), each of which is distributed with source or sink singularities having uniform density. The
distributed singularities are used to deflect the oncoming stream so that it will flow around the body. The velocity potential at any point \((x, y)\) in space caused by a panel \(j\) is

\[
\phi_{p} = \frac{\lambda_{j}}{4\pi} \int_{j} \ln(R_{j}) \, dl_{j}
\]

where \(R_{j} = (x - x_{j})^{2} + (y - y_{j})^{2}\), and \(\lambda_{j}\) is the strength of the source field per unit length. The overall potential created by an obstacle at a given point is the result of the net effect of all the panels that compose the obstacle frame, \(\phi_{ob} = \sum_{j=1}^{m} \phi_{j}\). The total artificial potential field is then given by

\[
\phi_{\text{total}} = \phi_{g} + \phi_{u} + \phi_{ob}
\]

\[
= -U(x \cos \alpha + y \sin \alpha) + \frac{\lambda_{g}}{2\pi} \ln(\sqrt{R_{g}})
\]

\[
+ \sum_{j=1}^{m} \frac{\lambda_{j}}{4\pi} \int_{j} \ln(R_{j}) \, dl_{j}
\]

A typical environment is illustrated in Fig. 7 where all the potentials are represented, let \(V_{i} > 0\) be the desired outward normal velocities at the center points of the panels

\[
\frac{\partial}{\partial n_{i}} \phi(x_{i}, y_{i}) = V_{i}, \quad i = 1, 2, ..., m
\]

where denotes \(n\) a vector normal to the panel. Setting \(\lambda_{g}\) and \(U\), \(\lambda_{j}\) are then chosen so that inequality (11) is verified. The equations for \(\lambda_{i}\)'s are derived in Kim Jin-Ho (1992) for a set of desired \(V_{i}\). The values set for \(V_{i}\) do however need to be limited so that convergence to the threshold is verified. The equations for \(\lambda_{i}\)'s are then chosen so that inequality (11)

\[
-\lambda_{g} > \lambda_{ob} > 0
\]

where the obstacle strength \(\lambda_{ob}\) was defined as

\[
\lambda_{ob} = \sum_{i=1}^{m} \lambda_{i} L_{i}
\]

The solution adopted in this paper takes form Fahimi (2009) a way to automatically compute a set \(m\) of \(V_{i}\) that satisfy inequality (12). The corresponding velocity field, \(\mathbf{v} = (\dot{x}, \dot{y})\), is then derived from \(\mathbf{v} = -\nabla \phi\) and using eq. (10).

\[
v_{x}(x, y) = U \cos \alpha - \frac{\lambda_{g}}{2\pi} \frac{\partial}{\partial x} \ln R_{g} - \sum_{j=1}^{m} \frac{\partial}{\partial x} \ln R_{j} \, dl_{j}
\]

\[
v_{y}(x, y) = U \sin \alpha - \frac{\lambda_{g}}{2\pi} \frac{\partial}{\partial y} \ln R_{g} - \sum_{j=1}^{m} \frac{\partial}{\partial y} \ln R_{j} \, dl_{j}
\]

This resultant velocity vector \(\mathbf{v} = (\dot{x}, \dot{y})\) is characterized by size and orientation, where the size is given by \(V = (V^{T} V)^{\frac{1}{2}}\) and orientation obtained from \(\beta = \arctan(\frac{\dot{y}}{\dot{x}})\). Following \(v\) throughout the potential field will result in an obstacle free path.

\subsection{4.2 Velocity correction}

Path re-planning which implies steering away from an obstacles trajectory will only occur when prediction determines that both paths overlap for a period of time \(\Delta t > \Delta t_{\text{threshold}}\), and is therefore safer to search for an alternate path. Such will be the scenario in the case of a static obstacle being detected, or a ship traveling in the opposite direction.

Velocity correction is applied when the detected trajectory intersections have a time span lower than the defined threshold \(\Delta t_{\text{threshold}}\), and so obstacles are avoidable by either increasing or decreasing velocity. Let’s refer to the data acquired in the collision prediction step depicted in Fig. 6 where overlapping of the vehicle path will occur for all points defined in

\[
(x, y) \equiv P_{v}(\gamma(t)) \in [P_{A}, P_{B}]
\]

during the time interval \([t_{i}, t_{f}]\), where

\[
P_{A} = P_{v}(\gamma(t_{i})
\]

\[
P_{B} = P_{v}(\gamma(t_{f})
\]

Problem: Consider the collision region given in (16), and let \(p_{d}(\gamma) \in \mathbb{R}^{2}\) be a desired path parameterized by a continuous variable \(\gamma(t) \in \mathbb{R}\) and \(v_{d}(\gamma) \in \mathbb{R}\) a desired speed assignment. Suppose also that \(p_{d}(\gamma)\) is sufficiently smooth and its derivatives with respect to \(\gamma\) are bounded. Derive a control law such that

\[
P_{v}(\gamma(t)) \notin [P_{A}, P_{B}] \Leftrightarrow \gamma(t) \notin [\gamma_{A}, \gamma_{B}], \forall t \in [t_{i}, t_{f}],
\]

which means the vehicle remains outside the collision area for the given time interval \([t_{i}, t_{f}]\). It is easy to see that arriving at \(P_{B}\) at time \(t_{i}\) or at \(P_{A}\) at time instant \(t_{f}\), verifies condition (17) and assures that the vehicle avoids collision with the moving obstacle.

To derive the velocity control law for the vehicle, we imagine a virtual target traveling along the path at a constant velocity \(V_{q}\), such that at an intended time instant \(t_{g}\), the target will be located at a goal position \(\gamma_{g}\) along the path. The virtual target position and velocity along the parameterized path are then given by

\[
V_{d} = \gamma_{g} - \gamma_{0} \quad t_{g} - t_{0} = \text{const}
\]

\[
\gamma_{d}(t) = \int_{t_{0}}^{t} V_{d} \, dt = V_{d}(t - t_{0}) + \gamma_{0}
\]

The objective is now to make the vehicle track this virtual target. Defining the position displacement between the target and the vehicle

\[
\Delta x = x_{g} - x_{0}
\]
\[ e(t) = \gamma(t) - \gamma_d(t) \]  
and the velocity error
\[ \dot{e}(t) = v(t) - v_d \]
one can easily derive a feedback law for \( V \) such that \( \dot{e} = -ke \), that is the error converges to zero exponentially fast. The control law for velocity is then given by
\[ v = V_d - k(\gamma - \gamma_d) \]
\[ = \frac{\gamma_g - \gamma_0}{t_g - t_0} - k(\gamma - \gamma_d) \]
Choosing the goal \((\gamma_g, t_g) \rightarrow (\gamma_A, t_f)\) or \((\gamma_g, t_g) \rightarrow (\gamma_B, t_i)\), will make the controller drive the vehicle to decrease or increase velocity respectively, realizing collision avoidance.

**Right of Way**

The primary drive for the vehicle is to give right of way to an obstacle, that means that once an obstacle is detected and is labeled has dynamic \((\Delta t < \Delta t_{\text{threshold}})\) the vehicle will take the safest approach, which is to decrease velocity and let the obstacle pass, resuming the velocity profile set for the mission once collision avoidance has been assured. The only exception to this behavior will occur when in the presence of another automated vehicle, possibly a team member, in this scenario priority will be issued to the vehicle traveling on starboard tack. With this behavior decentralized collision avoidance is realized, where the vehicles have no need to exchange data in order to achieve coordination.

5. SIMULATION RESULTS

The algorithms of the previous sections were tested through Matlab simulations, evaluation of the performance of the individual modules was first realized. The overall system was then tested in a Cooperative Mission Control (CMC) architecture by resorting to a a newly developed CMC simulator. Fully supported on Simulink and Matlab, the simulator models the key different aspects of cooperative multiple vehicle systems. All simulations relate to the application of the Collision Avoidance system to Autonomous Surface Crafts (ASC), for which the Path-following controllers developed in Maurya P. (2008) were used.

Fig. 8 shows collision prediction between the ASC and an incoming dynamic obstacle, the targets position is fed directly to the Interactive multi model Kalman filter (IMMKF) once every second with a zero mean Gaussian error. The estimates for the targets \( w = 0.0312 \) and \( v = 1.5723 \) returned by the IMM-KF are used to derive a probable trajectory for the obstacle. An intersection of paths is detected for a time span larger than the defined \( \Delta t_{\text{threshold}} = 15s \), resulting in the creation of a Static Virtual Obstacle that can later be used to compute an alternate path.

**Fig. 9. Path planing for a team of ASCs**

In Fig. 9 the results for harmonic potential field path planing are shown for a team of vehicles. Different values were set for the potential forces at the center of the panels \( V_i \) to stress out the situation where for high enough values of \( V_i \) the vehicle might miss the goal. To avoid this situation in both Fig.11 and Fig.10, path planing
is realized taking into account the bound values of \( V_i \), ensuring convergence to the goal position.

Fig. 10. Path Planing in a cluttered environment, gradient of the potential field

Fig. 11. Robust harmonic potential field - Path planing taking into account \( V_i \) bounds

Fig. 12. Velocity correction output in bottleneck situation, issued to achieve time decoupling of vehicle paths

Fig. 13 shows the implementation of the Collision Avoidance System in a team of three ASC performing a path-following mission. In this scenario, two of the team members perform path re-planing to avoid collision with the obstacles and reach a bottleneck situation, velocity control then comes into play (Fig. 12) to prevent inter-vehicle collisions. The team is thus able to move past the bottleneck in a coordinated manner.

6. CONCLUSIONS

The paper proposed a collision avoidance system to be applied to autonomous vehicles working in dynamic environments. To integrate collision avoidance in a typical cooperative motion control (CMC) architecture, an hierarchical structure was devised. This approach allows for a
team of autonomous vehicles to have the ability to stray away from mission parameters to prevent collisions. The problem was decoupled into a collision prediction stage and a corresponding collision avoidance maneuver. Prediction is realized by first estimating the targets velocity through the use of an Interactive multi model Kalman filter, and then deriving its probable trajectory within a given time window. Two strategies were then devised to avoid collision: Path re-planing based on harmonic potential field theory, or by controlling the speed of the vehicle along its assigned mission path. The efficiency of the solutions developed was shown with selected simulation examples. There are several important issues that need to be addressed in future research in this area, including: 1) Optimization of the devised trajectories to take into account energy consumption or other vehicle or mission constraints 2) Cooperative navigation and mapping, where the formation members share resources to reach a more accurate representation of the environment map.

REFERENCES


