Decentralized Motion Control of Multiple Mobile Agents†

Michael M. Zavlanos, Kostas J. Kyriakopoulos

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

Abstract: *Our navigation function methodology, established for centralized multiple robot navigation, is extended for decentralized navigation. The notion of interconnected systems is used to model the whole system, and asymptotic stability is guaranteed. The collision avoidance and global convergence properties 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.

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. A hybrid control architecture in combination with parallel problem solving, proposed by K. Azarm and G. Schmidt [3], guarantees collision avoidance, while L. Chun, Z. Zheng and W. Chang [4] divide the problem of path planning into global and local path planning and use AI techniques in combination with real-time techniques to realize their idea. J.P. Desai, J. Ostrowski and V. Kumar [5] propose the use of "Formations of Robots" where a motion plan for the overall formation is used to control a single "lead" robot and the "followers" are governed by local control laws, by sensing their relative position to their neighboring robots. H. Yamaguchi and J.W. Burdick [6], also 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. Similarly, M. Jäger and B. Nebel [7] use distributed algorithms to achieve global coordination and assume local communication only between pairs of physically close robots.

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. Taking those aspects into consideration, as well as the fact that feedback control provides the means to prove stability, 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. 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.

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 new terminology and mathematical tools required for the analysis. Section 4 presents simulation results for a number of non-trivial multi agent navigational tasks. Finally, section 5 summarizes the conclusions and indicates our current research.

2. Navigation Functions – Decentralized Approach to the Problem

Our work will be realized in two steps:

Step 1: *We derive a control law, driving a single agent from any initial configuration to a desired goal configuration, in an environment with moving obstacles, avoiding, at the same time, any collisions.*

Step 2: *We apply the previous results to a team of n agents and show that the method is globally*

---

† Research supported by the European Commission through projects HYBRIDGE (IST-2001-32460) project and IST-2001-33567-MICRON

asymptotically stable (all agents will eventually reach their goal configuration, avoiding collisions with each other). The main idea is that each agent treats the remaining n–1 agents as moving obstacles in the same workspace. Thus we have a decentralized approach to the problem of multiple agent navigation, which is the main purpose of this work.

2.1. Step 1

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 [8,9]. 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 objects (1 agent and $n-1$ moving obstacles) operating in the same workspace $W \subset R^2$. Agent R occupies a disk: $R = \{q \in R^2 : \|q - q_R\| \le r_R\}$ in the workspace where $q_R \in R^2$ is the center of the disk and r_R is the radius of the agent. Respectively, each obstacle O occupies a disk: $O_i = \{q \in R^2 : \|q - q_{Oi}\| \le r_{Oi}\}$ where $q_{Oi} \in R^2$ is the center of the disk and r_{Oi} is its radius. The configurations of the agent is q_r and that of the obstacles are $\boldsymbol{q_o} = [q_{o1} \quad q_{o2} \quad \dots \quad q_{o(n-1)}]^T$, where q_{oi} denotes the configuration of the i-*th* obstacle. The configuration space is spanned by $\boldsymbol{q} = [q_r \quad \boldsymbol{q_o}]^T$. 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 $j : F \to [0,1]$ 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) $\lim_{q \to \partial F} j(q) = 1$.*

If $\dot{q} = u$, the sought control law will be of the form: $u = -K \cdot \dfrac{\partial j(q)}{\partial q}$ where K is a gain. We will prove that the class of navigation functions introduced in [9] for single agent navigation in an environment with stable obstacles, if properly extended, can be used to navigate a single agent in an environment with moving obstacles. In this case, two special features should be taken into consideration:

- *A collision of the agent with more than one obstacles could occur, since they are moving and so the configuration of the space changes. Such a situation is not acceptable.*
- *Even if the agent has already reached its goal configuration, it should be able to move to avoid collision with an obstacle whose trajectory passes through the agent's goal configuration.*

To accomplish these objectives we consider the class of navigation functions: $j = s_d \circ s \circ \hat{j} = \left(\dfrac{g}{g+G}\right)^{1/k}$, which is a composition of $s_d = x^{1/k}$, $s = \dfrac{x}{1+x}$, and the cost function: $\hat{j} = \dfrac{g}{G}$, where $g^{-1}(0)$ denotes the desirable set (i.e. the goal configuration) and $G^{-1}(0)$, the set that we want to avoid (i.e. collisions with the obstacles). A suitable choice is: $g = (g_d + f(G))^k$, where: $g_d = \|q_r - q_{rd}\|^2$, is the squared metric of the current agent's configuration q_r from its destination q_{rd}, and $f(G)$ is a function, which practically constitutes a *perturbation* to the system, used to enable an agent (stable at its goal configuration) avoid an obstacle passing through this configuration. Function G has as arguments the coordinates of all obstacles and the agent, i.e. $G(q_r, \boldsymbol{q_o})$, in order to express all possible collisions of the agent with a number of obstacles, greater-equal to one. Thus, the navigation function proposed in the particular situation, wrt that proposed in [9], would be:

$$j(q_r, \boldsymbol{q_o}) = \frac{g_d(q_r) + f(G(q_r, \boldsymbol{q_o}))}{\left((g_d(q_r) + f(G(q_r, \boldsymbol{q_o})))^k + G(q_r, \boldsymbol{q_o})\right)^{1/k}}$$

and, as mentioned before, the agent control law is:

$$\dot{q}_r = -K \cdot \frac{\partial j(q_r, \boldsymbol{q_o})}{\partial q_r}$$

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

Theorem 1 [9]: *Let $I_1, I_2 \subseteq R$ be intervals, $\hat{j} : F \to I_1$ and $s : I_1 \to I_2$ be analytic. Define the composition $j : F \to I_2$ to be $j = s \circ \hat{j}$. If s is monotonically increasing on I_1, then the set of critical points of $\hat{j}$ and j coincide and the (Morse) index of each critical point is identical.*

Using the result of *theorem 1*, in section 3.3. (*proof of correctness*), we prove for $j(q_r, \boldsymbol{q_o})$ the following four propositions:

Proposition 1: *If the workspace is valid, the destination point q_{rd} is a non–degenerate local minimum of j.*

Proposition 2: *If the workspace is valid, all critical points of* j *are in the interior of the free space.*

Proposition 3: *For every e>0, there exists a positive integer N(e) such that if k>N(e) then there are no critical points of* $\hat{j}$ *in* $F_1(e)$*, where* $F_1(e)$ *denotes the set away from the obstacles.*

Proposition 4: *For any valid workspace, there exists an* $e_0 > 0$ *such that* $\hat{j}$ *has no local minimum in* $F_0(e)$*, as long as* $e < e_0$*, where* $F_0(e)$ *denotes the set near the obstacles.*

These propositions establish that goal configurations are achievable (not any collisions at the target) and that there will always be a direction of movement decreasing the potential function. This, according to *definition 1*, means that $j(q_r, \boldsymbol{q_o})$ is a navigation function, or, equivalently, a *Lyapunov function* for our system and, thus, the following properties hold:

a. $j(q_{rd}, \boldsymbol{q_o}) = 0$

b. $j(q_r, \boldsymbol{q_o}) > 0, \quad \text{for}: \quad q_r \neq q_{rd}$

c. $\dot{j}(q_{rd}, \boldsymbol{q_o}) = 0$

d. $\dot{j}(q_r, \boldsymbol{q_o}) \overset{\text{"almost"}}{<} 0, \quad \text{for}: \quad q_r \neq q_{rd}$

In property (d) we indicate by “almost” the fact 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 [8,9]. Hence, attempting to express mathematically this ascertainment, inequality (d) is not really a strict inequality due to the saddles. But since saddles are unstable equilibrium points, they practically do not affect the asymptotic stability of the system since a slight perturbation is enough to make the system diverge from a saddle point.

2.2. Step 2

Since, by definition, the *i-th* agent treats the remaining *n – 1* as obstacles, its control law, according to the previous analysis, would be: $\dot{q}_{ri} = -K_i \cdot \frac{\partial j_i(\boldsymbol{q_r})}{\partial q_{ri}}$, where: $\boldsymbol{q_r} = [q_{r1} \quad q_{r2} \quad \dots \quad q_{rn}]^T$ are the configurations of all agents. The control law of all *n* agents would be the system of equations of the previous form, for *i=1,…,n*:

$$\dot{q}_{r1} = -K_1 \cdot \frac{\partial j_1(\boldsymbol{q_r})}{\partial q_{r1}} \quad \cdots \quad \dot{q}_{rn} = -K_n \cdot \frac{\partial j_n(\boldsymbol{q_r})}{\partial q_{rn}}$$

It is obvious that the system is decentralized since each agent has knowledge only of its own goal configuration. Moreover, it is an interconnected nonlinear system [10], where the interconnection terms of the *i-th* equation are the configurations q_{rj} of the other agents *(j≠i)*. What is particular of this system is that the control law of the *i-th* agent, considered alone in the workspace, is asymptotically stable, since $j_i(\boldsymbol{q_r})$ is a *Lyapunov function* for this control law. This means that the *i-th* agent will eventually reach its goal configuration independently of the other agent trajectories, provided, of course, that no agents have the same goal configuration. Thus, all agents will eventually reach their goal configuration in the same workspace, which means that *the system, as a whole, is asymptotically stable*. What remains now is to prove in mathematical terms the former ascertainment.

According to *step 1*, $j_i(\boldsymbol{q_r})$ is a navigation function, or equivalently, a *Lyapunov function*, for the *i-th* agent. For each *Lyapunov function* $j_i(\boldsymbol{q_r})$ the following properties hold:

a. $j_i(\boldsymbol{q_{rdi}}) = 0$

b. $j_i(\boldsymbol{q_r}) > 0, \quad \text{for}: \quad \boldsymbol{q_r} \neq \boldsymbol{q_{rdi}}$

c. $\dot{j}_i(\boldsymbol{q_{rdi}}) = 0$

d. $\dot{j}_i(\boldsymbol{q_r}) \overset{\text{"almost"}}{<} 0, \quad \text{for}: \quad \boldsymbol{q_r} \neq \boldsymbol{q_{rdi}}$

where: $\boldsymbol{q_{rdi}} = [q_{r1} \quad q_{r2} \quad \dots \quad q_{rdi} \quad \dots \quad q_{rn}]^T$. Taking into consideration that the whole system is interconnected and having in hand a Lyapunov function for every subsystem, a common way to go on with our proof would be to consider a *composite Lyapunov function*, by: $j(\boldsymbol{q_r}) = \sum_{i=1}^{n} b_i \cdot j_i(\boldsymbol{q_r})$, where the b_is are positive constants, and prove each one of properties (a), (b), (c) and (d). The proof of properties (a) and (b) is obvious:

a. $j(\boldsymbol{q_{rd}}) = \sum_{i=1}^{n} b_i \cdot j_i(\boldsymbol{q_{rd}}) = 0$

b. $j(\boldsymbol{q_r}) = \sum_{i=1}^{n} b_i \cdot j_i(\boldsymbol{q_r}) > 0, \quad \text{for}: \quad \boldsymbol{q_r} \neq \boldsymbol{q_{rd}}$

where: $\boldsymbol{q_{rd}} = [q_{rd1} \quad q_{rd2} \quad \dots \quad q_{rdn}]^T$. Differentiating $j(\boldsymbol{q_r})$ with respect to time, we get: $\dot{j}(\boldsymbol{q_r}) = \sum_{i=1}^{n} b_i \cdot \dot{j}_i(\boldsymbol{q_r})$. Thus, the proofs of properties (c) and (d) are also obvious:

c. $\dot{j}(\boldsymbol{q_{rd}}) = \sum_{i=1}^{n} b_i \cdot \dot{j}_i(\boldsymbol{q_{rd}}) = 0$

d. $\dot{j}(\boldsymbol{q_r}) = \sum_{i=1}^{n} b_i \cdot \dot{j}_i(\boldsymbol{q_r}) \overset{\text{"almost"}}{<} 0, \quad \text{for}: \quad \boldsymbol{q_r} \neq \boldsymbol{q_{rd}}$

where in property (d) for the “equal” sign to hold, all agents should be trapped in saddle points. Since the composite Lyapunov function satisfies the desired properties, we conclude that the interconnected system is “almost” globally asymptotically stable.

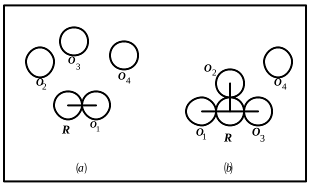
3. Mathematical Tools – Terminology[1]

3.1. "*G*" function

To simplify notation we denote by q instead of q_r the current agent configuration, by q_d instead of q_{rd} its goal configuration, and by q_j instead of q_{oj} the configurations of the obstacles.

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



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, are the combinations of the *n–1* obstacles by 1, i.e. $s_1 = \binom{n-1}{1}$. Respectively, the number of all the possible *level-2* relations are 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: $S_l = \{j \in \mathrm{Z} : 0 < j \le s_l\}$. Obviously, the set of all possible relations of all possible levels is: $S = \{j \in \mathrm{Z} : 0 < j \le s\}$.

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: $(b_{R_j})_l = \sum_{m \in (R_j)_l} b_m$, where the index j denotes the *j-th* relation of *level–l*. To simplify notation, the relation proximity function can be rewritten as: $b_i = \sum_{j \in R_l} b_j$, 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:

$$(g_{R_j})_l = \begin{cases} (b_{R_j})_l + \dfrac{l \cdot (b_{R_j})_l}{\left[(b_{R_j})_l + (B_{R_j^C})_l^{1/h}\right]}, l \le n-2 \\ (b_{R_j})_l, l = n-1 \end{cases}$$

where *?*, h are positive constants, n is the total number of agent – obstacles in the workspace, and: $(B_{R_j^C})_l = \prod_{m \in (R_j^C)_l} (b_m)_l \overset{\text{simplification}}{\Rightarrow} \tilde{b}_i = \prod_{m \in R_l^C} b_m$, 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:

$$g_i(b_i, \tilde{b}_i) = \begin{cases} b_i + \dfrac{? \cdot b_i}{b_i + \tilde{b}_i^{1/h}}, & l \le n-2 \\ b_i, & l = n-1 \end{cases}$$

where n is the total number of objects 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≤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

[1] Terminology and Functions of this section where firstly introduced in [2].

define $\left(B_{R_1^C}\right)_{n-1}$. 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: $g_i \to l$.
- 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: $G = \prod_{l=1}^{n_L} \prod_{j=1}^{n_{R,l}} \left(g_{R_j}\right)_l \overset{\text{simplification}}{\Rightarrow} G = \prod g_i$, 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_is.

3.2. "*f*" function

We define a function $f(G)$ by:

$$f(G) = \begin{cases} a_0 + \sum_{j=1}^{3} a_j G^j, & \text{for: } 0 \le G \le X \\ 0, & \text{for: } G > X \end{cases}$$

which satisfies the following properties:

a. $f(0) = Y$

b. $f'(0) = 0$

c. $f''(0) < 0$

(b, c: local maximum of f at $G = 0$)

d. $f(X) = 0$

e. $f'(X) = 0$

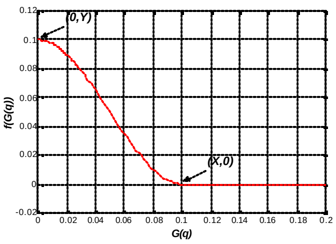
f. $f''(X) > 0$

(e, f: local minimum of f at $G = X$)

where X and Y are positive definite values of G and $f(G)$ respectively. From properties (a), (b), (d), (e) :

$$a_3 = 2 \cdot \frac{Y}{X^3} \qquad a_2 = -3 \cdot \frac{Y}{X^2} \qquad a_1 = 0 \qquad a_0 = Y$$

Properties (c) and (f) can be also easily verified. In the following figure we can see a diagram of the function $f(G)$, for *X*=0.1 and *Y*=0.1.



3.3. Proof of Correctness

Let $e > 0$. Define: $B_{j,l}(e) \equiv \left\{q \in E^n : 0 < \left(g_{R_j}\right)_l < e\right\}$.

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*: $F_0(e) = \bigcup_{l=1}^{n_L} \bigcup_{j=1}^{n_{R,l}} B_j^l(e) - \{q_d\}$
4. The set *away from the obstacles*: $F_1(e) = F - \left(\{q_d\} \bigcup \partial F \bigcup F_0(e)\right)$

Proofs of propositions 1 – 4 can be sent upon demand.

From *proposition 1* we result to some constraints that have to hold in order the destination point q_d to be a non–degenerate local minimum of j. To make those constraints more clear, we give here a definition on what we mean by the destination point q_d. Since the main goal of this work has to do with multiple agent cooperation, we focus there our definition.

Definition 2: *We define by* q_{di} *an "equilibrium" goal configuration which becomes feasible, only when proximity situations with the moving obstacles in the workspace can occur no more. When focused on multiple agents, the former definition implies that all agents have reached their goal configuration, i.e. no agent can obstruct an other that moves towards its goal configuration any more. This is the main sense of cooperation. According to this definition and referring to the i-th agent, at* q_{di} *we have:* $X_i = G_i(q_{d1}, q_{d2}, ..., q_{di}, ..., q_{dn})$

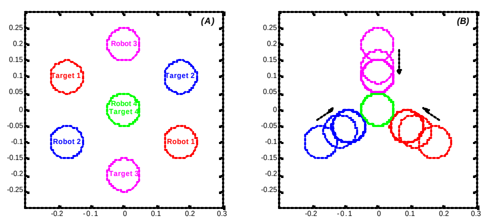
4. Simulation Results

To verify the navigation properties of our decentralized approach, we made a simulation of four and seven holonomic agents that have to navigate from an initial to a final configuration, avoiding collision with each other. The agents are placed at several initial configurations and the constructed paths are recorded and depicted in the relative figures. The chosen configurations constitute non-trivial setups since the straight paths connecting initial and final positions of each agent are obstructed by other agents.

Initial Configurations: q_1 = [.1732, -.1]^T, q_2 = [-.1732, -.1]^T, q_3 = [0, .2]^T, q_4 = [0, 0]^T.

Goal Configurations : q_{d1} = [-.1732, .1]^T, q_{d2} = [.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.



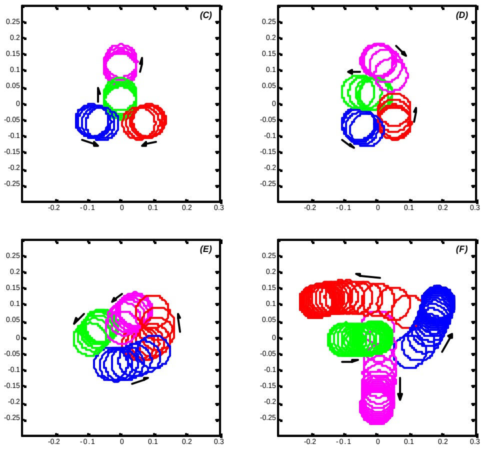


Figure 1: *(A) Initial – Goal Conf., (B) – (E) Intermediate Conf., (F) Intermediate and Final Conf.*

Initial Configurations: q_1 = [-.1299, .075]^T ,q_2 = [0, .15]^T ,q_3 = [.1299, .075]^T ,q_4 = [.1299, -.075]^T ,q_5 = [0, -.15]^T ,q_6 = [-.1299, -.075]^T ,q_7 = [-.35, 0]^T.
Goal Configurations: q_{d1} = [-.1299, .075]^T ,q_{d2} = [0, .15]^T ,q_{d3} = [.1299, .075]^T ,q_{d4} = [.1299, -.075]^T ,q_{d5} = [0, -.15]^T ,q_{d6} = [-.1299, -.075]^T ,q_{d7} = [0, 0]^T.
Parameters: X_1 = .0356, X_2 = .0356, X_3 = .0356, X_4 =.0356, X_5 = .0356, X_6 = .0356, X_7 = .0002, Y = .01, k=64.

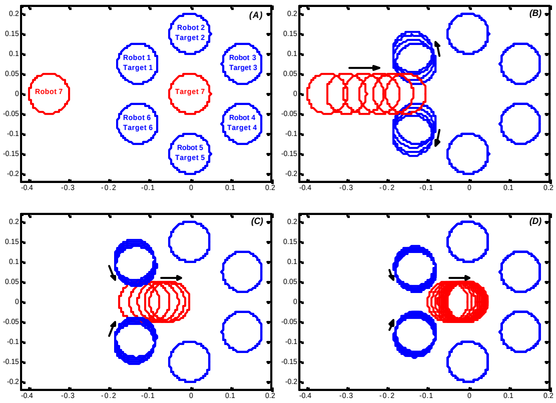


Figure 2: *(A) Initial – Goal Conf., (B), (C) Intermediate Conf., (D) Intermediate and Final Conf.*

5. 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. As in [2], the agent – obstacle potentials are formed by appropriately constructed agent proximity potentials which capture all the possible multi agent proximity situations. At present, the method is not designed to be applied in the more general case of n agents and m moving obstacles existing in the same workspace. However, its great advantage of the is its relatively low complexity wrt the number n of agents, compared to centralized approaches to the problem. 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 in a workspace also including obstacles.

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 Holomonic 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] L. Chun, Z. Zheng and W. Chang, "A Decentralized Approach to the Conflict-Free Motion Planning for Multiple Mobile Robots", IEEE International Conference on Robotics and Automation, pp 1544-1549, 1999

[5] J.P. Desai, J. Ostrowski and V. Kumar, "Controlling Formations of Multiple Mobile Robots", IEEE International Conference on Robotics and Automation, pp 2864-2896, 1998

[6] 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

[7] M. Jäger and B. Nebel, "Decentralized Collision Avoidance, Deadlock Detection and Deadlock Resolution for Multiple Mobile Robots", 2001 IEEE Intern. Conf. on Rob. & Autom., pp1213-1219

[8] 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

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

[10] H.K. Khalil, "Nonlinear Systems", Macmillan Publishing Company, 1992