Complete decentralised Navigation Functions for agents with finite sensing regions with application in aircraft conflict resolution

Giannis Roussos and Kostas J. Kyriakopoulos

Abstract—We develop the framework of decentralised Navigation Functions for the case of multiple agents of arbitrary shapes and sensing areas around them moving in \( N \)-dimensional space. Unlike previous approaches utilising the Navigation Functions methodology, the construction here does not rely on diffeomorphisms, thus reducing the computational cost of the algorithm. The resulting potential field is absolutely locally computable and can be easily adapted to decentralised aircraft conflict resolution, as shown in the paper. Simulation results of multi-aircraft conflict resolution are included to support the efficacy of the complete control scheme.

I. INTRODUCTION

The problem of navigation and motion planning for single, as well as for multiple agents is of great interest to a wide class of applications, ranging from the Control domain to Robotics and Air Traffic Management (ATM). Conflict avoidance and convergence to the target are the key requirements in such problems, while communication and performance constraints may apply.

Navigation Functions (NFs) have been proposed in [1] as an integrated solution to the path and motion planning problems, initially for the case of a single robot inside a bounded environment with stationary obstacles. The foundation of the NFs methodology emerged as an artificial potential field approach [2], where collision avoidance is achieved via repulsion between the robot and the obstacles, while an attractive component towards the destination is achieved via attraction. This general and flexible form in which agent and workspace are defined allows the use of onboard sensors with finite range and scalability of the algorithm in large scenarios. Limited sensing so far has been introduced in a number of ways in NFs. In [7] the authors use a \( C^0 \) sensing scheme, but assume a priori knowledge of the total number of agents. This requirement is removed in [8], where a switching sensing graph is used, resulting in a hybrid system. This approach does not ensure global convergence, as blocking situations may be reached. Thus, convergence occurs only if the switching of the sensing graph eventually stops. A completely locally computable NF has been presented in [9], but only for single-agent problems and with the assumption that at each time instant there is at most one visible obstacle. This effectively means that the algorithm solves the collision with one obstacle at a time, which is not practical in a multi-agent scenario. An initial approach towards the application of local sensing in a continuous manner for multiple obstacles has been presented in [10] for a circular sensing area, while a non-circular sensing scheme has been introduced in [11] that improves the quality of the resulting paths.

This work presents a generalisation of the NF construction in scenarios with non-circular agents without the need for diffeomorphisms, while local sensing is also implemented via non-circular sensing zones. The construction of the potential relies on \( C^2 \) functions that implicitly define the boundaries of the agents and their sensing areas in the \( N \)-dimensional space. This general and flexible form in which agent and sensing area shapes are described allows the tailoring of the methodology to the specific characteristics of aircraft conflict
resolution. The significant difference between horizontal and vertical separation minima in Air Traffic Control (ATC) can be easily taken into account by the algorithm. Moreover, a specially chosen non-circular sensing area can offer improved trajectories, as already demonstrated for the 2D case in [11].

The rest of the paper is organised as follows: in section II we review the basic structure of decentralised Navigation Functions (NFs) that we employ, along with its components. In the next section the proposed construction for the collision function is presented for the general case of non-circular shapes. In section IV the adaptation of the algorithm to the problem of aircraft conflict resolution in 3D space is developed. Simulation results are presented in section V, followed by our conclusions in section VI.

II. COMPLETELY DECENTRALISED NAVIGATION FUNCTIONS

The NF construction we present here is intended to cover a general class of problems where a group of $N$ agents of various shapes and sensing characteristics operate in a common environment. Specifically, our work here focuses on the construction of the collision function $\beta_{ij}$, which is a key component of the complete potential. The proposed approach inherently takes into account the shape of each agent $i$, $O_i$, and the shape of the sensing region $A_i$, within which it can detect other agents and obstacles, as in Fig. 1.

![Fig. 1: Agent $i$ located at $q_i$. Its shape is represented by $O_i$ and its sensing area by $A_i$](image)

The basic form of the decentralised NF $\Phi_i$ for agent $i$ is based on previous multi-agent NF approaches [6]:

$$\Phi_i = \frac{\gamma_{di} + f_i}{((\gamma_{di} + f_i)^k + G_i \cdot \beta_{0i})^{1/k}},$$

where $G_i$ quantifies the proximity to collisions involving agent $i$: it is zero when the $i$th agent participates in a conflict, and positive otherwise. $\gamma_{di}$ is the goal function, fading at the destination of agent $i$, $q_{id}$, and increasing away from it. Function $f_i = f_i(G_i)$ enables cooperation between neighboring agents as explained in [6], while $\beta_{0i}$ is the workspace bounding obstacle that limits the motion of agents inside the available workspace.

Our contribution in this paper, as described in the next section, lies mostly in the construction of the individual collision functions $\beta_{ij}$ that model the effect of other agents and obstacles to agent $i$. The product of all $\beta_{ij}$ forms the overall collision function $G_i$ that is used in the potential:

$$G_i = \prod_{j \in T_i} \beta_{ij}$$

where $T_i = \{1, \ldots, N\} \setminus i$.

For the target function $\gamma_i$ we use the normalised form introduced in [10]:

$$\gamma_i = \frac{||q_i - q_{id}||^2}{R_n^2}$$

where $R_n$ is a reference length used for the normalisation that reflects the physical scale of the problem, e.g. the largest dimension of the workspace.

The cooperation function $f_i$ is used here as in [7]:

$$f_i(G_i) = \begin{cases} a_0 + \sum_{i=1}^{3} a_i G_i^i, & G_i \leq X \\ 0, & G_i > X \end{cases}$$

where $a_0 = Y$, $a_1 = 0$, $a_2 = -\frac{3Y}{2}$, $a_3 = 2Y$ and $X$, $Y$ are positive parameters. Values of $G_i$ lower than $X$ activate the cooperation function $f_i$, while $Y$ defines the maximum value of $f_i$, which is attained when $G_i = 0$.

III. CONSTRUCTING THE COLLISION FUNCTION

The repulsive function $\beta_{ij}$ introduced here is redesigned to handle a more general class of shapes for the obstacles as well as the agents and their sensing regions. Instead of building the NF potential on a virtual sphere world, we adapt the methodology to real shapes that can be implicitly defined through the level sets of some functions.

A. Implicit obstacle description

In order to derive the contribution of an intruding agent $j$ to agent $i$’s potential $\Phi_i$, $\beta_{ij}$, we use an implicit obstacle function $\delta_{ij}$ as in [12], such that the boundary of collision between agents $i$ and $j$ corresponds to its zero level set:

$$O_{ij} \triangleq \{(q_i, q_j) | \delta_{ij} \leq 0\}$$

$$\partial O_{ij} = \{(q_i, q_j) | \delta_{ij} = 0\}$$

where $O_{ij}$ is the set that represents all possible collisions between agents $i$ and $j$. The obstacle function $\delta_{ij}$ is a measure of collision between agents $i$ and $j$: it is negative when the agents are in collision, fades to 0 when they touch and becomes positive when they are separated.

In applications where the collision avoidance requirements are expressed directly in terms of the relative position $q_{ij} = q_j - q_i$ between agents, the collision set $O_{ij}$ can be described via a relative obstacle function $\delta_{ij}(q_{ij})$, such that $\delta_{ij}(q_{ij}) \leq 0$ when agents $i$ and $j$ collide, i.e. $(0, q_{ij}) \in O_{ij}$. The
implicit obstacle function $\delta_{ij}(q_j)$ of the absolute position $q_j$ can be then derived directly as:

$$\delta_{ij}(q_i, q_j) = \tilde{\delta}_{ij}(q_j) + \langle \tilde{\delta}_i(q_i), q_j \rangle$$

Alternatively, the shape $\tilde{O}_i$ of each individual agent $i$ may be given as the zero level set of a relative shape function $\tilde{s}_i(q_i)$:

$$\tilde{O}_i \triangleq \{ q_i : \tilde{s}_i(q_i) = 0 \}$$

(7)

$$\partial \tilde{O}_i = \{ q_i : \tilde{s}_i(q_i) = 0 \}$$

(8)

where $q^i = q - q_i$ is the position in the local frame of agent $i$, originated at $q_i$. For an arbitrary position of agent $i$ the absolute shape function $\delta_i(q_i, q)$ defines the shape $O_i$ of agent $i$ in the global frame, via a translation of $O_i$ by $q_i$:

$$\delta_i(q_i, q) = \tilde{\delta}_i(q - q_i)$$

$$O_i(q_i) \triangleq \{ q : \delta_i(q_i, q) \leq 0 \}$$

$$\partial O_i(q_i) = \{ q^i : \delta_i(q_i, q^i) = 0 \}$$

(9)

The collision set $O_{ij}$ between two agents $i$ and $j$ will comprise all $(q_i, q_j)$ pairs that cause $O_i(q_i)$ and $O_j(q_j)$ to overlap. The intersection of $O_i(q_i)$ and $O_j(q_j)$ can be implicitly represented using the formula given in [13]:

$$\psi_i(\delta_i, \delta_j) = \delta_i + \delta_j + \sqrt{\delta_i^2 + \delta_j^2}$$

$$O_i(q_i) \cap O_j(q_j) = \{ q | \psi_i(\delta_i(q_i, q), \delta_j(q_j, q)) \leq 0 \}$$

$$\partial (O_i(q_i) \cap O_j(q_j)) = \{ q | \psi_i(\delta_i(q_i, q), \delta_j(q_j, q)) = 0 \}$$

(10)

Therefore, the obstacle set $O_{ij}$ will then comprise all $(q_i, q_j)$ pairs that result in an non empty $O_i(q_i) \cup O_j(q_j)$, i.e. $(q_i, q_j) \in O_{ij} \iff O_i \cap O_j \neq \emptyset$:

$$O_{ij} = \{ (q_i, q_j) | \min_q \psi_i(\delta_i(q_i, q), \delta_j(q_j, q)) \leq 0 \}$$

The boundary $\partial O_{ij}$ will comprise all $(q_i, q_j)$ pairs that cause $O_i(q_i)$ and $O_j(q_j)$ to touch, i.e. all their common points will be on $\partial O_i(q_i) \cup \partial O_j(q_j)$:

$$\partial O_{ij} = \{ (q_i, q_j) | \min_q \psi_i(\delta_i(q_i, q), \delta_j(q_j, q)) = 0 \}$$

(11)

For all $(q_i, q_j)$ pairs such that $O_i(q_i)$ and $O_j(q_j)$ overlap there is at least one common point $q_c$ that belongs to both of them, thus $\delta_i(q_i, q_c) \leq 0$ and $\delta_j(q_j, q_c) \leq 0$. Rewriting this with $q_c$ as a reference point and using $q'_j = q_j - q_c$ as the relative position vector of $q_j$ with respect to $q_c$ we obtain:

$$\delta_j(q_j, q_c) = \tilde{\delta}_j(-\langle q_j, q_c \rangle) = \tilde{\delta}(-q'^j_c) \leq 0$$

$$\delta_i(q_i, q_c) = \tilde{\delta}_i(q_i - q_c) = \tilde{\delta}(-q'^i_c) \leq 0$$

(12)

which means that $q_j$ belongs to the set $O'_j(q_c)$, produced by mirroring $O_j$ with respect to the hyperplane that is normal to the $N$-dimensional vector of ones $1 = \[ 1 \ 1 \ \ldots \ 1 \]$, and then translating it to $q_c$, as shown in Figure 2. Thus, the set of all $q_j$ such that $O_i, O_j$ have at least one common point $q_c$ can be composed of all $O'_j(q_c)$ for every $q_c \in O_i$:

$$O_{ij} = \bigcup_{q_c \in O_i} O'_j(q_c)$$

(13)

This allows us to construct $O_{ij}$ graphically, as shown in Figure 3, by taking the mirrored shape $O'_j(q_c)$ and deriving the Minkowski sum of $O_i$ and $O_j$, by sliding $q_c$ along the boundary of $O_i$.

Identifying an obstacle function $\delta_{ij}$ such that $\delta_{ij}(q_i, q_j) \leq 0 \iff (q_i, q_j) \in O_{ij}$ and $\delta_{ij}(q_i, q_j) = 0 \iff (q_i, q_j) \in \partial O_{ij}$ is not straightforward in general. Following the previous analysis, one may suggest the use of $\min \psi_i(\delta_i, \delta_j)$. Although this option is valid, it is not always practical, since the minimum of $\psi_i(\delta_i, \delta_j)$ is not easy to derive in general, especially through analytic calculations. However, this does not prevent the use of the method in ATM, since conflict avoidance constraints in this application are expressed with respect to the relative position of neighboring aircraft. Thus, $\delta_{ij}$ can be directly derived and the methodology described in the previous paragraph is immediately applicable.

**B. Local Sensing**

A limited sensing region $\tilde{A}_i$ around agent $i$ is used to restrict the sensing of other agents and obstacles by agent $i$: only those agents and obstacles that are inside $\tilde{A}_i$ can influence the potential $\Phi_i$. Similarly to $O_i$ above, $\tilde{A}_i$ is defined in the local coordinate frame of agent $i$ via the zero level set of the relative sensing function $\tilde{s}_i(q^i)$:

$$\tilde{A}_i \triangleq \{ q^i : \tilde{s}_i(q^i) = 0 \}$$

$$\partial \tilde{A}_i = \{ q^i : \tilde{s}_i(q^i) = 0 \}$$

(14)

(15)
In global coordinates the absolute sensing region $A_i$ is described by the implicit sensing function $s_i(q_i, q) = \tilde{s}_i(q - q_i)$:

$$A_i(q_i) \triangleq \{ q : s_i(q_i, q) = 0 \}$$

$$\partial A_i(q_i) = \{ q : s_i(q_i, q) = 0 \}$$

Obviously, each agent $i$ should be able to sense another agent or obstacle $j$ before an actual collision between them occurs. Thus, the set $O_j^i(q_i) = \{ q \lvert (q_i, q) \in O_{ij} \}$, which is the set of all $q_j$ that cause a collision for a given position $q_i$ of agent $i$, must always be completely contained inside the sensing region $A_i$ for all pairs of $i,j$. When inside the set $A_i(q_i) \setminus O_j^i(q_i)$, the agent (or obstacle) $j$ can influence the potential $\Phi_i$. Thus the properties of the functions $\delta_i$ and $s_i$, which will be used to derive $\beta_{ij}$, inside this region are important for the overall behavior of the potential $\Phi_i$.

### C. Collision function synthesis

As explained above, the collision function $\beta_{ij}$ is active only when $q_j \in A_i(q_i) \setminus O_j^i(q_i)$, i.e. inside $A_i$ where $s_i \leq 0$, and outside $O_j^i$, where $\delta_{ij} \geq 0$. Consequently, we can use the “conditioning” diffeomorphism $\sigma_\lambda$ from [12] to map $-\lambda s_i \in [0, +\infty)$ to $[0, 1]$:

$$\tilde{\beta}_{ij} \triangleq \left( \sigma_\lambda \circ \frac{\delta_{ij}}{s_i} \right) = \frac{\delta_{ij}}{\delta_{ij} - \lambda s_i} \quad (11)$$

where $\sigma_\lambda \triangleq \frac{x}{\lambda + x}$, $\lambda > 0$

By the above definition, $\tilde{\beta}_{ij}$ is zero when agents $i$, $j$ touch (i.e. when $\delta_{ij} = 0$ because $(q_i, q_j) \in \partial O_{ij}$), and increases up to 1 at the boundary of the sensing area $A_i$ where $s_i = 0$.

When $q_j$ is outside the sensing region $A_i$, the influence $\beta_{ij}$ should be inactive by assuming a constant value of 1. Thus, in order to ensure that $\beta_{ij}$ will be $C^2$, we apply a shaping function $L(x)$ to $\tilde{\beta}_{ij}$:

$$L(x) = x^3 - 3x^2 + 3x \quad (12)$$

The following properties hold for $L(x)$:

- $L(0) = 0$
- $L'(x) > 0 \forall x \in [0, 1]$
- $L(1) = 1$
- $L'(1) = L''(1) = 0$

Thus, the final collision function used is:

$$\beta_{ij} = \begin{cases} 
L(\tilde{\beta}_{ij}), & \tilde{\beta}_{ij} \leq 1 \\
1, & \tilde{\beta}_{ij} > 1 
\end{cases} \quad (14)$$

### D. The workspace bounding obstacle

The function $\beta_{\delta_0}$ models a special obstacle which ensures that all agents remain inside the available workspace $W \subset \mathbb{R}^n$. For its construction we follow an similar approach to the one presented for $\beta_{ij}$ above. We assume that the workspace is implicitly defined through a scalar function $\delta_0(q)$:

$$W \triangleq \{ q \lvert \delta_0(q) \leq 0 \}$$

$$\partial W \triangleq \{ q \lvert \delta_0(q) = 0 \}$$

Moreover, we define a region $F$ contained inside $W$, where the influence of the workspace boundary is eliminated:

$$F \triangleq \{ q \lvert s_0(q) \leq 0 \}$$

$$\partial F \triangleq \{ q \lvert s_0(q) = 0 \}$$

Thus, we allow the agents to be affected by the workspace boundary only inside the annulus $W \setminus F$, as shown in Figure 4. In this region $\delta_0 = \delta_0(q_i) \leq 0$ and $s_0 = s_0(q_i) \geq 0$, thus, similarly to the synthesis of $\tilde{\beta}_{ij}$, we use $\sigma_\lambda$ to map $-\delta_0 / s_0 \in [0, +\infty]$ to $\tilde{\beta}_{00} \in [0, 1]$:

$$\tilde{\beta}_{00} \triangleq \left( \sigma_\lambda \circ \frac{\delta_0}{s_0} \right) = \frac{\delta_0}{\delta_0 - \lambda s_0} \quad (19)$$

One can see that $\tilde{\beta}_{00}$ is zero on $\partial W$ and becomes 1 on the boundary of the influence area, $\partial F$. Consequently, the workspace obstacle function is derived by applying the $L(x)$ mapping to $\tilde{\beta}_{00}$ to ensure it is $C^2$:

$$\beta_{00} = \begin{cases} 
L(\tilde{\beta}_{00}), & \tilde{\beta}_{00} \leq 1 \\
1, & \tilde{\beta}_{00} > 1 
\end{cases} \quad (20)$$

### IV. Decentralised aircraft conflict resolution in 3D space

The application considered in this paper is the decentralised navigation of a group of $N$ independent aircraft in 3D airspace. Each aircraft is modeled as a kinematic agent with its configuration comprising its position $q_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^\top$ with respect to an earth-fixed frame $\mathcal{E}$, and its heading $\phi_i$. The global $x$ and $y$ axes lie on the horizontal plane, while axis $z$ is the vertical (altitude) axis. The heading angle $\phi_i$ is the angle of the aircraft’s horizontal velocity with respect to the global $x$ axis. Using $n_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^\top$ as the projection of the agent’s position on the horizontal $x - y$
plane, the kinematic model of each aircraft $i$ is:

\[
\begin{align*}
\dot{x}_i &= \dot{\theta}_i = J_i \cdot u_i, \\
\dot{y}_i &= w_i, \\
\dot{\theta}_i &= \omega_i,
\end{align*}
\]  

where $J_i = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) \end{bmatrix}^T$.

A. Collision and sensing regions

The potential construction described in section II can be well adapted to the case of aircraft conflict resolution. Decentralisation, as well as the ability to use 3D resolutions to increase airspace capacity and separation are very useful for ATM applications, like the future ATM concept developed within the project iFLY [14].

In order to apply the methodology described in the previous sections to aircraft conflict resolution, we define accordingly the shapes used for the collision set $O_{ij}$ and the sensing area $A_i$ of each aircraft. For $O_{ij}$ we employ a “pill-like” 3D oblate spheroid with horizontal radius equal to the minimum horizontal separation $d_h$ and a vertical semi-axis equal to the vertical minimum separation $d_v$, as shown in Figure 5. The principle in [11] is extended to the 3D case for $A_i$: the sensing range in the forward direction is significantly longer than the sides and rear, as shown in Figure 6. The benefits of such a sensing scheme have been demonstrated in [11], allowing smoother turns and smaller deviations in the resulting trajectories. The sensing region comprises two half-ellipsoids, one in front of the aircraft with semi-axes $R_f$, $R_r$ and $R_v$ in the forward, side and vertical directions respectively, and one in the rear with semi-axes $R_r$, $R_v$ and $R_v$. Thus, the two cross-sections on the plane normal to the major axis $x$ of the aircraft match exactly, while a longer range $R_w$ is applied forward compared to the shorter side and rear range $R_r$. As shown in [11], such a sensing scheme introduces prioritisation, since the interaction between two neighboring agents is unsymmetrical.

In order to model the shape of $O_{ij}$ with the implicit collision function $\delta_{ij}$ we use the standard ellipsoid formula:

\[
\delta_{ij} = \frac{(x_i^j)^2}{R_f^2} + \frac{(y_i^j)^2}{R_r^2} + \frac{(z_i^j)^2}{R_v^2} - 1
\]  

where $q_i^j = \begin{bmatrix} x_i^j & y_i^j & z_i^j \end{bmatrix}$ is the position of aircraft $j$ in the local coordinate frame of aircraft $i$; $x_i^j$ is along the longitudinal direction of aircraft $i$, $y_i^j$ along the lateral direction and $z_i^j$ on the vertical one:

\[
T_i = \begin{bmatrix} \cos(\phi_i) & -\sin(\phi_i) & 0 \\
\sin(\phi_i) & \cos(\phi_i) & 0 \\
0 & 0 & 1 \end{bmatrix}
\]  

Similarly, for the sensing function $s_{ij}$ we use the form:

\[
s_{ij} = \frac{(x_i^j)^2}{R_f^2} + \frac{(y_i^j)^2}{R_r^2} + \frac{(z_i^j)^2}{R_v^2} - 1
\]  

\[
R_f = \begin{cases} R_f, & x_i^j \geq 0 \text{ (agent } j \text{ in front of } i) \\
R_r, & x_i^j < 0 \text{ (agent } j \text{ behind } i) \end{cases}
\]

B. 3D motion control scheme for aircraft-like vehicles

By applying the forms for $s_{ij}$ and $\delta_{ij}$ to the NF construction described in II, we can create a potential that corresponds to the specific requirements of aircraft conflict resolution. In order to guide each aircraft $i$ using the potential $\Phi_i$, the control scheme presented by the authors in [15] can be employed. This control policy is specially designed to allow each aircraft to move towards lower potential values with a constant speed, while respecting climb and descent angle limits and reducing unnecessary steering.

V. RESULTS

The control scheme presented here has been employed in a simulation scenario, corresponding to aircraft Short-term conflict resolution, according to ATM standards. The
horizontal separation minimum $d_h$ has been set to 5 nautical miles (nm), while the vertical separation minimum used is 1000 feet (ft). The sensing ranges have been set as follows:

- $R_f = 40$nm for the forward sensing range, corresponding to about 5 minutes of flight with a typical cruising speed of 480 knots. This is in accordance with the iFLY Concept of Operations [14] that specifies 5 minutes as the limit for Short-term conflict resolution.
- $R_r = 15$nm, reducing influence from other aircraft in the rear and sides to improve the trajectories.
- $R_v = 1500$ft, in order to avoid excessive separation by more than two flight levels (2000 ft).

First, a simple scenario with 2 aircraft presents the principles of operation of the algorithm. The aircraft are flying in opposite direction with an initial vertical separation of only 500 ft. As shown in Figure 7, the aircraft perform vertical manoeuvres to increase their separation and then revert back to their desired altitude as they approach their goals.

The second scenario includes 5 aircraft flying at the same altitude in converging paths. As shown in Figure 8, vertical maneuvering enables safe resolution of all conflicts.

VI. CONCLUSIONS

We have presented a way to extend the application of Navigation Functions (NFs) to problems with agents of general shapes and sensing schemes. The scalar functions that implicitly describe the shapes of the sensing regions and the agents themselves are used to synthesise the obstacle functions for the potential. This NF construction enables the adaptation of the methodology to various applications by using appropriate shape functions. Application to the case of distributed aircraft conflict resolution is presented, for which NFs are a candidate solution. The specific requirements of this problem are considered for the definition of the agent and sensing region shapes. The algorithm is used in simulation scenarios to demonstrate its efficacy.

VII. ACKNOWLEDGEMENTS

The authors of this paper want to acknowledge the contribution of the European Commission through project iFLY.

REFERENCES


7475