Decentralized Motion Control of Multiple Holonomic Agents under Input Constraints

D. V. Dimarogonas, M. M. Zavlanos, S. G. Loizou and K. J. Kyriakopoulos

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

Abstract: ***The navigation function methodology, established in previous work for centralized multiple robot navigation, is extended for decentralized navigation with input constraints. In contrast to the centralized case, each agent plans its actions without knowing the destinations of the other agents. Asymptotic stability is guaranteed by the existence of a global Lyapunov function for the whole system, which is actually the sum of the separate navigation functions. The collision avoidance and global convergence properties as well as input requirements are verified through simulations.***

1. INTRODUCTION

Navigation of mobile agents has been an area of significant interest in robotics. Most efforts have focused at the case of single agent navigating in an environment with obstacles [1]. Recently, navigation for multiple agents has gained increasing attention. The basic motivation for this work comes from two application domains: (i) decentralized conflict resolution in air traffic management and (ii) the field of micro robotics, where a team of autonomous micro robots must cooperate to achieve manipulation precision in the sub micron level. In both cases, velocity constraints could be a crucial issue that should be dealt with directly.

Whenever multiple mobile agents share the same workspace, the potential for collisions among them must be taken into account. This can be done by either using a centralized approach to plan collision free trajectories for all [2] or by independently planning trajectories, in a decentralized manner. Lately, several ways for decentralized motion planning of multiple agents have been proposed. The hybrid control architecture in combination with parallel problem solving proposed in [3], guarantees collision avoidance, while in [4], the authors use decentralized optimization techniques to obtain optimal conflict-free paths in a multiple aircraft system. The authors in [5], use the sense of "Formations of Robots" where each robot has its own coordinate system to control its relative positions. Asymptotic stability is guaranteed based on Lyapunov's second method.

While centralized approaches have the disadvantage of being computationally demanding, inflexible and presupposing the existence of a global communication network, decentralized approaches presuppose inter-agent communication and sensory information that could be very demanding for the agent's onboard equipment. For example, in micro robotics, because of size constraints, such demands could possibly prove infeasible. The problem of input bounds is another important matter that should be taken into account, especially in the air traffic management application, where issues such as passenger comfort and fuel consumption are of major importance.

Taking those aspects into consideration, the multi agent navigation problem treated in this paper can be stated as follows: "*Derive a set of control laws (one for each agent) that drives a team of agents from any initial configuration to a desired goal configuration avoiding, at the same time, collisions and fulfilling pre-specified input constraints. The environment is assumed perfectly known and stationary, while each agent has global knowledge of it and the team configuration*". Our basic idea is to use the gradient of a potential function for each agent to navigate the whole team, while each agent acts as a potential obstacle to the others. We use stability results from the hybrid systems domain to guarantee fulfillment of the input constraints.

The rest of the paper is organized as follows: Section 2 outlines the concept of navigation functions and describes the idea of decentralized motion planning. Section 3 introduces the terminology and mathematical tools required for the analysis. Section 4 deals with the constrained input problem. Section 5 describes a method to overcome the difficulties arising when an agent is very close to its desired destination. Section 6 presents simulation results for a number of non-trivial multi agent navigational tasks. Finally, section 7 summarizes the conclusions and indicates our current research.

2. 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 [6,7]. Our assumption about spherical agents and obstacles does not constrain the generality of this work since it has been proven that navigation properties are invariant under diffeomorphisms.

Consider a system of n agents operating in the same workspace $W \subset R^2$. Each agent i occupies a disk: $R = \left\{ q \in R^2 : \left\| q - q_i \right\| \le r_i \right\}$ in the workspace where $q_i \in R^2$ is the center of the disk and r_i is the radius of the agent. The configuration space is spanned by $q = \left[q_1, \dots, q_n \right]^T$. A navigation function can be defined as follows:

Definition 1: *Let $F \subset R^n$ be a compact connected analytic manifold with boundary. A map $\varphi : F \to [0,1]$ is a navigation function if: (1) It is analytic on F, (2) It has only one minimum at $q_d \in \overset{o}{F}$, (3) Its Hessian at all critical points (zero gradient vector field) is full rank, and (4) $\lim_{q \to \partial F} \varphi(q) = 1$.*

In the centralized setup of [2], a central authority has knowledge of the current positions and desired destinations of all agents and the sought control law is of the form: $u = -K \bullet \nabla \varphi(q)$ where K is a gain. In the decentralized case encountered in this work, each agent has knowledge only of the current positions of the others, and not of their desired destinations. Hence each agent has a different navigation law.

We consider the following class of decentralized navigation functions: $\varphi_i = \sigma_d \circ \sigma \circ \hat{\varphi}_i = \left(\frac{\gamma_i}{\gamma_i + G_i} \right)^{1/k}$, which is a composition of $\sigma_d = x^{1/k}$, $\sigma = \frac{x}{1+x}$, and the cost function: $\hat{\varphi}_i = \frac{\gamma_i}{G_i}$, where $\gamma_i^{-1}(0)$ denotes the desirable set (i.e. the goal configuration) and $G_i^{-1}(0)$, the set that we want to avoid (i.e. collisions with the other agents). A suitable choice is: $\gamma_i = \left(\gamma_d^i\right)^k$, where: $\gamma_d^i = \|q_i - q_{di}\|^2$, 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*, wrt that proposed in [2] is $\varphi_i(q) = \gamma_d^i / \left((\gamma_d^i)^k + G_i\right)^{1/k}$ and the corresponding agent control law is $\dot{q}_i = -K_i \cdot \frac{\partial \varphi_i}{\partial q_i}$. The following theorem will help us on deriving results for the function φ_i by examining the simpler function $\hat{\varphi}_i$:

Theorem 1 [7]: *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 $q = \left[q_{d1}, ..., q_{dn}\right]^T$. 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.*

The whole proof of this statement, as well as the proofs of propositions 2-5, is provided in [8]. This set of measure zero corresponds to the points $\|\partial \varphi_i / \partial q_i\| = 0 \; \forall i$. We use the result of theorem 1, to show that in such a situation, unless an agent has reached its destination point, there is always a direction of movement decreasing its potential function. Similar to the results in [2], 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 the obstacles.*

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 the obstacles.*

In the decentralized setup, the sense of the term "critical point" is slightly different than that of the centralized case [2]. The set of critical points of φ_i is defined as $C_{\varphi i} = \{q : \partial \varphi_i / \partial q_i = 0\}$. A critical point is *non-degenerate* if $\partial^2 \varphi_i / \partial^2 q_i$ has full rank at that point.

An important feature that should be noticed is the fact that once an agent is very close to its destination, its cost function assumes very small values, and hence there is a small potential of leaving in a case of a possible collision. Hence, the overall scheme becomes less robust in such a situation. To avoid such a situation we make the following assumption:

Assumption 1: *Each agent disappears as a state of the system once it is sufficiently close to its destination, i.e. once $\gamma_d^i \le \delta$, where δ a sufficiently small positive parameter.*

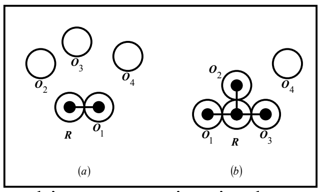
3. THE DECENTRALIZED CONTROL METHODOLOGY

3.1. "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 $\tilde{q}$ 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 [2] 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 and the obstacles with O_j.



Considering n objects operating in the same workspace (one agent and *n–1* obstacles), the agent, in order to reach its goal configuration, has to avoid collision with the other *n–1* obstacles. The number of all the possible *level-1* relations that could occur is the combination of the *n–1*obstacles by 1, i.e. $s_1 = \binom{n-1}{1}$. Respectively, the number of all possible *level-2* relations is given by the combinations: $s_2 = \binom{n-1}{2}$. Thus, the number of all the possible relations of all possible levels is given by the sum: $s = \sum_{i=1}^{n-1} s_i = \sum_{i=1}^{n-1} \binom{n-1}{i}$. It is obvious that the maximum number of *levels* that we could have for n objects operating in the workspace is *n–1*.

We define the (always nonempty) set of integers S_l including all possible relations in *level-l*, by: $S_l = \{ j \in Z : 0 < j \le s_l \}$. Obviously, the set of all possible relations of all possible levels is: $S = \{ j \in Z : 0 < j \le s \}$. We define by: $(R_j)_l$ the j relation of *level-l*, where $j \in S_l$ as defined above. For example, in figure *(1b)*: $(R_1)_3 = \{\{R, O_1\}, \{R, O_2\}, \{R, O_3\}\}$, where we have set arbitrarily *j=1*. In the same way, we define its complementary set by: $(R_j^C)_l = \{ i \in S_l, i \ne j : \{(R_i)_l\} \}$.

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 as it has been defined above. Obviously, i indicates a relation of *level–l*.

A *"Relation Verification Function" (RVF)* is defined by: $(g_{R_j})_l = (b_{R_j})_l + \lambda (b_{R_j})_l \Big/ ((b_{R_j})_l + (B_{R_j^C})_l^{1/h})$, for $l \le n-2$, 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_l^C indicates a complementary set of relations of *level–l*. Using the simplified notation introduced above, the relation verification function can be rewritten as: $g_i(b_i, \tilde{b}_i) = b_i + \lambda b_i \Big/ b_i + \tilde{b}_i^{1/h}$, for $l \le n-2$ 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. $l \le n-2$ and l=n-1) since for l=n-1, $j \in S_{n-1} = \{1\}$ and so the set $(R_1^C)_{n-1}$ is an empty set. Thus we can't define $(B_{R_1^C})_{n-1}$.

We could compute the following limits of *RVF* (using the simplified notation): when $b_i \to 0$ and $\tilde{b}_i \ne 0$, 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: $b_i \ne 0$, independently of how $\tilde{b}_i$ behaves: $g_i \ne 0$. 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*. This equation indicates that G is practically the product of a certain number of g_is.

3.2. Proof of Correctness

For a detailed proof of Propositions 1-5, the reader is referred to [8]. 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 [2]. 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 the obstacles*: $F_0(\varepsilon) = \bigcup_{l=1}^{n_L}\bigcup_{j=1}^{n_{R,l}} B_j^l(\varepsilon) - \{q_d\}$
4. The set *away from the obstacles*: $F_1(\varepsilon) = F - (\{q_d\}\bigcup \partial F \bigcup F_0(\varepsilon))$

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.

4. VELOCITY BOUNDS

Let us now return to the decentralized motion control problem. The following control law has been derived:

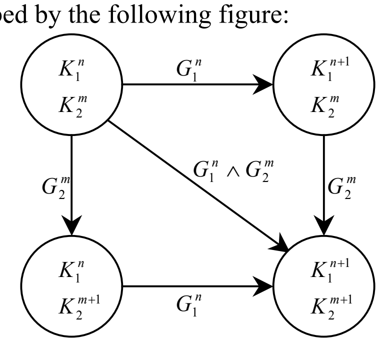
$$\dot{q}_1 = -K_1 \cdot \partial\varphi_1 / \partial q_1 \quad \ldots \quad \dot{q}_n = -K_n \cdot \partial\varphi_n / \partial q_n$$

where φ_i is a navigation function for each subsystem. Suppose that each agent must satisfy a velocity constraint: $K_i \left\| \partial\varphi_i / \partial q_i \right\| \le U_i$. It would be preferable not to change the navigation function φ_i to meet the input constraints so that the powerful convergence properties established earlier are fulfilled. Thus, the idea is to change the gain of each agent whenever its input constraints threaten to be violated. We consider multiple discrete states for each agent, to which the agent dynamics are switched whenever the previous constraints tend to be violated. The switching is state dependent and the dynamics of agent *i* are given by $\dot{q}_i = -K_i^j \partial\varphi_i / \partial q_i$, if $K_i^j \left\|\partial\varphi_i / \partial q_i\right\|_{t_j} \le U_i$, for $t \ge t_j$ and $\dot{q}_i = -K_i^{j+1} \partial\varphi_i / \partial q_i$, if $K_i^j \left\|\partial\varphi_i / \partial q_i\right\|_{t_{j+1}} \le U_i$ where $j \in N$, K_i^j denotes a "beginning" gain of agent *i* (the initial gain can be taken arbitrarily) and K_i^{j+1} its new gain when the input constraints become violated for the *(j+1)-th* time. Obviously $K_i^j > K_i^{j+1}$. Continuity of the state is assumed whenever a switching occurs.

In order to investigate the properties of this control law we apply existing stability analysis results for Hybrid Systems to the problem of existence of input constraints in the navigation function methodology. Specifically, we make use of the stability results in [9], [10] in order to guarantee that our proposed hybrid model will maintain the convergence properties of the decentralized algorithm. Consider the following hybrid system model [9]: $\dot{x}(t) = f(x(t), i(t))$ where $x \in R^n$ is the state space and $i(t)$ is the switching signal taking values in a finite set of indices $\{1,\ldots,M\}$. We assume that there are only a finite number of switches per unit time. We also assume continuity of the state at each switching instant. The hybrid dynamics define the switching sequence: $S = x_0; (i_0, t_0), (i_1, t_1), \ldots$ where x_0 is the initial condition and the notation (i_j, t_j) means that the state evolves according to $\dot{x} = f(x, i_j)$ for $t_j \le t < t_{j+1}$. A *Lyapunov-like* function $V_i : R^n \to R$ for the system $\dot{x} = f(x, i)$ with equilibrium $x_{eq} = 0$ is a positive definite function with negative semi-definite derivative whenever system *i* is active. The theorem from [9] provides sufficient conditions for stability of the overall hybrid system. Assuming the existence of a Lyapunov-like function for each system, the theorem simply indicates that stability in the sense of Lyapunov is guaranteed provided that the energy of each system does not increase between consecutive active intervals for any switching sequence. In fact, as remarked in [10], the origin is asymptotically stable provided that there are infinite switches and the *Lyapunov-like* functions are strictly decreasing between consecutive time intervals. Asymptotic stability is also guaranteed when the Lyapunov-like function of each system is strictly decreasing whenever it is activated. In the case that all the systems admit a common Lyapunov-like function the following corollary is straightforward:

Corollary 1: *Assume that for the hybrid system under consideration there is a common Lyapunov-like function for each of the subsystems, which is strictly decreasing for every switching sequence S. Then the origin is asymptotically stable.*

Lets return now to our problem and see whether the control law satisfies *Corollary 1*. We also assume that, whenever an agent switches to another gain's discrete state, it does not return to the previous state even if the condition: $K_i^j \left\|\partial\varphi_i / \partial q_i\right\|_{t_j} \le U_i$, generally, holds for $t > t_j$, where *j* denotes the last switching that took place. Hence, for two agents, the hybrid dynamics between two discrete states are described by the following figure:



where G_1^n and G_2^m denote the violation of the input constraints on agents 1 and 2 for the *n-th* and *m-th* time respectively. Of course, the former scheme can be generalized for *n>2* agents. It is obvious that the prescribed hybrid system satisfies Corollary 1, with common *Lyapunov*

Function $\varphi = \sum_{i=1}^{n} \varphi_i$. Hence, the convergence properties of the navigation functions are not violated.

What remains now is to propose a method of computing a sequence of gains for agent i. The proposed choice for the *j-th* gain is: $K_i^j = U_i / N_j$ where $N_j = \left\| \partial \varphi_i / \partial q_i \right\|_{t_j}$ is the norm taken at the time when the *j-th* switching occurs. Obviously, this switching is a direct consequence of the violation of the constraint: $K_i^{j-1} \left\| \partial \varphi_i / \partial q_i \right\|_{t_j} \leq U_i$. Hence, for each agent we can compute in real time a gain vector $K_i = \begin{bmatrix} K_i^0 & K_i^1 & \dots & K_i^j & \dots \end{bmatrix}^T$. With such a choice of the gain, the input constraints would always be fulfilled.

5. 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 and lacks in robustness in the case discussed at the end of section 2. This is because in these cases the numerator of φ_i is very small (zero when an agent has reached its destination) so the potential for an agent to move is negligible in a possible collision scheme (see Assumption 1). 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 q_{di}. The navigation function in this case becomes $\varphi_i(q) = (\gamma_d^i + f(G_i)) / \left((\gamma_d^i + f(G_i))^k + G_i \right)^{1/k}$. A suitable function is $f(G) = a_0 + \sum_{j=1}^{3} a_j G^j$ for $0 \leq G \leq X$ and $f(G) = 0$ for *G>X*. This function satisfies $f(0) = Y, f^{'}(0) = 0, f^{"}(0) < 0$ (local maximum at *G*=0) and $f(X) = 0, f^{'}(X) = 0, f^{"}(X) > 0$ (local maximum at *G*=*X*), where $X, Y > 0$. The coefficients a_i are evaluated in order to fulfill 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 fulfill definition 1. It is our current goal to extend the theory established in [7] from analytic to merely differentiable functions.

6. 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.

Initial Configurations: q_1 = [.1732, -.1] ^{T}, $q2$ = [-.1732, -.1]^T ,q_3 = [0, .2]^T ,q_4 = [0, 0]^T.

Goal Configurations: q_{d1} = [-.1732, .1] ^{T}, $qd2$ = [.1732, .1]^T ,q_{d3} = [0, -.2]^T ,q_{d4} = [0, 0]^T.

Parameters: X_1 = .2308, X_2 = . 2308, X_3 = . 2308, X_4 =.0024, Y = .1, k=100.

Velocity Bounds: U_1=U_2=2e-4, U_3=3e-4, U_4=1e-4

In the following sequence of figures one can see the paths followed under the proposed decentralized scheme during 20000 time units (t.u.). Fig. B shows the agent motion between 1 and 1000 t.u., fig. C between 1000 and 8000 t.u., fig. D between 8000 and 12000 t.u. and fig. E between 12000 and 20000 t.u. A-i and T-i denote agent i and its desired destination respectively.

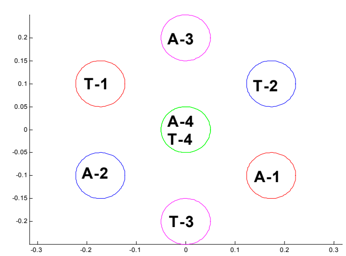


Fig.A

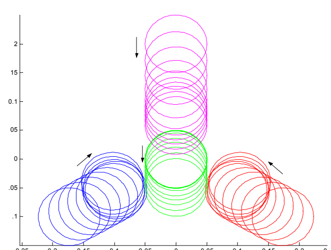


Fig.B

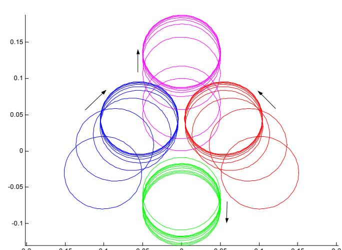


Fig.C

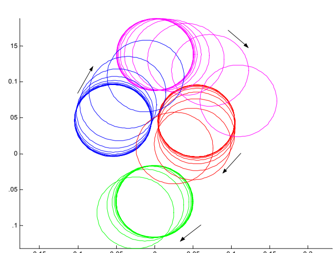


Fig.D

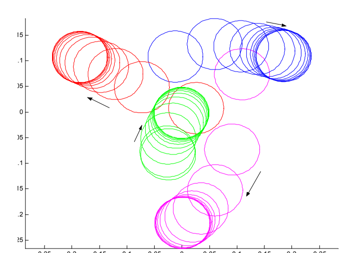


Fig.E

Furthermore, the following diagrams represent agent velocities throughout the encounter. Fig. F shows the velocities between 1 and 4000 t.u. and fig. G between 4000 and 20000 t.u.

The following remarks are in order: the input constraints of all agents and the navigation properties are fulfilled throughout the encounter. The first is clarified in figures A through E and the latter in the velocity-time diagrams. The initial velocity (time=0) of agents 1,2 and 3 is their maximum allowed velocity, since the method forbids higher ones. In figure F, the velocities of agents 1 and 2 (blue and red lines) coincide due to their symmetrical initial positions and the fact that they have the same maximum speed. When the velocity of agent 4 (green line) tends to reach its upper bound (time=500÷1000) -which occurs when the faster agent 3 captures agent 4 from behind-, the proposed methodology successfully prevents a violation of the constraints. At time 12000 (beginning of motion at fig. E) all agents apart from

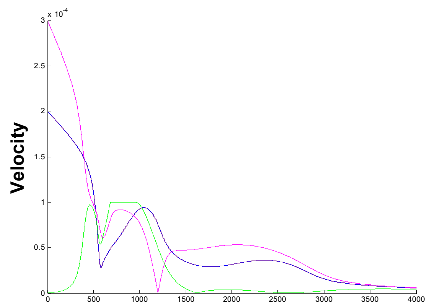


Fig.F

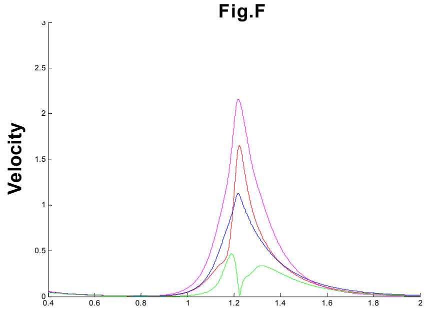


Fig.G

agent 4 reach a local maximum of their velocities (c.f. figure G). That is when the first three agents have a 'clear' path towards their destinations. Agent 4 has to wait until agent 1 provides it a 'clear' path to its destination and that happens approximately at time 1350 (c.f. figure G). The velocities of each agent decrease as they approach their destinations without obstacles.

7. CONCLUSIONS-ISSUES FOR FURTHER RESEARCH

In this paper, a methodology for multiple mobile agent navigation is presented. The methodology extends the centralized agent navigation established in [2] to a decentralized approach to the problem under input constraints. As in [2], the agent – obstacle potentials are formed by appropriately constructed agent proximity potentials, which capture all the possible multi agent proximity situations. The great advantage of the method is its relatively low complexity wrt the number n of agents, compared to centralized approaches to the problem and the application of velocity bounds. The number M of *RVF*'s for a group of n agents is given by: $M = n \cdot \sum_{i=1}^{n-1} \binom{n-1}{i}$. Thus, for $n=5$ agents we would have to compute: $M=75$ *RVF*'s, for $n=6$: $M=186$, for $n=7$: $M=441$ etc. The effectiveness of the methodology is verified through computer simulations. Current research directions are towards applying the methodology to the cases where each agent has knowledge of the velocities of the others and where there is some form of uncertainty in the agent movement.

8.ACNOWLEDGEMENTS

The authors want to acknowledge the contribution of the European Commission through contracts HYBRIDGE (IST-2001-32460) project and MICRON (IST-2001-33567).

9.REFERENCES

[1] J.C. Latombe, "Robot Motion Planning", Kluwer Academic Publishers, 1991

[2] S.G. Loizou and K.J. Kyriakopoulos, "Closed Loop Navigation for Multiple Holonomic Vehicles", Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 2861 – 2866, 2002

[3] K. Azarm and G. Schmidt, "Conflict-Free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation", IEEE Intern. Conf. on Rob. and Automation, pp 3526-3533, 1997

[4] G. Inalhan, D.M. Stipanovic and C.J. Tomlin, "Decentralized Optimization, with Application to Multiple Aircraft Coordination", in Proc. 41st IEEE Conf. Decision and Control, Kobe, Las Vegas, 2002

[5] H. Yamaguchi and J.W. Burdick, "Asymptotic Stabilization of Nonholonomic Mobile Robots Forming Group Formations", 1998 IEEE Intern. Conf. on Rob. and Autom., pp 3573-3580

[6] E. Rimon and D.E. Koditscheck, "Exact Robot Navigation Using Artificial Potential Functions", IEEE Trans. on Robotics and Automation, vol. 8, No. 5, pp 501-518, 1992

[7] D.E. Koditscheck and E. Rimon, "Robot Navigation Functions on Manifolds with Boundary", Advances Appl. Math., vol. 11, pp 412-442, 1990

[8] D. V. Dimarogonas, M. M. Zavlanos, S. G. Loizou and K. J. Kyriakopoulos, "Decentralized Motion Control of Multiple Holonomic Agents under Input Constraints", Tech.Rep., NTUA, http://users.ntua.gr/sloizou/-academics/TechReports/TR0301.pdf, 2003.

[9] M.S. Branicky, "Multiple Lyapunov Functions and other Analysis Tools for Switched and Hybrid Systems", IEEE Trans. Automatic Control, vol. 43, No. 4, pp 475-482, 1998

[10] S. Pettersson and B. Lennartson, "Stability and Robustness for Hybrid Systems", in Proc. 35th IEEE Conf. Decision and Control, Kobe, Japan, 1996, pp 1202-1207