Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

On Decentralized Navigation Schemes for Coordination of Multi-Agent Dynamical Systems

Hajir Roozbehani, Sylvain Rudaz, and Denis Gillet, Senior Member, IEEE

***Abstract*—Coordination of autonomous non-point agents in four-way crossings is studied in this work. A control scheme based on artificial potential functions is proposed in order to coordinate holonomic agents whose aim is to pass through an intersection while avoiding collisions. To do so, the agents are provided with a decentralized navigation function in which the goal and collision functions are decoupled. The local function defined on each agent requires no global knowledge on the desired destination of other agents. Furthermore, a reachability analysis via Lyapunov functions is proposed in order to guarantee marginal stability in our case study. In addition, conflict-free navigation and convergence properties are verified in simulation.**

I. INTRODUCTION

It is expected that in the near future autonomous navigation of multi-agent systems will be possible. An autonomous system consisting of multiple agents is aimed in most of the cases at solving tasks or achieving goals using interactions among the agents. The interactions, however, can vary from simple semantic operations such as information passing to very complex scenarios such as cooperation, coordination and negotiation [1]. When viewed from a control point of view, particularly in situations where the agents have competing interests or seek different goals, the coordination problem becomes difficult to deal with. Regardless of the purpose of the agents, there are qualitatively two ways to approach this problem: centralized coordination where a central planner decides what each agent should do and decentralized coordination where each agent is provided with the ability to make informed decisions on its own. There exists, however, intermediary levels of decentralization.

The primary motivations for decentralization can be related to the fact that the performance of any central planner is directly related to the accuracy of its knowledge about the framework. Considering the communication limits, calculation costs, and required devices to provide a perfect knowledge for a centralized planner, there are many real world problems in which a decentralized controller can be found preferable. For instance, path planning of multi-vehicle systems has been extensively tackled using decentralized algorithms [2]. In particular, problems such as flocking, formation, coverage, and consensus have been widely studied using decentralized methods [3]-[8].

While most of the mentioned examples deal with the question of how to achieve a collective performance through collaboration, there are certain applications in which the interaction aims mainly at avoiding conflicts while competing for different goals. This is the problem inherent in traffic congestion and collision avoidance which have been widely studied during the past decade.

In this paper, the focus is to develop a flexible, computationally efficient, decentralized algorithm to resolve conflicts among autonomous vehicles with simple internal dynamics at intersections. Hence, the agents should have the ability to independently generate collision-free trajectories in real time. The proposed solution can be further combined with high-level optimization algorithms used for flow management. However, we remain in this work at a middle level and consider developing an intermediate navigation scheme which takes the desired destination as input and generates a collision free trajectory given the current state of other vehicles. We have developed a platform in MATLAB in order to simulate and verify the proposed navigation function. The vehicles are called independent non-point agents, or simply agents as the focus is more on navigation functions taking into account very simple low order dynamics. However, the general pattern gives a good insight on the effectiveness of the proposed solution and how it behaves under various dynamics.

The crossing strategies studied in this work are based on the use of potential function, that is a function which attracts an agent toward its goal while repelling it from dynamic and static obstacles. While the concept originated in motor-schema architecture [10], the functions were introduced in [9] in order to drive a point-mass agent from an initial position to a desired destination.

Decentralized Navigation Functions (DNF) were recently introduced to motion control of groups of robots [11]. This work has been further extended to cope with different dynamics and constraints [12]. The stability properties and convergence of the so-call DNF methodology is further verified both analytically and in simulation. In a separate work [7], a different navigation function is developed for formation control of multi-agent systems using different control regions which allowed a higher level of decentralization compared to that of [11] and [12].

Although DNF has shown interesting stability and convergence properties, there are practical as well as theoretical concerns that motivate a different approach. For instance, DNF is computationally expensive as it attempts to consider all possible permutations of collision relations among the agents. Moreover, asymptotic stability can be accomplished via DNF only if a global knowledge about the final configuration of the system is available. In addition, it would be more complicated to maintain the interesting stability properties of DNF in cases where the number of agents is variant as it is the case in

All authors are with the Automatic Control Laboratory, EPFL `firstname.lastname@epfl.ch`

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

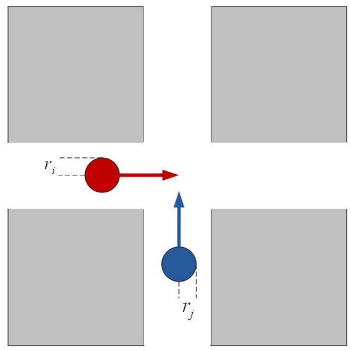


Fig. 1. A conflict scenario with two agents

intersections. We are particularly interested in those solutions that can cope with varying number of vehicles.

In this work, a decentralized navigation function is defined on each agent based on the position of dynamic and static obstacles in the environment assuming a full-range communication. It should be noted that in the proposed method the agents can cooperate to avoid collisions even if they have already settled at their desired position. Consequently, asymptotic stability would not be accomplished in our solution while marginal stability can be guaranteed. This implies certain constraints on the final destination of the agents. Moreover, the number of agents is not assumed to be known a priori.

Reducing the number of conflicts is always desirable. To do so, we provide a strict leader based formation control in order to reduce the complexity of the proposed approach. By clustering into lines, the agents are able to free some space which can potentially lead to a significant increase in the throughput of the system.

The rest of the paper is organized as follows: In section II the problem formulation is presented. Section III describes the proposed navigation function. Stability properties are analyzed in section IV. Section V presents a strict leader-follower formation network. We present the results in section VI and conclude in section VII.

II. PROBLEM FORMULATION

We consider a conflict scenario involving a set of N agents passing through the same intersection $I \subset \mathbb{R}^2$. $\mathbf{q}_i \in I$ represents the position of the center of agent i which is assumed to occupy a disc of radius r_i. We further assume that each agent has access to its own global position and desired destination $\mathbf{q}_{d_i} \in D$ where $D \subset \mathbb{R}^2$ stands for the set of desired destinations. Moreover, an agent can not have any access to the desired position of other agents. However, it perceives the position of other agents which are present in the range of the intersection. Fig. 1 shows a conflict scenario involving two independent agents. Furthermore, it is assumed that the motion of each agent is described by a first order dynamics:

$$\dot{\mathbf{q}}_i = \mathbf{u}_i \tag{1}$$

for each i where $\mathbf{q}_i = (q_{x_i}, q_{y_i})^T$ and $\mathbf{u}_i = (u_{x_i}, u_{y_i})^T$ are respectively the state and control vectors. It should be emphasized that a first order dynamics has been considered to focus our study on the trajectories that can be achieved using the proposed navigation function. The problem is to find the proper control inputs for each agent i such that the controlled agent, undergoing the dynamics given in (1), can reach its destination while avoiding collisions with other agents $j \neq i$. In addition, it is assumed that the desired destinations are located somewhere outside of the working space, i.e, $(D \cap I) = \emptyset$ so that the agents disappear from the environment when they reach their destination. Therefore, the concept of convergence to a final configuration is not critical in this work.

III. THE PROPOSED DECENTRALIZED NAVIGATION FUNCTION

A navigation function can be viewed as a smooth mapping which is analytic in the working space and whose negated gradient is attractive toward the goal and repulsive from moving and static obstacles. Therefore, such a function can be combined with proper control laws in order to derive agents toward their destination on collision free trajectories. In this section, we try to come up with a navigation function suited for our purposes. Though the navigation functions studied in [7,11,12] provide strong analytical results, there are few discussions in terms of their scalability and calculation cost. We provide schemes that are less sensitive to the number of the involved agents and yet less computationally expensive. We take intuitive inspiration from the methodologies studied in [7,11,12] and develop a new mapping between the state of the agents and their corresponding control inputs:

$$\begin{aligned}\phi_i &= \gamma_1 \left\|\mathbf{q}_i - \mathbf{q}_{d_i}\right\|^2 + \gamma_2 \sum_{j \neq i} \frac{1}{\alpha + \|\mathbf{q}_i - \mathbf{q}_j\|^2 - (r_i + r_j)^2} \\ &+ \gamma_3 \sum_{k=1}^{m} c_k(\mathbf{q}_i)\end{aligned} \tag{2}$$

The above function is composed of three terms. The first term is the squared distance of agent i from its destination and attains small values as the agent approaches the goal. The second term is a collision relation between agents i and all other agents and its negated gradient is repulsive from them. It is worth mentioning that no agents can enter the physical boundary of other agents, therefore, the denominator of the second term is always positive provided that α is a positive constant. Furthermore, it is interesting to note that the closer two agents are the higher is the value of the second term. The collision relation of agent i is composed of a constant γ_2 which determines the importance of the second term with respect to other terms, and a rational function proportional to the inverse of its relative distance from other agents. It should be emphasized that the negated gradient is the most repulsive from the nearest agent. The third term is to avoid static obstacles and walls, i.e., $c_k(\mathbf{q}_i)$ is a positive definite function which indicates the k-th nonlinear constraint of the environment. r_i and r_j are radius of the discs occupied by agents i and j respectively. α,γ_1,γ_2 and γ_3 are positive constants. It is worth mentioning that the constant α determines the magnitude of the repulsive force generated by agent i which is experienced by agent j when it has come to a close vicinity of agent i. This number can be used to define a security region around each agent, that is, a region around the agent where the repulsive force is stronger that the attractive force so that no two agents run into collision assuming that they start somewhere outside of their security areas.

Definition 1: A security zone $S_r(q_i)$ is defined as the Euclidian ball of radius r centered at q_i with $r > r_i$.

In section IV, the criteria to choose these constants will be discussed in more details. The same control law as the one proposed in [11,12] is applied in order to generate trajectories which lie on the negated gradient of the above navigation field:

$$u_{q_i} = -K_i \frac{\partial \phi_i}{\partial q_i} \tag{3}$$

Note that u_{q_i} denotes an arbitrary component of the control vector. We also let $q_i(0) \in \mathbb{R}$ describe an arbitrary component of the initial position of agent i in the workspace. Intuitively, the above control law attempts to minimize the navigation function in (2) by deriving the controlled agent i on the negated gradient of the function. In particular, the control law described above is a minimizer for the following optimal control problem:

Proposition 1: The control law given in (3) satisfies the local optimality axioms for the following cost function:

$$J_i = \int_0^T (u_i^T u_i + K_i \nabla \phi_i^T \nabla \phi_i) dt, \;\; i = 1, ..., N$$

subject to (1) *Proof:* Define the Hamiltonian as follows:

$$H = u_i^T u_i + \lambda^T u_i + K_i \nabla \phi_i^T \nabla \phi_i$$

optimality requires that:

$$\lambda^* = -2u_i^*$$

which is:

$$\lambda^* = 2K_i \nabla \phi_i$$

it remains to show that:

$$\dot{\lambda} + \frac{\partial H}{\partial q} = 0$$

We start from $\lambda^* = 2K_i \nabla \phi_i$ and derivate both sides w.r.t time to obtain:

$$\dot{\lambda}^* = 2K_i \nabla \dot{\phi}_i$$

that is:

$$\dot{\lambda}^* = 2K_i \nabla^2 \phi_i \dot{q}_i = -2K_i \nabla^2 \phi_i \nabla \phi_i$$

Having $\dot{\lambda}^*$ from the above, one can write:

$$-2K_i \nabla^2 \phi_i \nabla \phi_i + 2K_i \nabla^2 \phi_i^T \nabla \phi_i = 0 \qquad \text{Q.E.D}$$

Equation (2) when combined with (3) will generate trajectories which are directed toward the goal and avoid both static and moving obstacles. However, the positive gradient field decreases as the agents approach their destination. Given that our interest is to make the agents leave the intersection as fast as they can, one can either consider a different control law or keep the first term in the navigation function as a constant, i.e., to reach the final state by incrementally changing the target position. It should be noted that different control laws are implementable as well. For instance, this can be done by combining the above navigation function with control Lyapunov functions in an inverse optimal control scheme where the navigation function is the cost to be minimized by the controller. This would allow obtaining other patterns and trajectories. In [14], such a control law is considered.

Under the assumption that the desired positions are located far enough from one another, the definition given in [11] that the desired configuration is a minimum of the navigation function would be satisfied with the difference that the minimum is no longer zero. Hence, it can be stated that the approach is close to complete decentralization as it only makes a simple assumption on the final configuration of the system. We have assumed that:

- each agent has access to the relative position of other agents;
- each agent has access to its own position in the workspace;
- each agent has access to its own desired position;
- the desired positions of the agents are located far enough from each other.

It is worth mentioning that the authors in [11]-[12] have also assumed that the final destinations are far enough from each other, otherwise, the cooperation term would not let the agents settle at their desired location. To make asymptotic stability possible, they have considered switching to a different control law when the team has reached a close enough distance to its final configuration.

IV. REACHABILITY ANALYSIS AND CONTROLLER DESIGN

It should be noted that we are interested in providing the agents with the ability to pass through the intersection and leave it from any other side. Therefore, it is the concept of reachability which should be considered. By constructing a local reachability region for each agent and applying Lyapunov analysis, we intend to derive requirements which when fulfilled can ensure that the controlled agent would be able to reach its desired state. This way of analysis however, is suitable in our case and can not be extended to the problems where asymptotic stability is desirable such as [11,12]. To this goal, we let $g(\mathbf{q}_i)$ be a local function defined on each vehicle which is negative in each direction as long as the vehicle has not reached its destination. Each component of $g(\mathbf{q}_i)$ is defined as:

$$g(q_i) = \begin{cases} (q_i - q_{d_i}) & if \; q_i(0) < q_{d_i} \\ -(q_i - q_{d_i}) & if \; q_i(0) \geq q_{d_i} \end{cases} \tag{4}$$

The agents start in $g(\mathbf{q}_i) < 0$ and the region they are interested to reach is defined by $g(\mathbf{q}_i) \geq 0$, where the inequalities refer to componentwise operations. The agent's dynamics is given by $\dot{\mathbf{q}}_i = \mathbf{v}_i$. Let's use the proposed control law (3) where ϕ_i is the local navigation function given by (2). Let $V(\mathbf{q})$ be a Lyapunov function of the form:

$$V(\mathbf{q}) = \sum_{i=1}^{n} (\mathbf{q}_i - \mathbf{q}_{d_i})^2 \tag{5}$$

which is constructed locally on $\mathbf{q}_i$ and strictly positive except at:

$$\mathbf{q} = [\ \mathbf{q}_{d_1} \quad \mathbf{q}_{d_2} \quad \dots \quad \mathbf{q}_{d_n}]$$

where it is zero. We note that for all initial conditions $V(\mathbf{q}) > \mathbf{0}$. Its derivative $\dot{V}$ along any system trajectory is:

$$\dot{V} = \sum_{i=1}^{n} \frac{\partial V}{\partial \mathbf{q}_i}\dot{\mathbf{q}}_i = \sum_{i=1}^{n} -2K_i(\mathbf{q}_i - \mathbf{q}_{d_i})[2\gamma_1(\mathbf{q}_i - \mathbf{q}_{d_i}) + \gamma_2 \sum_{j\neq i} \frac{(\mathbf{q}_i - \mathbf{q}_j)}{(\alpha + (\mathbf{q}_i - \mathbf{q}_j)^2 - (r_i + r_j)^2)^2} + \gamma_3 \sum_{k=1}^{m} \frac{\partial c_k}{\partial \mathbf{q}_i}(\mathbf{q}_i)] \tag{6}$$

Since the vehicles have no exact information about the destination of each other, one could adjust the weights such that each component of the term in the brackets has the same sign as the corresponding component of $\mathbf{q}_i - \mathbf{q}_{d_i}$ for each vehicle. In other words, in order to reach $g(\mathbf{q}_i) \geq 0$, it is required that $V(\mathbf{q})$ decreases as long as the $g(\mathbf{q}_i) \geq 0$ is not reached. Furthermore, we note that $\mathbf{q}_i - \mathbf{q}_{d_i}$ does not change its sign as long $g(\mathbf{q}_i) < 0$:

$$2\gamma_1(\mathbf{q}_i - \mathbf{q}_{d_i}) - \gamma_2 \sum_{j\neq i} \frac{(\mathbf{q}_i - \mathbf{q}_j) - (r_i + r_j)^2}{(\alpha + (\mathbf{q}_i - \mathbf{q}_j)^2 - (r_i + r_j))^2} + \gamma_3 \sum_{k=1}^{m} \frac{\partial c_k}{\partial \mathbf{q}_i}(\mathbf{q}_i) = \xi(\mathbf{q}_i) \tag{7}$$

where we choose $\xi(\mathbf{q}_i)$ to be a constant vector whose components have the same sign as those of $\mathbf{q}_i(0) - \mathbf{q}_{d_i}$. The analysis provided in this section can be utilized as a basis to design controllers which make it possible for the agents to reach their ultimate goal. However, it should be emphasized that there is no notion of obstacle avoidance in the concept of reachability or stability. Hence, it is inevitable to make some configurations unreachable in order to guarantee collision avoidance. This could be the most critical point in the system where the agents are most likely to run into collisions. More details will be provided in the next section.

V. Non-Cooperative Formation Control

We address the problem of formation control in this section and further apply it to provide stable platooning in multi-vehicle systems. Four types of vehicles can be distinguished in our framework, depending on the direction of their motion. Roughly speaking, the vehicles that move toward the same direction can be said to have similar interests. Intuitively, we expect to see an improvement on the overall capacity of the intersection by urging those vehicles which have similar goals into platoons and free some space for other vehicles to pass.

A feedback formation protocol is introduced which can be combined with the control law of section III, in order to achieve conflict-free navigation. We need an additional definition for conflict-free platooning.

Definition 2: For every leader i, the nearest neighbor for each vehicle is the one among its neighbors that is the closest to its destination and is outside of its security zone:

$$N_N^i(G) = \{v \in N^i(G) \mid \ \|q_v - q_{d_i}\| \leq \|q_j - q_{d_i}\| \\ for\ all\ j \in N^i(G)\}$$

where the set of the neighbors for each vehicle is given by $N^i(G)$.

The following protocol is proposed:

Feedback Formation Protocol: Each agent i, except for the leader, follows its nearest neighbor $j = N_N^i$ of the its same type through the following formation dynamics:

$$\dot{q}_i = q_j - q_i + d_{ij}$$

provided that $q_j \notin S_r(q_i)$, which is the consensus equation except for d_{ij} which is distributed as additional error over the consensus. The control law of (3) is applied for all $q_k \in S_r(q_i)$. One component of d_{ij} is always zero, depending on the type of the agents. The other components must be defined in such a way that prevents collisions and imposes a desirable security zone for the platoon: $r_i + r_j < |d_{ij}| < r_i + r_j + s$, $s <= r_{min}$ with r_{min} being the radius of the smallest disc. The leader of each type is the closest to leave the intersection.

It should be noted that the underlying graph used in this section is not a structural formation graph and, therefore, the security zone is defined to guarantee collision avoidance amongst the members of a group. Local changes can be made to the above protocol in order to guarantee collision avoidance between platoons. It suffices for every agent to use the control law given in (3) to avoid agents of other platoons (or those of its own platoon that have entered its security zone) and follows the above protocol to join the agents of its own type. Once the platoon is stable, the agents would be safe using the above protocol. Besides, it is shown in [14] that the proposed platoon graph (a tree in this case) is leader-follower controllable, i.e., the followers can be driven to their destination by controlling the motion of the leader.

VI. Simulation

In this section numerical simulation is hired in order to demonstrate the functionality of the proposed navigation function. We start with designing a controller for each agent based on the results of the previous section. As the first step, we intend to come up with some design rules, i.e., to define the reachable region where (7) is satisfied. Thus, we fix the positive gradient and choose the gains for the collision relation such that (7) is satisfied outside the security region of the agent i, i.e., no other agent should be allowed to enter the security zones. As an example, we consider the design process for choosing the gains of the navigation function in (2). We let $\alpha = 0.01$ and $q_i - q_{d_i}$ be fixed at 12 (2 units bigger than the size of the environment used in the simulation section), $\gamma_1 = 0.5$, and $r_i = 0.3$ for all the agents. The security boundary for moving obstacles is fixed at $s_{ij} = 0.15$, i.e., $q_i - q_j$ should be always greater than 0.75 which yields:

$$\gamma_2 = \frac{2\gamma_1(q_i - q_{d_i})}{(r_i + r_j + s_{ij})}(\alpha + (r_i + r_j + s_{ij})^2 - (r_i + r_j)^2)^2 = 0.73$$

It would be reasonable to assume a larger security region, here arbitrarily chosen to be $s_w = 0.3$, for structured obstacles as they can not cooperate in collision avoidance:

$$\gamma_3 = \frac{2\gamma_1(q_i - q_{d_i})}{(r_i + s_w)}(\beta + (r_i + s_w)^2 - (r_i)^2)^2 = 1.6$$

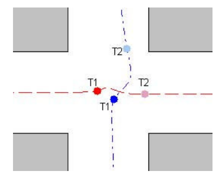
with $\beta = \alpha$. We arbitrarily choose $K_i = 1$ in the control law of agent i. We apply these same parameters for the numerical simulations in this section.

A conflict scenario involving two agents is shown in Fig. 2. The agent are able to resolve the conflict even though they are situated in symmetric configuration. The agents in this example utilize the same controllers and are encountered by a symmetric conflict to solve which they rely on the principal property of saddle points, i.e., any small deviation would make the agents leave their path and continue to resolve the conflict.

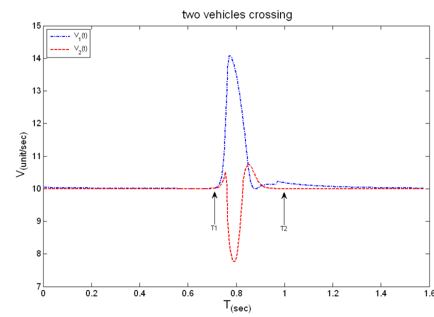
A view of the platform in operation using the non-cooperartive formation control is shown in Fig. 5. (In [13] the role of formation is demonstrated in movies). Starting from an initial configuration, the agents follow the formation protocol to follow their nearest neighbor till they reach the intersection at which time the leaders start to avoid other platoons using the control law in (3).

When viewed as a multi-agent system, the agents working under the assumptions listed earlier in section III and using the control law (3) are involved in a non-cooperative scenario as each of them is trying to reach its own destination as quickly as possible through minimizing a navigation function. However, there is also a sense of cooperation as the agents admit an equilibrium strategy in which they cooperate in the collision relation of each other. Therefore, though a single agent is not concerned about how the ensemble is working, whenever required it interacts with other agents in such a way that no collision occurs. For instance, as shown in Fig. 2 a controlled agent would willingly deviate from its optimal path, that is a straight path drawn from the source to the destination, in order to resolve a conflict.

More complex scenarios can be studied using the proposed navigation function. Fig. 3 presents the results at the presence of a static obstacle in the center of the intersection. Both agents have to deviate more from their optimal path compared to the previous scenario. Fig. 4 considers a scenario with numerous agents in the environment. The shape of the trajectory and the speed of the vehicles can change depending on the choice of design parameters. Ideally, the optimal policy for an agent would be to move on a straight line directed toward its goal. The problem appears when there are moving or static obstacles on this straight line. In such a case, the agent has to leave its path, avoid the encountered obstacle, and enter the path again. The larger is the security bound around the agents, the sooner they leave their path, and the later they come back to it.

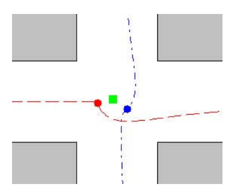


(a) Trajectories

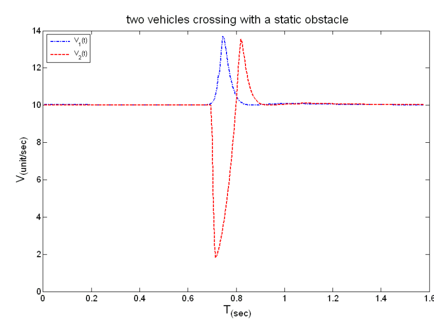


(b) Velocities

Fig. 2. Simulation in MATLAB: *a) trajectories:* the agents are able to resolve conflicts using the decentralized navigation function in (2). *b) velocities:* One agent decelerates and the other accelerates in order to resolve the conflict.

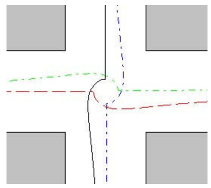


(a) Trajectories

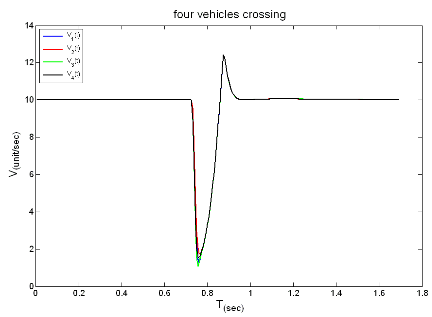


(b) Velocities

Fig. 3. Simulation in MATLAB: *a) trajectories:* the agents are able to resolve conflicts using the decentralized navigation function in (2). *b) velocities:* When confronted by a static obstacle (square), one agent is forced to decelerate much more comparing with Fig. 2b.

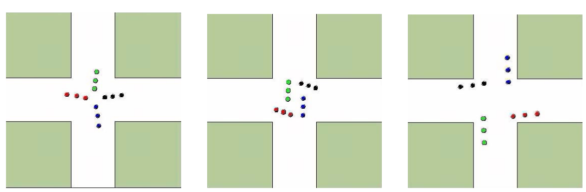


(a) Trajectories



(b) Velocities

Fig. 4. Simulation in MATLAB: *a) trajectories:* from a scenario involving 4 agents passing through the same crossing in a symmetric situation. *b) velocities:* They all decelerate at the same time and accelerate after resolving the conflict. This might imply some lose of efficiency. The problem with head-to-head encountering is mitigated in [14] at the cost of heavy computations.



(a) Conflict (b) Avoidance (c) Resolved

Fig. 5. Each leader avoids its adjacent platoons. The followers will be safe by maintaining the formation (see [13] for visual demonstrations)

VII. CONCLUSIONS AND FUTURE WORKS

A decentralized navigation scheme has been proposed in this work. Decentralized conflict resolution together with global convergence properties were analyzed and further verified in simulation for holonomic agents with first order dynamics. The proposed navigation function requires no global knowledge about the final configuration of the system which makes it suitable for the cases when the number of participating agents is variant as it is the case in intersections. It is assumed that the desired position of the agents are located far enough from one another which seems reasonable in intersections. However, no further information is required on the desired configuration of the agents. Furthermore, the complexity of the proposed methodology increases linearly with the number of agents (see (2)) which makes it implementable for large scale multi-agent systems. The collision-free trajectories were obtained using the negated gradient field. It is interesting to note that the proposed navigation function offers a high flexibility in the sense that new terms can be easily added or removed in order to achieve different patterns such as clustering or platooning. One should keep in mind that it would be more complicated to provide asymptotic stability for the navigation function used in (see (2)) which is a desirable property in platooning. This is the reason why a different protocol is considered for formation control. Nevertheless, the flexibility of the decentralized scheme was exploited in the formation section through having a different control law for the collision avoidance than that of formation control.

A future research can focus on the feasibility of extending the developed navigation function to provide asymptotic stability wherever required. A further research can aim at combining the navigation function with a high level flow manager for the case of multiple intersections and evaluate the efficiency of the algorithm in heavy traffics. Lastly, we intend to investigate the features of the proposed navigation function when combined with optimal control schemes where the navigation function is the cost to be minimized by the controller.

REFERENCES

[1] Jennings, N. R. and Bussmann, S. (2003) Agent-based control systems. *IEEE Control Systems*, 23 (3). pp. 61-74.

[2] Azarm K., Schmidt, G., Conflict-Free motion of multiple mobile robots based on decentralized motion planning and negotiation, *Proceeding of IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997.

[3] Dimarogonas D. V., Kyriakopoulos K. J., A Feedback Control Scheme for Multiple Independent Dynamic non-Point Agents, *International Journal of Control*, vol. 79, no. 12, December 2006, pp. 1613-1623.

[4] Jadbabaei A., Lin J., Morse A. S., Coordination of Groups of Mobile Autonomous Agents using Nearest Neighbor Rules, *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 988-1001, June 2003.

[5] Saber R.O., Murray R.M., Flocking in multi-agent dynamic systems: Algorithms and theory, *IEEE Trans. on Automatic Control*, vol, 51, March 2006.

[6] Beard R., Ren W., Trajectory Tracking for Unmanned Air Vehicles with Velocity and Heading Rate Constraints, *IEEE Trans. on Control Systems Technology*, vol. 12 no.5. September 2004.

[7] De Gennaro M.C, Jadbabaie A., Formation Control for a Cooperative Multi-Agent System using Decentralized Navigation Functions, *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, June 14-16,2006.

[8] Schwager M., Rus D., Slotine J.J., Decentralized, Adaptive Control for Coverage with Networked Robots, *Proc. International Conference on Robotics and Automation*, Rome, April, 2007.

[9] Rimon E. and Koditschek D.E., Exact robot navigation using artificial potential functions, *IEEE Trans. on Robotics and Automation*, vol. 8, no. 5, pp. 501-518, 1992.

[10] Arkin R.C., Motor Schema Based Mobile Robot Navigation. *Int. J. of Robotics Research*, 1989, Vol. 8, No. 4, pp. 92-112.

[11] Dimarogonas D.V.,Zavlanos M. M.,Loizou S. G., and Kyriakopoulos K.J., Decentralized motion control of multiple holonomic agents under input constraints, *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii USA, December 2003.

[12] Dimarogonas D.V., and Kyriakopoulos K.J., A feedback control scheme for multiple independent dynamic non-point agents, *International Journal of Control*, 79(12),(2006) pp. 1613-1623.

[13] http://lawww.epfl.ch/page34497.html

[14] Roozbehani H.,Rudaz S.,Gillet D., Optimal Cooperative Control of Multi-Vehicle Systems in Intersection, *submitted to the IEEE International Conference on Systems, Man, and Cybernetics*, October 11-14, 2009, San Antonio, Texas, USA.