HYBRIDGE

Distributed Control and Stochastic Analysis of Hybrid Systems
Supporting Safety Critical Real-Time Systems Design

WP6: Decentralized Conflict Prediction and Resolution

Intermediate Report on Decentralized Conflict Resolution Algorithms

Dimos V. Dimarogonas[1] and Kostas J. Kyriakopoulos

October 25, 2004

Version:	*0.4*
Task number:	*6.2*
Deliverable number:	*D6.2i*
Contract:	*IST-2001-32460 of European Commission*

[1] National Technical University of Athens (NTUA).

DOCUMENT CONTROL SHEET

Title of document: Intermediate Report on Decentralized Conflict Resolution Algorithms

Authors of document: *Dimos V. Dimarogonas and Kostas J. Kyriakopoulos*
Deliverable number: *D6.2i*
Contract: *IST-2001-32460 of European Commission*
Project: *Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design (HYBRIDGE)*

DOCUMENT CHANGE LOG

Version #	Issue Date	Sections affected	Relevant information
0.1	14-12-2003	All	First draft
0.2	19-01-2004	All	Second Draft
0.3	19-05-2004	2	UCAM comments incorporated
0.4	25-10-2004	1,4	NLR comments incorporated

Version 2.0		Organisation	Signature/Date
Authors	Dimos V. Dimarogonas	NTUA	
	Kostas J. Kyriakopoulos	NTUA	
Internal reviewers	Andrea Lecchini	UCAM	
	Henk Blom	NLR	

HYBRIDGE, IST-2001-32460
Work Package WP6, Deliverable D6.2i

Intermediate Report on Decentralized Conflict Resolution Algorithms

Prepared by:

Dimos V. Dimarogonas and Kostas J. Kyriakopoulos

Control Systems Laboratory, Mechanical Eng. Dept.
National Technical University of Athens, Greece
{ddimar@mail.ntua.gr, kkyria@central.ntua.gr}

Abstract

This is the second deliverable under work package WP6 of the HYBRIDGE project.

This report aims to summarize the work held so far under WP6 on decentralized conflict resolution. We use the navigation function methodology to solve the problem of decentralized motion planning of multiple agents in both cases where the agent dynamics are holonomic or nonholonomic. Simulation results in both cases indicate the effectiveness of the method.

In the last part of the report, a brief presentation of the current research efforts taking place in the Control Systems Lab of NTUA under WP6 is given. These involve the extension of the underlying methodology to the case where the sensing capabilities of each agent are limited and where uncertainty has to be taken into account.

CONTENTS

Chapter 1

Introduction

This deliverable summarizes the work held so far on decentralized conflict resolution in NTUA under WP6. It presents the extension of navigation functions, which have been proven a very powerful tool for centralized navigation and collision avoidance, to decentralized navigation, for both the cases where the aircraft dynamics are considered holonomic or nonholonomic. The underlying work is held under the guidance of Task 6.2 of WP6.

The navigation functions approach is a method related to the artificial potential fields methodologies in robotics. These methods have been used extensively in the last two decades in robot path planning and cooperative control both for holonomic and nonholonomic dynamics. Nonholonomic constraints arise when the velocity constraints of a moving vehicle cannot be written as an algebraic constraint in the configuration space. When the constraints are explicitly integrable, then they can take the form of an algebraic constraint. Hence one can relate the words holonomic and nonholonomic to integrable and non-integrable respectively.

In Air Traffic Management, decentralized conflict detection and resolution involves reassignment of the control tasks from the central authority, i.e. the Air Traffic Controllers, to the agents, i.e. the cockpit. The level of decentralization depends on the knowledge an agent has on the other agents' actions and objectives. In this deliverable the decentralization factor lies in the fact that each agent/aircraft has knowledge only of its own desired destination, but not of the desired destinations of the others. Clearly, this is a suitable model for a futuristic distributed ATM system, where each aircraft will have knowledge of the actions and positions of the other aircraft at each time instant, but not of their destinations.

Chapters 2 and 3 present the decentralized navigation function methodology for the holonomic and nonholonomic case respectively. In the first case, we show that by appropriate selection of the controller parameters, the (asymptotic) stability of the whole scheme is guaranteed. This is also proved in the nonholonomic case, where we enhance the holonomic navigation functions of Chapter 2 with a *dipolar* structure, so that the nonholonomic constraints are dealt with. Simulation results in both cases indicate the effectiveness of the method.

The final chapter of this report summarizes current research issues within Task 6.2. In the prescribed methodology each agent has global knowledge of the positions of the others at each time instant. In practice however, the sensing capabilities of each aircraft are limited. Consequently, each aircraft cannot have knowledge of the positions and/or velocities of every aircraft in the workspace but only of the aircraft within its *sensing zone* at each time instant. We discuss in section 4.1 how such a situation can be modelled as a deterministic hybrid system and indicate ways to check the stability of the overall scheme. Finally, section 4.2 discusses possible extensions of the underlying method to the case where uncertainty in the plant’s and/or pilot’s behaviour is taken into account.

Chapter 2

Decentralized Conflict Resolution for Multiple Holonomic Agents

In this chapter, we consider the decentralized conflict avoidance problem for the case when the dynamics of each aircraft are considered purely holonomic, i.e. the position of aircraft *i* is described by the following differential equation: $\dot{q}_i = u_i$, where u_i is the velocity of the aircraft. The problem that we treat can be stated as follows: "*Derive a set of control laws (one for each agent) that drives a team of n aircraft from any initial configuration to a desired goal configuration avoiding at the same time collisions.*" We make the following assumptions:

- Each aircraft has global knowledge of the position of the others at each time instant.
- Each aircraft has knowledge only of its own desired destination but not of the others.
- We consider spherical agents.
- The agent's motion is planar, i.e. we consider 2D movement.
- The workspace is bounded and spherical.

The second assumption makes the problem decentralized. Clearly, in the centralized case a central authority has knowledge of everyone's goals and positions at each time instant and it coordinates the whole team so that the desired specifications (destination convergence and collision avoidance) are fulfilled. In the current situation no such authority exists and we have to deal with the limited sensing capabilities of each agent.

We approach the problem with the navigation function method, which has been used for the centralized case in [12]. The following discussion is based on [3]. The proofs of the aforementioned propositions and more details can be found in [4].

2.1 Decentralized Navigation Functions

Navigation functions are real valued maps realized through cost functions, whose negated gradient field is attractive towards the goal configuration and repulsive wrt obstacles. It has been shown by Koditscheck and Rimon that "almost" global navigation is possible since a smooth vector field on any sphere world with a unique attractor, must have at least as many saddles as obstacles [9]. Our assumption about spherical agents does not constrain the generality of this work since it has been proven that navigation properties are invariant under diffeomorphisms.

We consider a team of *n* aircraft/agents operating in a spherical workspace $W \subset R^2$. Each agent *i* occupies a disc of radius r_i. Its center is denoted q_i. The configuration space is spanned by $q = [q_1, \ldots, q_n]^T$. A navigation function is a map $\varphi : F \to [0,1]$, where *F* is a compact connected analytic manifold with boundary, which (1) is analytic on *F*, (2) has only one minimum in the interior of *F*, (3) its Hessian $\partial^2 \varphi / \partial q^2$ at all critical points (zero gradient vector field) is full rank, and (4) is maximized on the boundary of *F*, i.e. $\lim_{q \to \partial F} \varphi(q) = 1$.

Property (3) of the above definition guarantees that the flow of the vector field is not "blocked" towards the desired destination by critical points not coinciding with it. Property (1) reflected Koditscheck and Rimon's perspective that analytic expressions are "a preferable encoding of actuator commands to algorithms that include logical decisions". They hint however, that merely C^2 functions are needed.

We consider the following class of decentralized navigation functions: $\varphi_i = \sigma_d \circ \sigma \circ \hat{\varphi}_i = \left(\frac{\gamma_i}{\gamma_i + G_i}\right)^{1/k}$, where $\sigma_d = x^{1/k}, \sigma = \frac{x}{1+x}$, and the cost function $\hat{\varphi}_i = \gamma_i / G_i$, where $\gamma_i^{-1}(0)$ denotes the desirable set (desired destination), and $G_i^{-1}(0)$, the set that aircraft *i* wants to avoid (collisions with other aircraft). A suitable choice is $\gamma_i = \gamma_{di}^k = \left\| q_i - q_{di} \right\|^{2k}$, where γ_{di} is the squared metric of the current agent's configuration q_i from its destination q_{di}. Function G_i has as arguments the coordinates of all agents, i.e. $G_i = G_i(q)$, in order to express all possible collisions of agent i with the others. The proposed navigation function for agent i is $\varphi_i(q) = \gamma_{di} / (\gamma_{di}^k + G_i)^{1/k}$ and the corresponding agent control law is $u_i = -K_i \bullet \frac{\partial \varphi_i}{\partial q_i}$.

The following theorem will help us on deriving results for the function φ_i by examining the simpler function $\hat{\varphi}_i$:

Theorem 1 [4]: *Let $I_1, I_2 \subseteq R$ be intervals, $\hat{\varphi}: F \to I_1$ and $\sigma: I_1 \to I_2$ be analytic. Define the composition $\varphi: F \to I_2$ to be $\varphi = \sigma \circ \hat{\varphi}$. If σ is monotonically increasing on I_1, then the set of critical points of $\hat{\varphi}$ and φ coincide and the (Morse) index of each critical point is identical.*

The first step is to prove the existence of an energy function that asymptotically stabilizes the system to $q = [q_1, \ldots, q_n]^T$. The obvious choice is to choose the sum of the separate decentralized navigation functions, i.e. to choose $\varphi = \sum_{i=1}^{n} \varphi_i$.

Proposition 1*: The derivative of φ assumes negative values up to a set of measure zero if the exponent k assumes values bigger than a finite lower bound.*

This set of measure zero corresponds to saddle points of the navigation functions. The third property of the definition indicates that the critical points of each navigation function are isolated and that the set of initial conditions that leads to saddle points is of measure zero. We use the result of theorem 1, to show that the critical points of the navigation functions are such saddle points, so that there is always a direction of movement decreasing its potential function inside the free space, i.e. inside the subspace of *W* which is free of collisions. Similar to the results in [12], we make use of the following propositions:

Proposition 2: *If the workspace is valid, the destination point q_{di} is a non–degenerate local minimum of φ_i.*

Proposition 3: *If the workspace is valid, all critical points of* φ_i *are in the interior of the free space.*
Proposition 4: *For every ε>0, there exists a positive integer N(ε) such that if k>N(ε) then there are no critical points of* $\hat{\varphi}_i$ *in* $F_1(\varepsilon)$*, where* $F_1(\varepsilon)$ *denotes the set away from collisions.*
Proposition 5: *For any valid workspace, there exists an* $\varepsilon_0 > 0$ *such that* $\hat{\varphi}_i$ *has no local minimum in* $F_0(\varepsilon)$*, as long as* $\varepsilon < \varepsilon_0$*, where* $F_0(\varepsilon)$ *denotes the set near collisions.*

Clearly, Proposition 2 guarantees that the desired destination is the only local minimum of φ_i inside the free space of aircraft *i*, Propositions 3,4 clear the set away from collisions from critical points and Proposition 5 establishes that even when a critical point occurs near the collision set, it is never a local minimum so there is always a direction of movement decreasing the potential function inside the free space. This, along with the fact that the Hessian at all critical points is full rank, guarantees that the set of initial conditions that leads to saddle points is of measure zero.

2.2 Construction of the *G*-function

Unlike the centralized case, in the proposed decentralized control law, each agent has a different G_i which represents its relations with all the other agents. To simplify notation we denote by *q* instead of q_i the current agent configuration, by q_d instead of q_{di} its goal configuration, by *G* instead of G_i its *"G"* function and by q_j the configurations of the other agents. Actually, each agent treats the remaining *n*-1 agents as *n*-1 moving obstacles. We use this terminology in the following paragraphs. The mathematical tools of the following paragraphs are a simple extension of the notions introduced in [12] to the decentralized setup.

A "*Robot Proximity Function*", a measure of the distance between the agent and the *j-th* moving obstacle in the workspace, is defined by: $\beta_j = \|q - q_j\|^2 - (r + r_j)^2$, where *r* is the radius of the agent and r_j the radius of the *j-th* moving obstacle.

We will use the term *"relation"* to describe the possible collision schemes that can be defined in a *single agent – multiple moving obstacles* scene. A *binary relation* is a relation between the agent and a single obstacle. We will call the number of binary relations in a relation, the *"relation level"*. With this terminology in hand, the relation of figure *(1a)* is a *level-1* relation (one binary relation) and that of figure *(1b)* is a *level-3* relation (three binary relations), where with *R* is denoted the agent though O_j indicate the obstacles.

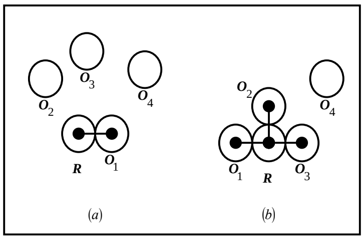


Fig.2.1

A *"Relation Proximity Function"(RPF)* provides a measure of the distance between the agent and the obstacles involved in a relation. Each relation has its own *RPF*. An *RPF* assumes the value of zero whenever the agent – obstacles involved in the relation collide and increases wrt the distance of the related objects: $(b_{R_j})_l = \sum_{m \in (R_j)_l} \beta_m$, where the index j denotes the *j-th* relation of *level–l*. To simplify notation, the relation proximity function can be rewritten as: $b_i = \sum_{j \in R_l} \beta_j$, where R_l indicates *level–l* relations and the index i belongs to the set S of all possible relations wrt to the specific agent. Obviously, i indicates a relation of *level–l*.

A *"Relation Verification Function" (RVF)* is defined by:
$(g_{R_j})_l = (b_{R_j})_l + \frac{\lambda (b_{R_j})_l}{(b_{R_j})_l + (B_{R_j^C})_j^{1/h}}$, for $l \leq n-2$ and $(g_{R_j})_l = (b_{R_j})_l$ for l=n-1 where λ, h are positive constants, n is the total number of agent – obstacles in the workspace, and: $(B_{R_j^C})_l = \prod_{m \in (R_j^C)_l} (b_m)_l$, or $\tilde{b}_i = \prod_{m \in R_j^C} b_m$ for simplicity where in the simplified equation, R_j^C indicates a complementary set of relations of *level–l*. Using the simplified notation introduced above, the relation verification function can be rewritten as: $g_i(b_i, \tilde{b}_i) = b_i + \frac{\lambda b_i}{b_i + \tilde{b}_i^{1/h}}$, for $l \leq n-2$ and $g_i(b_i, \tilde{b}_i) = b_i$ for l=n-1 where n is the total number of agents in the workspace, as defined previously. The basic property that we demand from *RVF* is that it assumes the value of zero if a relation holds, while no other relations of the same or other levels hold. In other words it should indicate which of all possible relations holds. In *RVF*'s definition we distinguish two situations (i.e. $l \leq n-2$ and l=n-1) since for l=n-1, only one relation is defined and so the set $(R_j^C)_{n-1}$ is an empty set. Thus we can't define $(B_{R_j^C})_{n-1}$. We could compute the following limits of *RVF* (using the simplified notation): when $b_i \to 0$ and $\tilde{b}_i \neq 0$, obviously: $g_i \to 0$. When: $b_i \to 0$ and $\tilde{b}_i \to 0$, because of the power *1/h* on $\tilde{b}_i$, it tends to zero faster than b_i does, and so we conclude that: $g_i \to \lambda$. When: $b_i \neq 0$, independently of how $\tilde{b}_i$ behaves: $g_i \neq 0$. These limits guarantee that *RVF* will behave in the way we want it to, as an indicator of a specific collision. We can now define: $G = \prod_{l=1}^{n_L} \prod_{j=1}^{n_{R,l}} (g_{R_j})_l$, where n_L is the number of levels and $n_{R,l}$ the number of relations in *level-l*. The simplified equation indicates that G is practically the product of a certain number of g_i's.

The definition of the G function in the multiple moving agents situation is slightly different than the one introduced by the authors in [9]. The collision scheme in that approach involved a single moving point agent in an environment with static obstacles. A collision with more than one obstacle was therefore impossible and the obstacle function was simply the product of the distances of the agent from each obstacle. In our case however, this is inappropriate, as can be seen in the following figure. The control law of agent A should distinguish when agent A is in conflict with B, C, or B and C simultaneously. Mathematically, the first two situations are level-1 relations and the third a level-2 relation with respect to A. Whenever the latter occurs, the RVF of the level-2 relation tends to zero while the RVF's of the two separate level-1 relations (A,B and A,C) are nonzero. The key property of an RVF is that it tends to zero only when the corresponding relation holds. Hence it serves as an analytic switch that is activated (tends to zero) only when the relation it represents is realized.

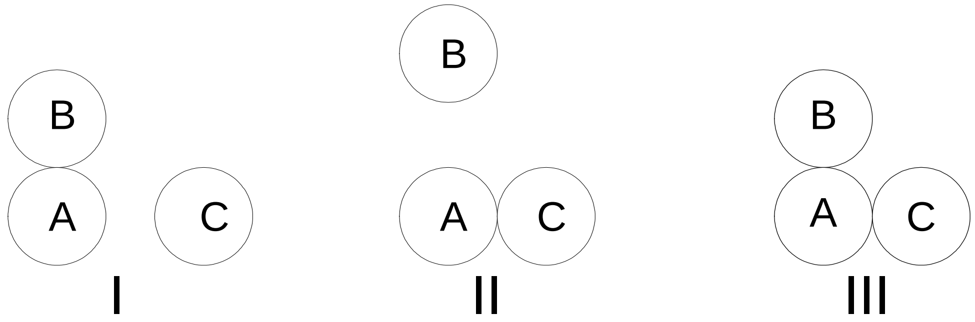


Fig.2.2
I,II are level-1 relations with respect to A, while III is level-2. The RVF's of the level-1 relations are nonzero in situation III.

2.3 Proof of Correctness

For a detailed proof of Propositions 1-5, the reader is referred to [4]. We first proceed with the proof of Proposition 1 and then move on with the proofs of Propositions 2-5, which are simple extensions of the proofs in [12]. For the latter we make use of the following geometry: let $\varepsilon > 0$. Define $B_{j,l}(\varepsilon) \equiv \{q : 0 < (g_{R_j})_l < \varepsilon\}$. We can then discriminate the following topologies:

1. The *destination point*: q_d
2. The *free space boundary*: $\partial F(q) = G^{-1}(0)$
3. The set *near collisions*: $F_0(\varepsilon) = \bigcup_{l=1}^{n_L} \bigcup_{j=1}^{n_{R,l}} B_{j,l}(\varepsilon) - \{q_d\}$
4. The set *away from collisions*: $F_1(\varepsilon) = F - (\{q_d\} \cup \partial F \cup F_0(\varepsilon))$

Proposition 1 guarantees asymptotic stability to the destination point, while 2-5 guarantee that there will always be a direction of decrease of the potential function inside the free space.

2.4 The *f*-function

The prescribed method does not apply to the case when the initial conditions of some of the agents coincide with their desired destinations. This is because in these cases the numerator of φ_i so the potential for an agent to move is negligible in a possible collision scheme. A way to overcome this is to add a function *f* so that the cost function φ_i attains bigger positive values in proximity situations even when *i* has reached its destination. The navigation function in this case becomes

$\varphi_i(q) = \dfrac{(\gamma_{di} + f(G_i))}{((\gamma_{di} + f(G_i))^k + G_i)^{1/k}}$. A suitable function is $f(G) = a_0 + \sum_{j=1}^{3} a_j G^j$ for $0 \le G \le X$ and $f(G) = 0$ for G>X. This function is locally maximum at G=0 and locally maximum at G=X, where X,Y>0. The coefficients a_i are evaluated in order to fulfil these properties. This choice of f has been proven to be very satisfying in simulation. The problem is that in this way the function φ_i is no longer analytic so it does not fulfil the definition of navigation functions given by Koditscheck and Rimon. However, as we have discussed previously, the analyticity property can be replaced by mere second order differentiability.

2.5 Simulation Results

To demonstrate the navigation properties of our decentralized approach, we present a simulation of four holonomic agents that have to navigate from an initial to a final configuration, avoiding collision with each other and satisfying velocity bounds. The chosen configurations constitute non-trivial setups since the straight-line paths connecting initial and final positions of each agent are obstructed by other agents.

In the first simulation (Simulation A) we use navigation functions without the function f in the nominator. In the second (Simulation B) we use the function f in the nominator.

<u>Initial Conditions:</u> $q_1 = [.1732, -.1]^T$, $q2 = [-.1732, -.1]^T$,$q_3 = [0, .2]^T$,$q_4 = [0, 0]^T$.
<u>Goal Conditions:</u> $q_{d1} = [-.1732, .1]^T$, $qd2 = [.1732, .1]^T$,$q_{d3} = [0, -.2]^T$,$q_{d4} = [0, 0]^T$.

The fulfillment of the collision avoidance as well as of the destination convergence properties is obvious in the following five figures. Notice that in Simulation A, since the green agent's initial condition coincides with its desired destination, it does not participate in the navigation process. Hence there is no cooperation between the green agent and the remaining team. This is overcome in Simulation B which successfully forces the green agent to participate in the navigation process.

Simulation A (navigation function:

$\varphi_i(q) = \gamma_{di} / (\gamma_{di}^k + G_i)^{1/k}$)

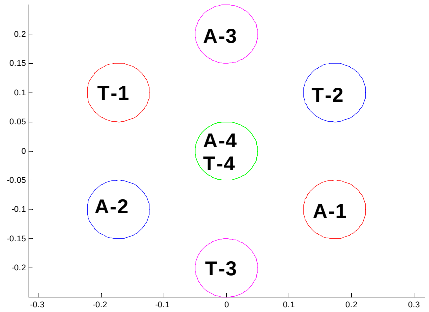


Fig.A

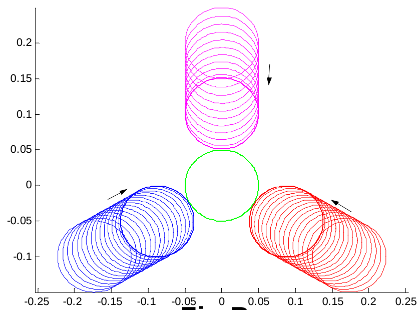


Fig.B

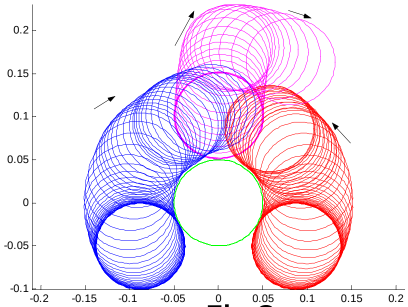


Fig.C

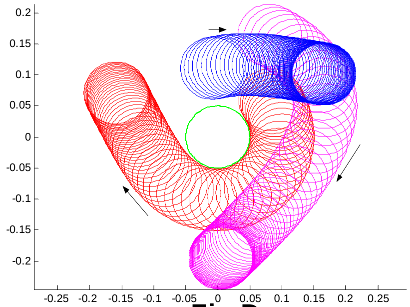


Fig.D

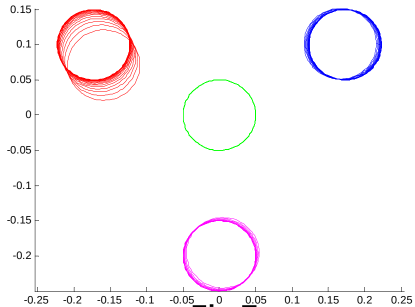


Fig.E

Simulation B (navigation function:

$$\varphi_i(q) = \frac{(\gamma_{di} + f(G_i))}{((\gamma_{di} + f(G_i))^k + G_i)^{1/k}}\ )$$

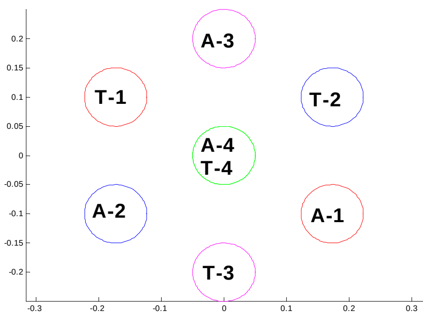


Fig.A

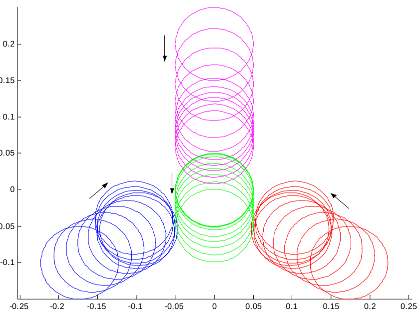


Fig.B

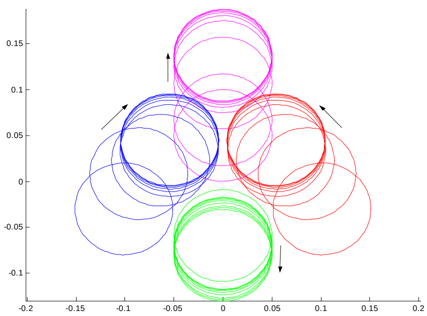


Fig.C

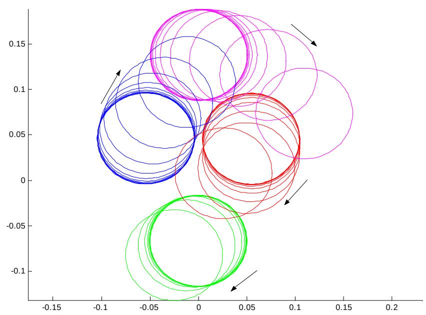


Fig.D

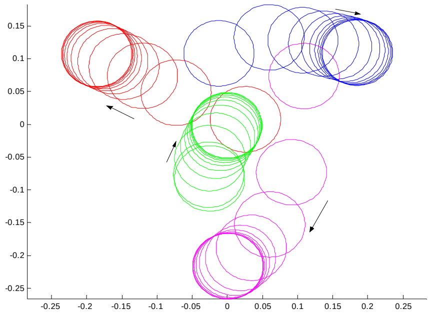


Fig.E

Chapter 3

Decentralized Conflict Resolution for Multiple non-Holonomic Agents

In this chapter, we discuss the problem of decentralized collision avoidance for multiple agents but in this case, the system dynamics are non-holonomic, i.e. the position of aircraft i is described by the following differential equations: $\dot{x}_i = u_i \cos\theta_i, \dot{y}_i = u_i \sin\theta_i, \dot{\theta}_i = \omega_i$, where u_i is the translational velocity of the aircraft, and ω_i its angular velocity. The problem that we treat can be stated as follows: "*Given a team of n nonholonomic aircraft derive a control law that steers every system from any feasible initial configuration to its goal configuration avoiding at the same time collisions.*" We make the following assumptions:

- Each aircraft has global knowledge of the position and velocity of the others at each time instant.
- Each agent has no information about other agents' targets.
- Around the target of each agent *i* there is a region called agent's *i* safe region.
- Agent's *i* safe region is only accessible by agent *i*, while regarded as an obstacle by other agents.
- The minimum distance between any two safe regions of any two agents is greater than the diameter of the largest agent.

The following discussion is based on [11].

3.1 Decentralized Nonholonomic Navigation Functions

In this section, we show how the decentralized holonomic navigation functions of the previous section can be extended in order to provide trajectories suitable for the nonholonomic case. This is accomplished by a enhancing a *dipolar* structure [20] to the navigation functions. Dipolar potential fields have been proven a very effective tool for stabilization [19] as well as for centralized coordination of multiple nonholonomic agents [13]. The key advantage of this class of potential fields is that they drive the controlled agent to its destination with the desired orientation. The following discussion, which is based on [11], provides the extension of [3] to the case of multiple nonholonomic agents.

We consider a team of *n* aircraft/agents operating in a spherical workspace $W \subset R^2$. Each agent *i* occupies a disc of radius r_i. Its center is denoted q_i. The position vector of the agents is represented by $q = [q_1, \ldots, q_n]^T$. The orientation vector of the agents is represented by $\theta = [\theta_1, \ldots, \theta_n]^T$, where θ_i denotes the orientation of each agent. Let $W_i \subseteq R^2 \times (-\pi, \pi]$ represent each agent's workspace. The configuration of each agent is represented by $p_i = [q_i \ \theta_i]^T \in W_i$ and its target by $p_{di} = [q_{di} \ \theta_{di}]^T \in W_i$.

To be able to produce a dipolar potential field, φ_i of the previous section must be modified as follows: $\varphi_i = \dfrac{\gamma_{di}}{\left(\gamma_{di}^k + H_{nh_i} \cdot G_i\right)^{1/k}}$, where H_{nh_i} has the form of a pseudoobstacle. A possible selection of H_{nh_i} would be $H_{nh_i} = \varepsilon_{nh} + \eta_{nh_i}$, with $\eta_{nh_i} = \left\|(q_i - q_{di}) \cdot n_{di}\right\|^2$, where $n_{di} = [\cos(\theta_{di}) \sin(\theta_{di})]^T$. Subscript d denotes destination. Moreover $\gamma_{di} = \left\|q_i - q_{di}\right\|^2$, i.e. the angle is not incorporated in the distance to the destination metric. Figure shows a 2D dipolar navigation function.

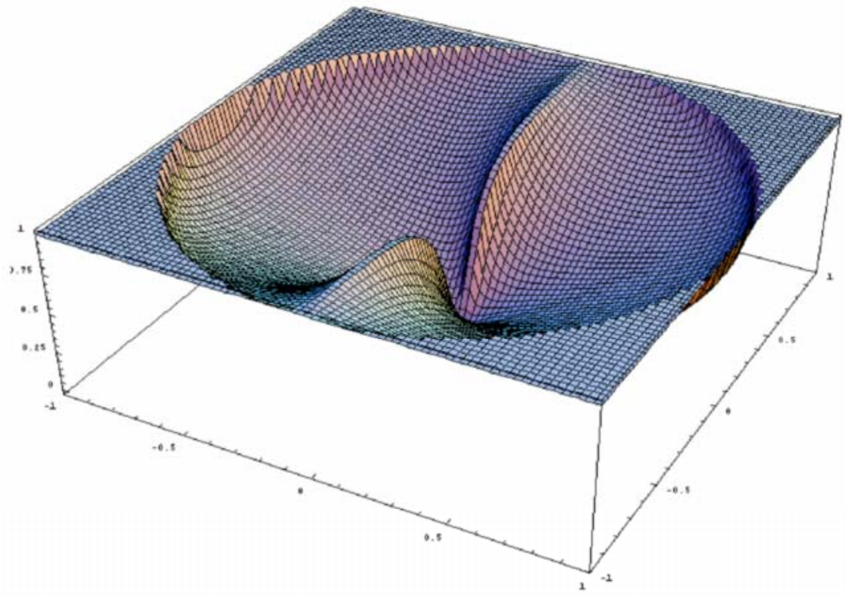


Fig. 3.1 2D dipolar navigation function

As is shown in [13], the proposed modifications of the potential function do not affect its navigation properties, as long as the workspace is bounded and $\varepsilon_{nh} > \varepsilon(k)$.

3.2 Nonholonomic Control

Thus far we have established that the dipolar function φ_i has navigation properties. We consider convergence of the multi-agent system as a two-stage process: In the first stage agents converge to a ball of radius ε called safe region, containing the desired destination of each agent. Each agent can get in its own safe region but not in others. The safe region of one agent is regarded as an obstacle from the other agents. Once an agent gets in its own safe region, it remains in the set and asymptotically converges to the origin.

Before defining the control we need some preliminary definitions:

We define by $\frac{\partial^2}{\partial q_i^2}\varphi_i(q_i,t) = {}^i\nabla^2\varphi_i(q_i,t)$ the Hessian of φ_i. Let $\lambda_{\min}, \lambda_{\max}$ be the minimum and maximum eigenvalues of the Hessian and $\hat{\upsilon}_{\lambda_{\min}}, \hat{\upsilon}_{\lambda_{\max}}$ the unit eigenvectors corresponding to the minimum and maximum eigenvalues of the Hessian. Since navigation functions are Morse functions [9], their Hessian at critical points is never degenerate, i.e. their eigenvalues have always nonzero values.

As discussed before, φ_i is a dipolar navigation function. The flows of the dipolar navigation field provide feasible directions for nonholonomic navigation. What we need now is to extract this information from the dipolar function. To this extend we define the ``nonholonomic angle'':

$$\theta_{nh_i} = \begin{cases} \arg\left(\frac{\partial\varphi_i}{\partial x_i}\cdot s_i + \mathrm{i}\frac{\partial\varphi_i}{\partial y_i}\cdot s_i\right), \neg P_1 \\ \arg\left(d_i\cdot s_i(\upsilon^x_{\lambda_{\min}} + \mathrm{i}\upsilon^y_{\lambda_{\min}})\right), P_1 \end{cases},$$

where condition P_1 is used to identify sets of points that contain measure zero sets whose positive limit sets are saddle points:

$$P_1 = (\lambda_{\min} < 0) \wedge (\lambda_{\max} > 0) \wedge (\left|\hat{\upsilon}_{\lambda_{\min}} \cdot {}^i\nabla\varphi_i\right| < \varepsilon_1)$$

where

$$\varepsilon_1 < \min_{C=\{q_i:\|q_i - q_{di}\|=\varepsilon\}}\left(\left\|{}^i\nabla\varphi_i(C)\right\|\right), s_i = \operatorname{sgn}((q_i - q_{di})\cdot\eta_{di})$$

$$d_i = \operatorname{sgn}(\upsilon_{\lambda_{\min_i}} \cdot {}^i\nabla\varphi_i), \eta_{di} = \left[\cos(\theta_{di})\ \sin(\theta_{di})\ \right]^T$$

$$\eta_i = \left[\cos(\theta_i)\ \sin(\theta_i)\ \right]^T$$

Before proceeding we need the following:

Lemma 1: *If* $\left|\hat{\upsilon}_{\lambda_{\min}} \cdot {}^i\nabla\varphi_i\right| = 0$ *then* P_1 *consists of the measure zero set of initial conditions that lead to saddle points.*

For a proof of this lemma as well as of the following proposition the reader is referred to [11].

In view of Lemma 1, ε_1 can be chosen to be arbitrarily small so the sets defined by P_1 eventually consist of thin sets containing sets of initial conditions that lead to saddle points.

The following provides a suitable nonholonomic controller for the first stage:

Proposition 1: *The system under the control law*

$$u_i = -\operatorname{sgn}({}^i\nabla\varphi_i\cdot\eta_i)\cdot\left(K_{u_i}K_{z_i} + c_i\frac{\left|\partial\varphi_i/\partial t\right|}{\left|{}^i\nabla\varphi_i\cdot\eta_i\right|}\tanh(|\,{}^i\nabla\varphi_i\cdot\eta_i\,|^2)\right)$$

$$\omega_i = \frac{\partial\theta_{nh_i}}{\partial t} + u_i\cdot{}^i\nabla\theta_{nh_i} + (\theta_{nh_i} - \theta_i)\cdot\left(K_{\theta_i} + c_i\frac{\left|\partial\varphi_i/\partial t\right|}{2(\theta_{nh_i} - \theta_i)^2}\tanh(|\,\theta_{nh_i} - \theta_i\,|^3)\right)$$

converges to the set $B_i = \{p_i : \left\|q_i - q_{di}\right\| \le \varepsilon - \delta,\ \ \theta_i \in (-\pi,\pi]\}, i \in \{1,...,n\}$, almost everywhere in W_i. Here $K_{z_i} = \left\|{}^i\nabla\varphi_i\right\|^2 + \left\|q_i - q_{di}\right\|^2$, K_{u_i}, K_{θ_i} are positive constants and $0 < \delta < \varepsilon$. The parameter c_i must be chosen such that $c_i > \frac{\varepsilon_2 + 1}{\varepsilon_2}$ where $\varepsilon_2 = 2\pi^3\varepsilon_1^2\left(4\varepsilon_1 + \sqrt{2}\pi^{3/2}\right)^{-2}$.

For the second stage each agent is isolated from the rest of the system. The dipolar navigation function for this case becomes $\varphi_{\text{int}_i} = \dfrac{\gamma_{d,\theta_i}}{\left(\gamma_{d,\theta_i}^k + H_{nh_i} \cdot \beta_{\text{int}_i}\right)^{1/k}}$, where $\beta_{\text{int}_i} = \varepsilon^2 - \left\| q_i - q_{di} \right\|^2$, and $\gamma_{d,\theta_i} = \left\| q_i - q_{di} \right\|^2 + (\theta - \theta_{di})^2$. Similarly to [13] define

$$\Delta_i = K_{\theta_i} \cdot \frac{\partial \varphi_{\text{int}_i}}{\partial \theta_i} \cdot \left(\theta_{inh_i} - \theta_i\right) - K_{u_i} \cdot K_{z_i} \cdot \left| {}^i\nabla \varphi_{\text{int}_i} \cdot \eta_i \right|$$

and

$$\theta_{inh_i} = \arg\left(\frac{\partial \varphi_{int_i}}{\partial x_i} \cdot s_i + \mathrm{i} \cdot \frac{\partial \varphi_{int_i}}{\partial y_i} \cdot s_i \right)$$

Then for each aircraft in isolation we have the following:

Proposition 2: *Subsystem i under the control law*

$$\begin{aligned} u_i &= -K_{u_i} \cdot K_{z_i} \cdot \operatorname{sgn}\left({}^i\nabla \varphi_{\text{int}_i} \cdot \eta_i \right) \\ \omega_i &= K_{\theta_i} \cdot \left(\theta_{inh_i} - \theta_i\right), \;\; \Delta_i \le 0 \\ \omega_i &= -K_{\theta_i} \cdot \frac{\partial \varphi_{\text{int}_i}}{\partial \theta_i}, \;\; \Delta_i > 0 \end{aligned}$$

converges globally asymptotically to p_{di}.

The following lemma states that once an agent has entered its safe region, it will never leave:

Lemma 2: For *the subsystem i under the control law of Proposition 2, the set* $B_{\text{int}_i} = \left\{ p_i : \left\| q_i - q_{di} \right\| \le \varepsilon, \theta_i \in (-\pi, \pi] \right\}$ *is positive invariant.*

3.3 Simulation results

To demonstrate the navigation properties of our decentralized approach, we present three simulations of four nonholonomic agents that have to navigate from an initial to a final configuration, avoiding collisions. The chosen configurations constitute non-trivial setups since the straight-line paths connecting initial and final positions of each agent are obstructed by other agents. The following sequence of figures verifies the collision avoidance and global convergence properties of our algorithm. In each figure the circles denote the targets of each agent while the ring around each target represents the corresponding transition guard where the transition from the first to the second stage takes place.

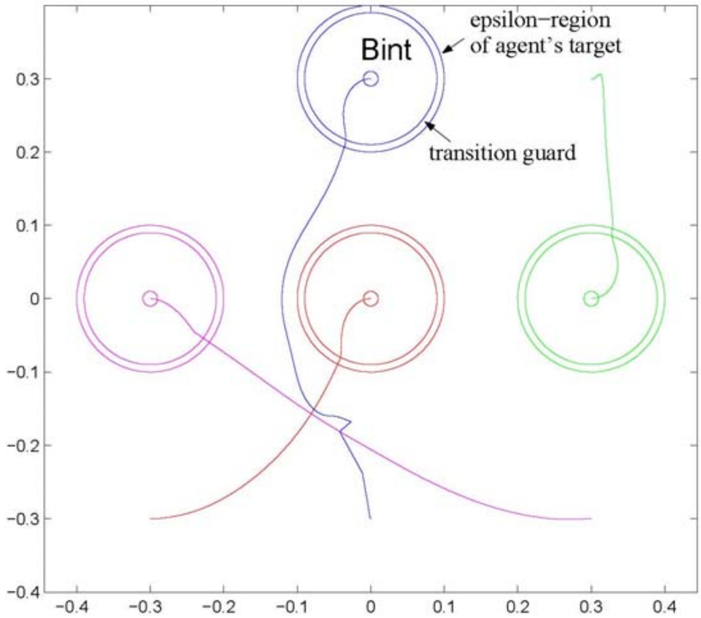
Simulation A

Initial Conditions

$p_1 = [-.3 \quad -.3 \quad 0]^T$ (red), $p_2 = [0 \quad -.3 \quad \pi/2]^T$ (blue), $p_3 = [.3 \quad -.3 \quad \pi]^T$ (pink), $p_4 = [.3 \quad .3 \quad \pi]^T$ (green)

Goal Conditions:

$p_{d1} = [0 \quad 0 \quad 0]^T$, $p_{d2} = [0 \quad .3 \quad 0]^T$, $p_{d3} = [-.3 \quad 0 \quad 0]^T$, $p_{d4} = [.3 \quad 0 \quad 0]^T$



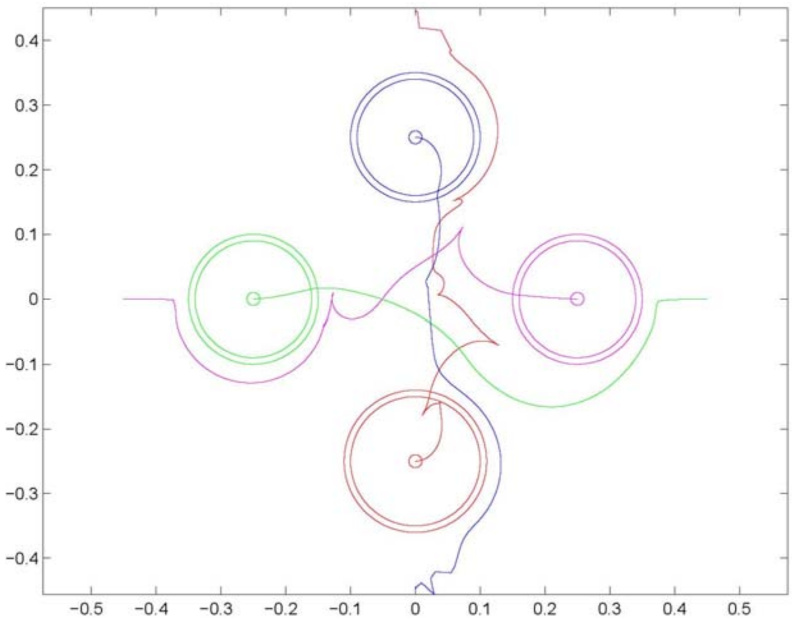
Simulation B

Initial Conditions

$p_1 = [0 \quad .5 \quad -\pi/2]^T$, $p_2 = [0 \quad -.5 \quad -\pi/2]^T$, $p_3 = [-.5 \quad 0 \quad 0]^T$, $p_4 = [.5 \quad 0 \quad \pi]^T$

Goal Conditions:

$p_{d1} = [0 \quad -.3 \quad \pi]^T$, $p_{d2} = [0 \quad .25 \quad -\pi/4]^T$, $p_{d3} = [.3 \quad 0 \quad 0]^T$, $p_{d4} = [.25 \quad 0 \quad \pi]^T$



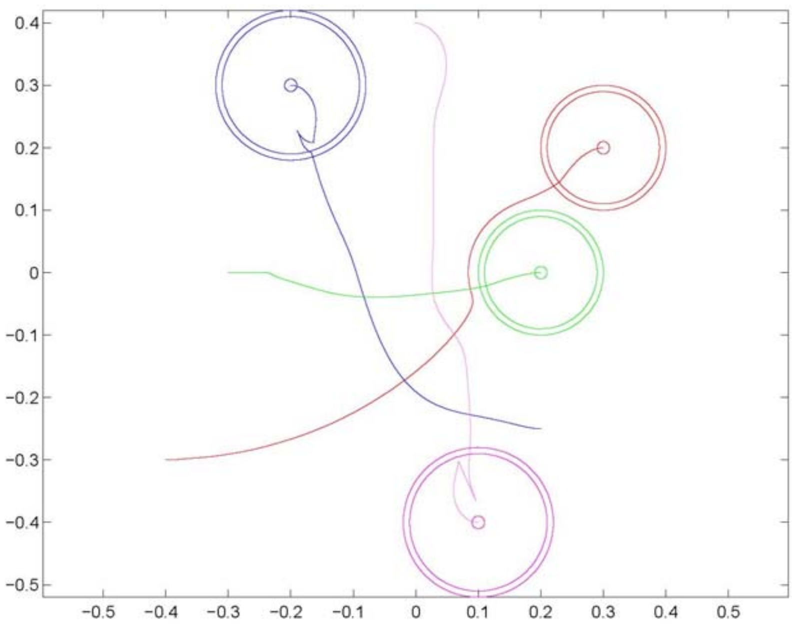
Simulation C

Initial Conditions

$$p_1 = [-4 \quad -.3 \quad 0]^T, p_2 = [.25 \quad -.3 \quad \pi]^T, p_3 = [0 \quad .4 \quad 0]^T, p_4 = [-.3 \quad 0 \quad 0]^T$$

Goal
Conditions:

$$p_{d1} = [.4 \quad .2 \quad 0]^T, p_{d2} = [-.1 \quad .3 \quad \pi]^T, p_{d3} = [.1 \quad -.4 \quad -\pi/3]^T, p_{d4} = [.3 \quad 0 \quad 0]^T$$



Chapter 4

Discussion and Current Research Issues

4.1 Discussion on Previous Results

In previous chapters we discussed methodologies developed in the first 18 months of WP6 in the Control Systems Lab of NTUA. We show how the powerful navigation functions tool established by Koditscheck and Rimon in [9], has successfully been extended to navigation and collision avoidance of multiple agents, in the centralized [12], [13] as well as in the decentralized case [3], [11]. We have enhanced the navigation functions of the holonomic case with a dipolar structure, so that the nonholonomic dynamics are taken care of. The analytic expression of the controller in all cases provides for fast feedback in real time. Simulation results have proven the power and effectiveness of our method.

Nevertheless, there are still some remaining issues that should be taken into account when aiming to apply the navigation function method in real world situations:

- In practice, the sensing capabilities of each aircraft are limited. Consequently, each aircraft cannot have knowledge of the positions and/or velocities of every aircraft in the workspace but only of the aircraft within its *sensing zone* at each time instant. A preliminary definition of the sensing zone of an aircraft could be a circle of constant radius around its centre of mass, in the vein of [1],[8].
- Although the motion of a robot can be adequately modelled in a deterministic vein, this is not the case for air traffic management systems. Sources of uncertainty arise either from the pilot's behaviour and/or the lack of exact knowledge of the neighbouring aircraft's motion in the decentralized case ([7],[16]). It is therefore necessary to add stochastic components to the existing model.
- Velocity bounds have so far only been treated in [3]. We have imposed a hybrid structure on the dynamics of each agent so that its velocity satisfies a pre-specified *upper bound.* The method can easily be extended to the nonholonomic case. However, it is currently under investigation how this method can be applied in order to include *lower bounds* in the velocities of the moving agents.

The following subsections provide highlights of current research under WP6 in order to solve these problems. The third issue, namely the handling of velocity bounds is left for a future deliverable.

4.2 Handling of Limited Sensing Capabilities

In practice, the sensing capabilities of each aircraft are limited within the area of its *sensing zone*. This, at a first approach, can be defined as a circle of constant radius around its centre of mass. In this case the control law of each agent depends on the state of the set of agents within its sensing zone at each time instant. It is obvious that the control scheme in this case incorporates both continuous (e.g. the aircraft's motion) and discrete (e.g. the entry/leave of an aircraft in another's sensing zone) dynamics. The overall scheme can be modelled as a deterministic hybrid system. The following figure, taken from [1], represents the hybrid structure in a three aircraft encounter. S_i denotes the information pattern (i.e. which aircraft are within the sensing zone of *i* at each time instant) of *i* at each different mode of operation I_i. The guards C_{ij} represent transitions between different modes of operation, i.e. *j* enters/leaves the protected zone of *i*.

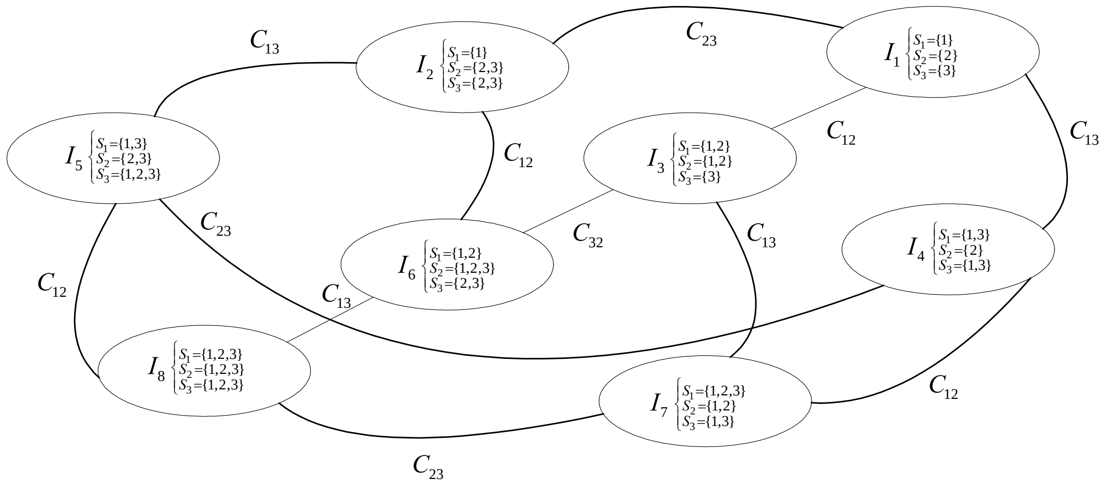


Fig. 4.1 Hybrid Automaton Model of the Decentralized ATM Process

In [1], the authors treat the problem using optimal control tools. Since in our case we are interested in closed-loop solutions, it is crucial to verify the stability of the overall hybrid system. There are two ways to treat such a problem: (i) use existing results on stability of switched/non-smooth systems [2], [17] to prove the convergence of our algorithm or (ii) treat the problem in a graph-theoretic perspective. The latter has already been used in [18] to prove the stability of flocking motion of multiple agents with limited sensing capabilities. We aim to extend this approach to the case of decentralized collision avoidance.

4.3 Dealing with Uncertainty

Uncertainty is an inevitable issue in air traffic systems. Several results computing the *Probability of Conflict* between two aircraft have appeared in literature in the past decade [10],[15]. A review of these methods has been presented in the first deliverable under WP6 [6, Chapter 3]. While most work on probabilistic models for CDR focuses on the estimation of prediction errors on the aircraft trajectory, very few of these provide specific routing instructions to the aircraft involved in the encounter. Work under HYBRIDGE WP5 provides centralized algorithms for stochastic conflict resolution. Decentralized algorithms for stochastic CDR have been presented in [7], [16].

It is our goal to treat the problem in a closed-loop fashion. The whole ATM process can now be modelled as a stochastic hybrid system. Each aircraft can only have knowledge of an estimate of the current positions of the other aircraft within its sensing zone. Whenever the sensing zone of aircraft *i* is empty the dynamics of the aircraft are purely deterministic, namely a navigation function driving *i* towards its destination. Whenever an aircraft enters the protected zone of aircraft *i*, the control law is switched in order to meet both specifications: destination convergence(DC) and collision avoidance(CA). Hence the switching control strategy is given by:

$$\dot{q}_i = \begin{cases} DC, \text{ if } N(i) = 0 \\ DC \wedge CA, \text{ if } N(i) \neq 0 \end{cases}$$

where q_i the configuration of i and $N(i)$ the number of aircraft in the protected zone of *i*. In this equation DC denotes the control imposed on i in order to meet the destination convergence goal whereas $DC \wedge CA$ denotes the control imposed on *i* in order to meet the destination convergence and collision avoidance goals simultaneously whenever there are intruding aircraft in *i*'s sensing zone.

Each agent treats the movement of the other agents as a stochastic differential equation. For example let the dynamics of aircraft *i* be given by $dq_i(t) = b_i(q(t))dt$ and the dynamics of an intruding aircraft *j* be given by the stochastic differential equation $dq_j(t) = b_j(q(t))dt + G_j(q(t))dB(t)$, where $q(t) = \begin{bmatrix} q_i(t) & q_j(t) \end{bmatrix}^T$. Then the relative position of aircraft *i* with respect to *j* is given by

$$dq_{ij}(t) = (b_i(q(t) - b_j(q(t)))dt - G_j(q(t))dB(t)$$

A possible interpretation of the two performance objectives *DC* and *CA* could then have the form:

$$DC: \text{ design } b_i(q) \text{ so that } P\left\{\sup_{t \geq o} \|q_i - q_{di}\| > \varepsilon\right\} \leq N(\varepsilon) > 0, \forall \varepsilon > 0$$

$$CA: \text{ design } b_i(q) \text{ so that } P\left\{\inf_{t \geq o} \|q_{ij}\| \leq d\right\} \leq M, M \text{ suff. small}$$

where q_{di} the desired destination of *i* and *d* the separation minimum between two aircraft.

It is obvious that such a system has a stochastic hybrid structure. Theorems for checking the (asymptotic) stability in probability of such a scheme are discussed in [4].

References

[1] A. Bicchi and L. Pallottino, “On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems”, *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221-232, 2000.

[2] M.S. Branicky, “Multiple Lyapunov Functions and other Analysis Tools for Switched and Hybrid Systems”, *IEEE Transactions on Automatic Control,* 43(4):475-482, 1998.

[3] D.V. Dimarogonas, M.M. Zavlanos, S.G. Loizou and K.J. Kyriakopoulos, “Decentralized Motion Control of Multiple Holonomic Agents under Input Constraints”, *Proceedings of the 42nd IEEE Conference on Decision and Control,* Maui, Hawaii, 2003.

[4] D.V. Dimarogonas, M.M. Zavlanos, S.G. Loizou and K.J. Kyriakopoulos, “Decentralized Motion Control of Multiple Holonomic Agents under Input Constraints”, Technical Report, NTUA, 2003.

[5] D.V. Dimarogonas and K.J. Kyriakopoulos, “Lyapunov-like Stability of Switched Stochastic Systems”, *submitted to the 2004 American Control Conference.*

[6] D.V. Dimarogonas and K.J. Kyriakopoulos, “Inventory of Decentralized Conflict Detection and Resolution Systems in Air Traffic”, Technical Report HYBRIDGE, WP6, NTUA, Deliverable D6.1.

[7] J. Hu, M. Prandini and S. Sastry , "Aircraft conflict detection in presence of spatially correlated wind perturbations", *submitted to Journal of Guidance, Control and Dynamics,* Jan. 2003.

[8] G. Inalhan, D.M. Stipanovic and C.J. Tomlin, “Decentralized Optimization, with Application to Multiple Aircraft Coordination”, *Proceedings of the 41st Conference on Decision and Control*, Las Vegas, 2002.

[9] D.E. Koditscheck and E. Rimon, “Robot Navigation Functions on Manifolds with Boundary”, *Advances Appl. Math.*, 11:412-442, 1990.

[10] J. Krozel and M. Peters, “Strategic Conflict Detection and Resolution for Free Flight”, *Proceedings of the 36th Conference on Decision and Control*, San Diego, 1997.

[11] S.G. Loizou, D.V. Dimarogonas and K.J. Kyriakopoulos, “Decentralized Feedback Stabilization of Multiple Nonholonomic Agents”, *submitted to the 2004 IEEE International Conference on Robotics and Automation.*

[12] S.G. Loizou and K.J. Kyriakopoulos, “Closed Loop Navigation for Multiple Holonomic Vehicles”, *Proceedings of the IEEE Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002.

[13] S.G. Loizou and K.J. Kyriakopoulos, "Closed Loop Navigation for Multiple Non-Holonomic Vehicles", *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[14] J.Lygeros and M. Prandini, "Aircraft and Weather Models for Probabilistic Collision Avoidance in Air Traffic Control" , *Proceedings of the 41st Conference on Decision and Control*, Las Vegas, 2002.

[15] R.A. Paielli, H. Erzberger, "Conflict Probability Estimation for Free Flight", *Journal of Guidance, Control and Dynamics*, 20(3):588-596, 1997.

[16] M. Prandini, J.Hu, J. Lygeros and S.S. Sastry, "A Probabilistic Approach to Aircraft Conflict Detection", *IEEE Transactions on Intelligent Transportation Systems*, 1(4):199-220, 2000.

[17] D. Shevitz and B. Paden, "Lyapunov Stability Theory for Nonsmooth Systems", *IEEE Transactions on Automatic Control,* 39(9):1910-1914, 1994.

[18] H.G. Tanner, A. Jadbabaie and G.J. Pappas, "Stable Flocking of Mobile Agents, Part II: Dynamic Topology", *Proceedings of the 42nd IEEE Conference on Decision and Control,* Maui, Hawaii, 2003.

[19] H.G. Tanner, S.G. Loizou and K.J. Kyriakopoulos, "Nonholonomic Navigation and Control of Cooperating Mobile Manipulators", *IEEE Transactions on Robotics and Automation*, 19(1):53-64 , 2003.

[20] H.G. Tanner and K.J. Kyriakopoulos, "Nonholonomic Motion Planning for Mobile Manipulators", *Proceedings of the IEEE International Conference on Robotics and Automation,* San Francisco, CA, 2000.