Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Decentralized Control of Swarms with Collision Avoidance Implications

Marta Niccolini * **Lorenzo Pollini** * **Mario Innocenti** *

* *Department of Electrical Systems and Automation, University of Pisa, 56126 Pisa, Italy.*

Abstract: The paper reviews some approaches to the decentralized control of a swarm of unmanned vehicles, and then proposes a new algorithm capable of managing collisions between vehicles and with obstacles. The swarm goal is to achieve a desired shape and position in space, formalized using an abstraction based approach. Formation statistics are defined in analogy with phisical bodies: center of mass position and inertia moments. Each agent elaborates its own estimate of these variables using a consensus algorithm capable of tracking a ramp reference, in order to reduce tracking errors. Gyroscopic and damping terms are added to the control law in order to avoid collision between vehicles and with obstacles. The obstacle avoidance terms appear in the control law only in the presence of obstacles or nearby vehicles; thus system dynamics change during the system evolution. This behavior was modeled as a hybrid system and proof of stability is given, under mild conditions, using the Common-Lyapunov function approach. The proposed methodology is validated through extensive numerical simulation.

Keywords: Swarm behaviour and multi-agent systems; Cooperative perception.

1. INTRODUCTION

Several approaches for swarm control have been proposed in recent years. Leader-Follower techniques [1] assume the presence of a swarm leader, which may be a physical vehicle or a ground-based station serving as a reference for all the agents, which move autonomously following some kind of geometrical law. The main drawback is the intrinsic dependance of the swarm health on the leader's health. Other approaches use Artificial Potential Functions ([2],[3]), which specify attraction and repulsion forces that each agent of a swarm computes depending on its neighbors location. Some authors assume that each agent knows the complete swarm state (e.g. all other agents positions and velocities), and this may be unrealistic with sparsely distributed swarms and limited communication ranges due to limited power consumption. Cohesion of the swarm can be demonstrated but the swarm shape cannot be easily controlled. The Virtual Structure approaches [4] define a set of geometrical constraints for the swarm so that the resulting agent formation is single and stiff. Within this framework, determination of the trajectories of the swarm is computational intensive, and must be performed in a centralized way. The central control system may be substituted by a virtual leader (VL), which follows the desired trajectory; additional geometrical constraints with respect to the VL are used to drive the swarm. The Behavior based techniques [5] address the intrinsic complexity of coordination of a swarm of unmanned vehicles by defining a set of basic tasks, which determine simple behaviors (e.g. move toward goal, avoid the nearest obstacle, maintain formation, etc.). Each agent performs its own tasks independently resulting into a so called Emergent Behavior of the swarm, which should be somehow related to the actual desired swarm behavior. Fully decentralized estimation and control is exploited by Consensus based approaches [6] where each agent state is driven to converge to a common swarm value (e.g. velocity, direction, position). This class of control systems has been employed to solve rendez-vous, flocking and coverage problems. Analysis of convergence and stability of equilibrium points are usually performed using graph theory. Consensus algorithms may be employed for distributed estimation of swarm variables [7] to be used as feedback for a swarm control system. The goal of Abstraction based techniques ([7],[9]) is to reduce the complexity of swarm coordination by introducing a set of abstract swarm variables, which encode its shape and position. This approach disregards agents' positions as long as the swarm has the desired shape. Collisions between agents and with obstacles are not considered so far.

The aim of the paper is a decentralized control system capable of managing collisions with obstacles and between vehicles. The swarm goal is to achieve a desired shape and desired position in space. This is specified using an abstraction based approach, which defines formation statistics in analogy with physical bodies: center of mass position and inertia moments, for instance. Defining formation statistics, which are representative of the desired formation shape and position is a non trivial task, thus a technique for easy computation of formation statistics is given. Afterwards, each agent elaborates its own estimate of these variables using a consensus algorithm. A consensus algorithm capable of tracking a ramp reference was developed, in order to reduce tracking errors, and control saturation. The proposed decentralized control law curves the vehicle trajectory when the vehicle gets near to an obstacle or another vehicle; an additional braking term is employed to reduce vehicle velocity inversely proportional

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.3299

to the obstacle distance. The obstacle avoidance terms appear in the control law only in the presence of obstacles or nearby vehicles. This dynamic behavior was modeled as a hybrid system and its stability was studied using the Common-Lyapunov function approach.

2. PROBLEM STATEMENT

The paper proposes a decentralized control scheme for a swarm of vehicles capable of achieving desired formation statistics and of avoiding obstacles and collisions within the swarm itself. The swarm is composed of n agents with positions $p_1, ..., p_n$. Each agent's position vector has the dimensionality of the motion space under consideration, and in this work planar motion is assumed, thus the i-th vehicle position is: $p_i = [p_{ix}\ p_{iy}]^T$.

The swarm statistics are defined via an abstraction-based approach, introduced in [7], which employs the inertia moments of the swarm as shape variables to be tracked by the control system. The statistics are represented by the swarm moment vector $f(p) \in \Re^l$, which uses a C^2 vector moment generating function $\phi : \Re^m \rightarrow \Re^l$:

$$f(p) = \frac{1}{n}\sum_{i=1}^{n}\phi(p_i) \tag{1}$$

Each agent is modeled by a double integrator. It knows the desired swarm moment vector, represented by the vector $f^* \in Im(f)$, and measures its position and velocity. The agent's control input is vehicle acceleration.

The communication network, over which the vehicles may exchange information, is assumed to be distance dependent, i.e. each agent can communicate with its neighbors only if the relative distance is less than some specified communication radius. This is one of the commonly used model for describing the time-varying topology in mobile sensor networks, and accomplishes quite well with the mode of operation of commercial wireless radio modems.

In order to achieve the goal, each vehicle must estimate the current swarm statistics. This is done using a dynamic consensus algorithm capable of tracking the mean values of the vector moments of all the agents, under the assumptions that the networks is connected and undirected.

Collisions between obstacles and other agents will be taken into account using a gyroscopic correction term and appropriate braking forces [10] and [11].

3. MOMENT STATISTICS

The moments of a set of unit mass points in a plane are defined as:

$$M_{ab} = \frac{1}{n}\sum_{i=1}^{n} p_{ix}^a p_{iy}^b, \quad a, b \geq 0$$

where $a+b$ is the order of the moment. Even if moments can provide an exact formation description, we are interested in low order moments through which we specify a family of formations. We will focus on first and second order moments, since they give the minimum number of statistics sufficient for specifying pose, orientation and shape of the swarm. Under this assumption, the vector moment generating function is:

$$\phi(p_i) = [p_{ix}\ p_{iy}\ p_{ix}^2\ p_{ix}p_{iy}\ p_{iy}^2]^T \tag{2}$$

The formation statistics are:

$$\begin{aligned} \mu_x &= \frac{1}{n}\sum_{i=1}^{n} p_{ix} \qquad \mu_y = \frac{1}{n}\sum_{i=1}^{n} p_{iy} \\ M_{2x} &= \frac{1}{n}\sum_{i=1}^{n} p_{ix}^2 \quad M_{2y} = \frac{1}{n}\sum_{i=1}^{n} p_{iy}^2 \\ M_{xy} &= \frac{1}{n}\sum_{i=1}^{n} p_{ix}p_{iy} \end{aligned} \tag{3}$$

The first-order moments μ_x and μ_y specify the pose of the swarm, i.e. the center of mass. Through M_{xy} is possible to define the orientation as the rotation of the reference frame centered in the center of mass for which the formation satisfies:

$$\sum_{i=1}^{n} x_i^{rel} y_i^{rel} = 0 \tag{4}$$

The relative variables can be written as:

$$[x_i^{rel}\ y_i^{rel}]^T = R^T(p_i - \mu). \tag{5}$$

$$\theta = \frac{1}{2}atan\left(\frac{2\sum_{i=1}^{n}(p_{ix}-\mu_x)(p_{iy}-\mu_y)}{\sum_{i=1}^{n}(p_{ix}-\mu_x)^2 - (p_{iy}-\mu_y)^2}\right) \tag{6}$$

where θ is restricted to be between $[-\frac{\pi}{2}, \frac{\pi}{2}]$, and $\mu = [\mu_x, \mu_y]^T$.

The shape of the swarm is assumed to be independent from its pose; geometric considerations can be made using relative second-order moments:

$$\begin{aligned} a_1 &= \frac{1}{n}\sum_{i=1}^{n}(x_i^{rel})^2 \\ a_2 &= \frac{1}{n}\sum_{i=1}^{n}(y_i^{rel})^2 \end{aligned} \tag{7}$$

These relative abstract variables provide a bound for the region occupied by the agents, as derived in [9]:

$$\begin{aligned} |x_i^{rel}| &\leq \sqrt{na_1} \\ |y_i^{rel}| &\leq \sqrt{na_2} \end{aligned} \tag{8}$$

It is then possible to determine a spanning rectangle, centered in μ, and with sides given by $2\sqrt{na_1}$ and $2\sqrt{na_2}$.

These measures are expressed in the relative frame. Therefore, using the equivalence in (5) a convenient expression for calculating inertia moments has been derived:

$$\begin{aligned} M_{2x} &= \frac{1}{2}(2\mu_x^2 + a_1 + a_2 + (a_1 - a_2)\cos 2\theta) \\ M_{2y} &= \frac{1}{2}(2\mu_y^2 + a_1 + a_2 - (a_1 - a_2)\cos 2\theta) \\ M_{xy} &= (\mu_x\mu_y + (a_1 - a_2)\sin\theta\cos\theta) \end{aligned} \tag{9}$$

From (9) we can come back to the relative shape variables using the following:

$$\theta = \frac{1}{2}atan\left(\frac{2n(M_{xy} - \mu_x\mu_y)}{n(M_{2x} - \mu_x^2 - M_{2y} + \mu_y^2)}\right) \tag{10}$$

$$a_1 = (M_{2x} - \mu_x^2)\cos^2\theta + (M_{2y} - \mu_y^2)\sin^2\theta + (-\mu_x\mu_y + M_{xy})\sin^2\theta \quad (11)$$

$$a_2 = (M_{2y} - \mu_y^2)\cos^2\theta + (M_{2x} - \mu_x^2)\sin^2\theta + (-\mu_x\mu_y - M_{xy})\sin^2\theta \quad (12)$$

Using (9) is possible to obtain rigid translations by selecting (a_1, a_2, θ) and deriving the corresponding second-order moments.

4. CONSENSUS ESTIMATION

Dynamic consensus algorithms are used to estimate the swarm current moment vector. A consensus algorithm allows the internal variables of all agents to converge to a common value in a decentralized way. Following [8] we distinguish between static and dynamic consensus; in the static case the consensus is achieved at the average of the initial values of all agents variables. On the other hand, a dynamic consensus algorithm requires an additional input variable for each agent; the problem is then to track the average of the input values with the consensus variables.
Consensus algorithms are based on the topology of the underlying communication/sensing graphs. A graph is defined as a set of nodes and arcs. The adjacency matrix (A) is a $n \times n$ matrix whose $ij - th$ entry is 1 if the edge (i,j) is included in the graph, and 0 otherwise. The diagonal degree matrix is defined as follows:

$$D_{ii} = \sum_j A_{ij}.$$

The Laplacian matrix is defined in terms of the adjacency matrix and the diagonal degree matrix:

$$L = D - A.$$

In the following we will assume a connected and undirected graph. A static consensus algorithm is the following:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} (x_j - x_i)$$

where $\mathcal{N}_i$ is the set of neighbors of agent i. The whole system dynamics can be stated using the Laplacian matrix:

$$\dot{x} = -Lx$$

where $x = [x_1, ..., x_n]^T$. The convergence of this algorithm can be stated as follows

$$\lim_{t\to\infty} x_i(t) = \frac{1}{n}\sum_{i=1}^{n} x(0)$$

and its proof uses properties of the Laplacian matrix of an undirected and connected graph, [8].
Tracking performance of dynamic algorithms depends on time variation of the input signal $z(t) = [z_1(t) \ldots z_n(t)]^T$. The dynamic algorithm proposed in [8] is capable of reaching consensus equilibrium on any signal that has a steady-state value (i.e. $Z(s)$ has all its poles in the left-half plane and at most one pole in zero). The resulting system dynamics are given by:

$$\dot{x} = -Lx + \dot{z} \quad (13)$$

and consensus is achieved when

$$\lim_{t\to\infty} (x_i(t) - \frac{1}{n}\sum_j z_j(t)) = 0.$$

The LTI system can be described by the MIMO transfer function matrix:

$$H_{xz} = s(sI + L)^{-1}.$$

Using step references as commands to the swarm may often result in saturation of vehicle maneuvering capabilities; providing commands in terms of ramps instead of steps has several advantages; the most notable is that it reduces the actual tracking error, thus reducing the amount of control required by the decentralized autopilots, yielding better overall performance. Using a transfer function matrix of the form

$$H_{xz} = \frac{s^2}{(s+\alpha)^2}\left(\frac{s^2}{(s+\alpha)^2}I + L\right)^{-1}, \alpha > 0$$

it is possible to reach consensus state on ramp inputs. The Laplacian matrix of an undirected graph is symmetric and admits spectral decomposition

$$L = \sum_{i=1}^{n} \lambda_i P_i$$

where λ_i are the eigenvalues of L and P_i orthogonal projections onto mutually orthogonal eigenspaces. For a connected graph the following conditions are satisfied:

(1) $\lambda_1 = 0$
(2) $P_1 = \frac{1}{n}11^T$
(3) $\lambda_i > 0,\ \forall i > 1$

The input to error transfer function can be derived as:

$$E(s) = X(s) - \frac{1}{n}11^T Z(s) \to H_{ez} = H_{xz} - \frac{1}{n}11^T$$

using the above conditions, it is possible to write H_{ez} as:

$$H_{ez} = \sum_{i>1} \frac{s^2}{(1+\lambda_i)s^2 + 2\alpha\lambda_i s + \alpha^2\lambda_i} P_i$$

where all terms in the summation are stable. From the final value theorem $e(t) \to 0$ as $t \to \infty$, if $Z(s)$ is an arbitrary signal with at the most two poles in $s = 0$.
We can then derive the algorithm to run in each vehicle:

$$(1 + |\mathcal{N}_i|)\ddot{x}_i = \sum_{j\in\mathcal{N}_i} \ddot{x}_j + 2\alpha \sum_{j\in\mathcal{N}_i} (\dot{x}_j - \dot{x}_i) + \alpha^2 \sum_{j\in\mathcal{N}_i} (x_j - x_i) + \ddot{z} \quad (14)$$

the input signal is the vehicle's acceleration. From an implementation point of view the agents now need to transmit not only the consensus variable x but also $\dot{x}$ and $\ddot{x}$.

The improvement in estimation is shown in Figure 1 and Figure 2 with the comparison of the maximum and minimum estimation errors of the two first order moments (μ_x, μ_y) using consensus algorithm described by (13) (dotted line) and (14) (solid line). Simulations were performed using 25 vehicles with a fixed communication topology, where agent i is connected to agents $i-1$ and $i+1$ if $i \neq 1, n$; agents 1 and n had only one neighbor, agent 2 and $n-1$ respectively. The control law used is given in (16) and described in more detail in the next section.
The reference signal is a ramp on the center of mass position; second order moments were computed keeping a_1 and a_2 constant, and using (9).

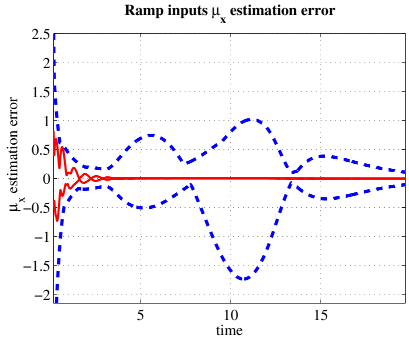

Fig. 1. Ramp inputs, μ_x minimum and maximum estimation error

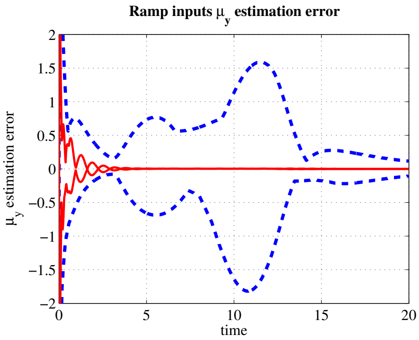

Fig. 2. Ramp inputs, μ_y minimum and maximum estimation error

5. CONTROL

The global control objective is defined by the minimization of a suitable potential function, $\Xi : \Re^{mn} \to \Re$, of the form:

$$\Xi = [f(p) - f^*]^T \Gamma [f(p) - f^*] \tag{15}$$

where $\Gamma \in \Re^{l \times l}$ is chosen such that $\Xi(p)$ has the unique global minimum in $f(p) = f^*$.
Each agent will follow the gradient of $\Xi(p)$, estimating the global moment vector $f(p)$ using consensus algorithms. The applied control law is the following:

$$\begin{aligned} \ddot{p}_i = -B_i\dot{p}_i - [D\phi(p_i)]^T \Delta_i [D\phi(p_i)]\dot{p}_i \\ - [D\phi(p_i)]^T \Gamma [x_i - f^*] \end{aligned} \tag{16}$$

where the first and second terms are damping terms and x_i is the estimate of the global moment vector $f(p)$ done by agent i and $D\phi(\cdot)$ defines the $l \times m$ Jacobian matrix of ϕ. The second damping term is needed to prove the boundness of the involved signals and will be discarded during simulations.
The consensus algorithm proposed in [7] is:

$$\begin{aligned} \dot{w}_i &= -\gamma w_i + \sum_{j \in \mathcal{N}_i} (x_j - x_i) \\ x_i &= w_i + \phi(p_i) \end{aligned} \tag{17}$$

where $\phi(p_i)$ is the vector moment generating function. Looking at the derivative of x the algorithm is the same as in (13), except for the forgetting factor γ that takes into account the events of an agent leaving the formation or joining of a new agent:

$$\dot{x} = \dot{w} + \dot{\phi}(p) = -\gamma w - Lx + \dot{\phi}(p)$$

where $z(t) = \phi(p(t))$.
The closed-loop system can be described in terms of $(p, \dot{p}, E)$, where $E = [e_1, \ \dots e_n]^T$ is the column vector whose elements are the estimate error of each agent. In [7] a proof of the boundness of each trajectory of system (16)-(17) is provided using two Lyapunov functions, V and U, and then summing them in a convenient way:

$$V(p, \dot{p}) = \dot{p}^T \dot{p} + n\Xi(p)$$

$$U(E) = Tr(EE^T) \tag{18}$$

$$\Psi(p, \dot{p}, E) = V + (1+\nu)U, \quad \nu > 0$$

The derivative of Ψ is non-increasing along trajectories in time, in fact:

$$\dot{\Psi} \le -\sum_{i=1}^{n} [\dot{p}_i^T [B_i + B_i^T] \dot{p}_i + \nu\epsilon |e_i|^2], \quad \epsilon > 0 \tag{19}$$

where the B_i terms had been chosen to verify $[B_i + B_i^T] > 0 \; \forall i$.

6. OBSTACLE AVOIDANCE

In order to avoid obstacles and collisions with other agents, gyroscopic and braking forces were added ([10],[11]). Gyroscopic forces are used as steering forces that curve the trajectories when an obstacle is in the detection shell of the agent. Steering forces alone are generally not sufficient for collision avoidance, thus braking forces terms are added when obstacles or other vehicles are in close proximity.
Each agent has its own *detection shell*, within which it can sense obstacles and neighboring agents. The detection shell will be part of the ball of radius r_s centered at the agent position; more precisely the area within the angles $(\angle\dot{p}_i - \alpha_s, \angle\dot{p}_i + \alpha_s)$. Reaction behaviors will depend on r_s and α_s. Agents only react to the nearest obstacle within their detection shell so, if there are no obstacle the gyroscopic and braking forces are set to zero. Gyroscopic forces are orthogonal to the velocity vector, while braking forces are damping terms acting in the opposite direction of the velocity vector. The resulting obstacle avoidance force is given by:

$$F_{OA} = F_G + F_B$$

The turning direction is chosen depending on the *approach angle*, i.e. the angle between the velocity vector of the agent and the vector from the agent to the nearest point of the obstacle (denoted as n_{va}); in order to have an escape behavior the gyroscopic force must produce an infinitesimal rotation around the vector $n_{va} \times \dot{p}_i$. This yields:

$$\begin{aligned} F_G &= S(n_{va}, \dot{p})\dot{p} \\ F_B &= -D(n_{va})\dot{p} \end{aligned} \tag{20}$$

where the matrix D is symmetric and positive-definite, and the matrix S is skew-symmetric. The latter implies:

$$< S(n_{va}, \dot{p}), \dot{p} >= 0 \ \Rightarrow F_G \ \dot{p} = 0$$

that is the force is in fact orthogonal to the velocity vector. The magnitude of the braking force is selected so as to generate infinite force as distance tends to zero.

The structure can be therefore described by a hybrid system with three states:

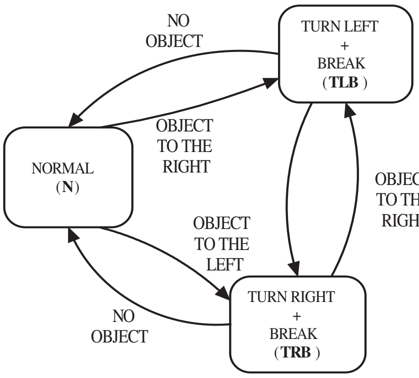

Fig. 3. Control system hybrid dynamics

- Normal (N)
- Turn Left and Brake (TLB)
- Turn Right and Brake (TRB)

as shown in Figure 3.

The control inputs for the three states are:

$$\begin{aligned} \ddot{p}_i &= -B_i\dot{p}_i + g(p_i, \dot{p}_i, x_i) \ (N) \\ \ddot{p}_i &= -B_i\dot{p}_i - D\dot{p}_i + S_L\dot{p}_i + g(p_i, \dot{p}_i, x_i) \ (TLB) \\ \ddot{p}_i &= -B_i\dot{p}_i - D\dot{p}_i + S_R\dot{p}_i + g(p_i, \dot{p}_i, x_i) \ (TRB) \end{aligned} \quad (21)$$

where $g(p_i, \dot{p}_i, x_i)$ is the sum of the gradiental and second damping term of equation (16).

Boundness of all signals can be found for each subsystem, as done in [7] for the *Normal* subsystem (equations (18) and (19) show this result). This is a necessary, but not sufficient, condition for the boundness of all signals of the resulting system subject to arbitrary switching. The problem can be solved using the Common Lyapunov Function approach. Recall the definition:

A positive definite continuously derivable (C^1) function $V : \Re^n \to \Re$ is a Common Lyapunov Function for the family of systems described by

$$\dot{x} = f_p(x), \quad x \in \Re^n, \quad p \in P$$

(where P is the index set of all subsystems) if there exists a continuous positive definite function $W(x) : \Re^n \to \Re$ for witch is verified

$$\frac{\partial V}{\partial x} f_p(x) \leq -W(x) < 0, \ \forall x \neq 0, \ \forall p \in P.$$

A sufficient condition for the boundness of all signals of the switched system is the existence of a Common Lyapunov Function. Finding one function for all subsystems gives us a measure of the loss of energy in a monotonic way; i.e. we can assure that during switching, the Lyapunov function is still decreasing.

From (21) and using the properties of matrices in (20):

$$D > 0 \qquad S_L + S_L^T = S_R + S_R^T = 0 \quad (22)$$

we can see the Lyapunov function in (18) as Common Lyapunov Function for the family of systems in (21). The derivatives of the Lyapunov function for the three subsystems are:

$$\dot{\Psi}_N \leq -\sum_{i=1}^{n} [\dot{p}_i^T [B_i + B_i^T] \dot{p}_i + \nu\epsilon |e_i|^2] = -W(p, \dot{p}, E) \quad (23)$$

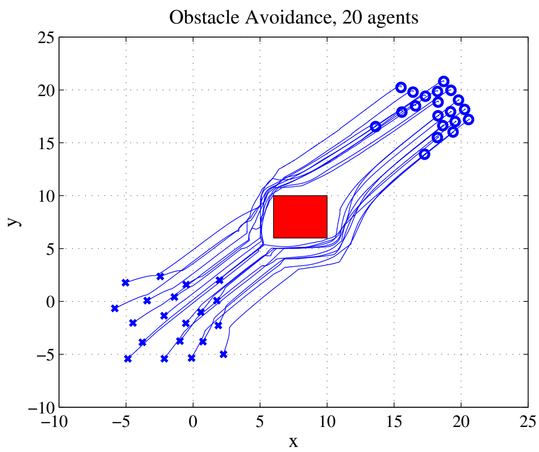

Fig. 4. Sample obstacle avoidance maneuver

$$\begin{aligned} \dot{\Psi}_{TLB} &= \dot{\Psi}_{TRB} \\ &\leq -\sum_{i=1}^{n} [\dot{p}_i^T [B_i + B_i^T + D + D^T] \dot{p}_i + \nu\epsilon |e_i|^2] \\ &\leq -W(p, \dot{p}, E) \end{aligned} \quad (24)$$

7. NUMERICAL RESULTS

A simulation was performed using 20 agents with a full connected topology. The objective of the simulation is to test the abstraction based approach with the added obstacle avoidance terms, instead of weighting the estimation performance. The control law and the consensus algorithm are defined in equations (16) and (17).
From the analysis of the time evolution of the global function $\Xi(p)$ (Figure 5) is possible to evaluate the behavior of the swarm. A larger value of this function corresponds to larger errors of the current moment vector. Figure 6 shows the tracking error history of the five absolute abstract variables ($\mu_x, \mu_y, M_{2x}, M_{xy}, M_{2y}$), and the evolution of the relative abstract variables (a_1, a_2).
From the numerical simulation, we can extract the following information:

- $t \leq 3s$ the system evolves in order to reach the desired statistics $f^* = [0\ 0\ 3\ 0\ 3]^T$;
- $3 < t \leq 6s$ the system is subjected to a ramp command on the center of mass and tracks it with an finite error; the errors on second order moments are still decreasing;
- $6 < t \leq 17s$ the agents are subjected to the obstacle avoidance forces; near the obstacle the second order moments are greater than desired (error is negative) to avoid the obstacle, while the motion of the center of mass slows down (error is positive). When leaving the obstacle region, the errors decrease;
- $17 < t \leq 21s$ the agents are subjected to a ramp command on the center of mass and the errors decrease;
- $t > 21s$ the system evolves in order to reach the desired statistics ($\mu = [17\ 17], a_1 = a_2 = 3, \theta = 0$).

Figure 7 shows the evolution of the minimum distance between any two agents: while approaching the obstacle it decreases, and increases when the swarm is leaving the obstacle.

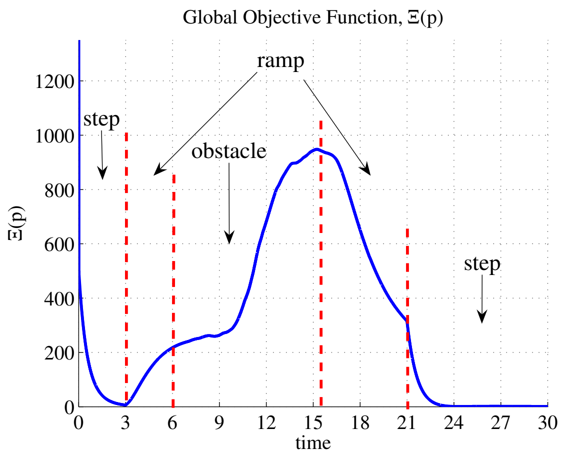

Fig. 5. Global Objective Function

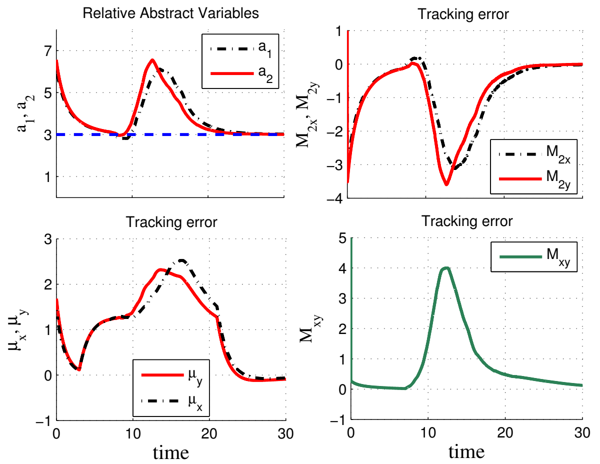

Fig. 6. Swarm abstract variables

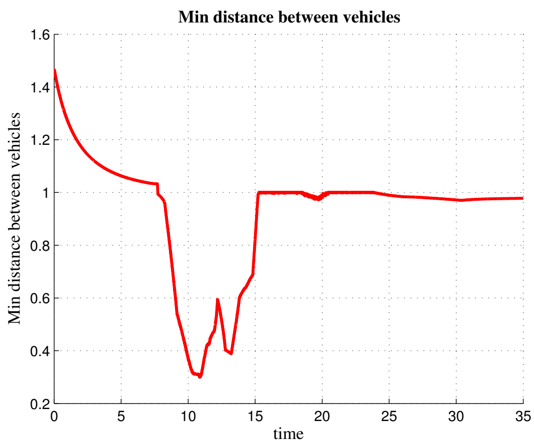

Fig. 7. Minimum distance amongst agents

8. CONCLUSIONS AND FUTURE WORK

The problem of decentralized control of swarms was addressed using an abstraction based approach. Summary statistics based on inertia moments were considered, and a general expression for calculating the absolute second order moments proposed. A new algorithm capable of reaching consensus on ramp inputs was proposed and convergence verified for undirected and connected communication/sensing topologies. Obstacle avoidance terms were added to the control law; boundness of all signals of the resulting switching systems was proved using a Common Lyapunov Function for all the subsystems.
Implementation of the proposed law on real vehicles poses several implementation issues; for instance: selection of sensors and communication systems, stability and performance in presence of sensor noise, communication occlusions due to obstacles or other vehicles, realistic vehicle dynamics etc. All of these are topics of current research and will be discussed in an upcoming paper [12].

REFERENCES

[1] Jaydev P. Desai, Jim Ostrowski, Vijay Kumar: *Modelling and Control of Formations of Nonholonomic Mobile Robots*, Transactions on Robotics and Automation, Vol. 17, No. 6, 2001.

[2] Veysel Gazi, Kevin M. Passino: *Stability Analysis of Swarms*,IEEE Transaction on Automatic Control, Vol. 48, No. 4, April 2003.

[3] Naomi E. Leonard, E. Fiorelli: *Virtual Leaders, Artificial Potentials and Coordinated Control of Groups*, CDC, 2001.

[4] Petter Ögren, Magnus Egerstedt, Xiaoming Hu: *A Control Lyapunov Function Approach to Multiagent Coordination*, Transactions on Robotics and Automation, Vol. 18, No. 5, 2002.

[5] Tucker Balch,Maria Hybinette: *Behavior-Based Coordination of Large-Scale Robot Formations*, IEEE, 2000.

[6] Reza Olfati-Saber, Richard M. Murray: *Flocking with Obstacle Avoidance: Cooperation with Limited Communication in Mobile Networks*, CDC 2003.

[7] R.A. Freeman, P. Yang, K.M. Lynch: *Distributed Estimation and Control of Swarm Formation Statistics*, American Control Conference, 2006.

[8] Demetri P. Spanos, Reza Olfati-Saber, Richard M. Murray: *Dynamic consensus on mobile networks* in Proceedings of the 2005 IFAC World Congress, Prague, 2005.

[9] C. Belta and V. Kumar: *Abstraction and control for groups of robots*, IEEE Transactions on Robotics, vol. 20, no. 5, pp. 865.875, Oct. 2004.

[10] Reza Olfati-Saber, Shawn C. Shadden, Jerrold E. Marsden, Dong Eui Chang: *Collision Avoidance for Multiple Agent Systems*, CDC December 2003.

[11] Dong Eui Chang, Jerrold E. Marsden: *Gyroscopic Forces and Collision Avoidance with Convex Obstacles*, New Trends in Nonlinear Dynamics and Control, and Their Applications, Springer, 2003.

[12] Lorenzo Pollini, Marta Niccolini, Mario Innocenti: *Experimental Evaluation of Decentralized Swarm Control Laws*, submitted to the 2008 AIAA Guidance Navigation and Control Conference.