Decentralized Cooperative Conflict Resolution for Multiple Nonholonomic Vehicles

E. Frazzoli*
University of California – Los Angeles, Los Angeles, CA 90095

L. Pallottino†, V. Scordio‡, and A. Bicchi§
University of Pisa, 56126 Pisa, Italy

In this paper, we consider the problem of collision-free motion planning for multiple nonholonomic planar vehicles. Each vehicle is capable of moving at constant speed along paths with bounded curvature, and is aware of the position and heading of other vehicles within a certain sensing radius. No other information exchange is required between vehicles. We propose a spatially decentralized, cooperative hybrid control policy that ensures safety for arbitrary numbers of vehicles. Furthermore, we show that under certain conditions, the policy avoids dead- and livelock, and eventually all vehicles reach their intended targets. Simulations and experimental results are presented and discussed.

I. Introduction

In this paper, we consider the problem of collision-free motion planning for a number of mobile agents evolving on the plane. Agents are modeled as nonholonomic vehicles, constrained to move at constant speed along path with a bounded curvature: such a model for the agent dynamics is very similar to the well-known model for car-like vehicles due to Dubins [1], with the only difference being that in our case the agents cannot vary their speed, and are therefore unable to stop. The environment in which the agents move is considered to be unbounded and free of obstacles. The agents are aware of the position and orientation of nearby agents, within a certain sensing or communication radius, but they do not have access to any other information. In particular, agents are not required to communicate explicitly their intentions or their objectives. However, all agents make decisions based on a common set of rules, decided a priori; since each agent can rely on the fact that other agents will follow the rules, we are aiming at the definition of a cooperative strategy for conflict avoidance and resolution.

The problem we are studying is motivated mainly by Air Traffic Control applications; in fact, the model of the vehicle dynamics we are using is representative of the motion of airliners during the enroute phase of their flight. Furthermore, since we require that only position and heading be communicated between agents, the algorithm can be implemented in practice using navigation data (e.g., from GPS sensors) and on-board transponders, with no direct input from the pilot. Other areas of application include manufacturing plants, automated factories, and intelligent transportation systems.

In recent years, the problem of safely coordinating the motion of several robots sharing the same environment has received a great deal of attention, both in robotics and in other application domains. A number of techniques have been developed for omni-directional (holonomic) robots, most of them requiring some form of central authority, either prioritizing robots off-line, or providing an online conflict-resolution mechanism, e.g., [2–4]; a characterization of Pareto-optimal solutions has been provided in [5]. Decentralized algorithms have appeared recently, e.g., [6, 7], for holonomic robots, and [8] for aircraft-like vehicles. The

*Assistant Professor, Mechanical and Aerospace Engineering Department, 420 Westwood Plaza, Los Angeles, CA 90095. AIAA Member.
†Researcher, Interdepartmental Research Center "E. Piaggio," Via Diotisalvi 2, 56126, Pisa (Italy).
‡Ph.D. candidate, Interdepartmental Research Center "E. Piaggio," Via Diotisalvi 2, 56126, Pisa (Italy).
§Professor, Interdepartmental Research Center "E. Piaggio," Via Diotisalvi 2, 56126, Pisa (Italy).

literature on flocking and formation flight, which has flourished recently (e.g., [9–11]), while ultimately leading to conflict-free collective motion, does not address individual objectives, and agents are not guaranteed to reach a pre-assigned individual destination.

In this paper, we wish to develop a control policy that is (i) spatially decentralized, and (ii) provably safe; we are also interested in pursuing a policy that is provably dead- and livelock-free, even though our results are more limited in this respect. Ideally, we aim at safety and liveness properties that are independent of the number of vehicles present in the environment; in addition, by relying only on limited information and local sensing/communication, we aim at ensuring scalability to systems composed by large numbers of vehicles. Our work builds on [6], in which the case of holonomic robots moving in an environment with stationary obstacles was considered: the authors introduced a spatially decentralized cooperative control scheme that guaranteed that no collisions would occur between robots, even when considering a limited sensing range. We present a control policy that is applicable to the non-holonomic vehicle case. In [6], no guarantees were given on the liveness of the policy, and in fact examples were presented that showed that the presence of obstacles, in a limited-information setting, would result in unavoidable deadlock conditions.

In the air traffic control literature, a control policy that shares some of the qualitative characteristics with the one we propose in this paper is the so-called roundabout technique introduced in [12]. The roundabout technique was proven safe for two- and three-aircraft conflicts, but it is not known to be safe for conflicts involving more than three aircraft [13]. Another approach, relying on the solution of Mixed-Integer Linear Programs (MILPs), and on the local exchange of information among "teams" of aircraft, was proven safe for encounters of up to five aircraft [14]. Both techniques relied on instantaneous direction changes on the part of the aircraft involved in the conflict. More recently, the safety of conflict resolution techniques using a model similar to the one we will use in this paper was proven in [15], using numerical techniques; however, scalability of such verification techniques to larger conflict is as yet uncertain. Remarkably, to the authors' best knowledge, no paper in the air traffic control literature has focused on the liveness issue, concentrating solely on proving the safety of the proposed policies.

The contributions of this paper can be summarized as follows: we introduce a novel spatially decentralized policy, which we will call *"generalized roundabout policy,"* that provides provably safe sensor-based motion planning for an arbitrary number of agents. We give sufficient conditions for the liveness of our policy for two vehicles, and give large-scale simulation examples suggesting that liveness properties extend to very complicated systems. The paper is organized as follows: In Section II we introduce some notation and define the problem we wish to address. In Section III we present the proposed generalized roundabout policy, and in Section IV we analyze its properties. In Section V we present and discuss some simulation results. Finally, in Section VI, we draw some conclusions and discuss some directions for future work.

II. Problem Formulation

Let us consider n mobile agents, able to move on the plane at constant speed, along paths with bounded curvature. For the sake of simplicity, and with no loss of generality, we will assume that both the agent's speed and the maximum curvature are unitary. Let the configuration of the i-th agent be specified by $g_i \in SE(2)$, the group of rigid body transformation on the plane. In coordinates, the configuration of the i-th agent is given by the triple $g_i = (x_i, y_i, \theta_i)$, where x_i and y_i specify the coordinates of a reference point on the agent's body with respect to an orthogonal fixed reference frame, and the heading θ_i is the angle formed by a longitudinal axis on the agent's body with the $y = 0$ axis.

Each agent enters the environment at the initial configuration $g_i(0) = g_{0,i} \in SE(2)$, and is assigned a target configuration $g_{\mathrm{f},i} \in SE(2)$. The agents move along a continuous path $g_i : \mathbb{R} \to SE(2)$ according to the model

$$\begin{aligned} \dot{x}_i(t) &= \cos(\theta_i(t)) \\ \dot{y}_i(t) &= \sin(\theta_i(t)) \\ \dot{\theta}_i(t) &= \omega_i(t) \end{aligned} \qquad (1)$$

where $\omega_i : \mathbb{R} \to [-1, 1]$ is a bounded signed curvature control signal.

Define the map $d : SE(2) \times SE(2) \to \mathbb{R}^+$ as the distance between the positions of two agents; in coordinates,

$$d(g_1, g_2) = \|(x_1, y_1) - (x_2, y_2)\|_2.$$

A *conflict* is said to occur at time t_c between the i-th and the j-th agents, if the agents are closer than a specified safety distance d_s, i.e., if $d(g_i(t_c), g_j(t_c)) < d_\text{s}$

A dynamic feedback control policy is a map $\pi_i : Z \times 2^{SE(2)} \to [-1, 1]$, $(z_i, \bar{g}) \mapsto \omega$ that associates to the i-th agent a control input, based on a set of locally-available *internal variables* $z_i \in Z$, and on the current configuration of the other agents in the environment. We use the shorthand $\bar{g} \subseteq \{g_1, \ldots, g_n\} \subset SE(2)$, to indicate a set of cardinality $\text{card}(\bar{g}) \leq n$, summarizing the available information about other agents. The policy π is said *spatially decentralized* if it is a function only of the configurations of agents that are within a given alert distance d_a from the computing agent; that is, we say that policy π is spatially decentralized if

$$\pi_i(z_i, \bar{g}) = \pi(z_i, \text{Neigh}(g_i, \bar{g}, d_\text{a})),$$

where the map Neigh extracts neighbors of g_i from $\bar{g}$, i.e., $d(g_i, g_\text{n}) \leq d_\text{a}$, $\forall g_\text{n} \in \text{Neigh}(g, \bar{g}, d_\text{a})$. Decentralized control policies, acting solely on locally available information, are attractive because of their scalability to large-scale systems, and of their robustness to single-point failures. However, since the agents act only on local information, global properties of a decentralized control policy are often hard to establish.

The objective of this paper is to design a spatially decentralized feedback control policy that satisfies, possibly under certain conditions to be specified, the following two properties:

- **Safety:** No conflicts are generated, i.e.,

$$\forall t > 0,\, i, j \in \{1, \ldots, n\},\, i \neq j : \, d(g_i(t), g_j(t)) \geq d_s. \tag{2}$$

- **Liveness:** At least one vehicle eventually reaches its destination:

$$\exists t_\text{f} \geq 0,\, i \in \{1, \ldots, n\} : \, g_i(t_\text{f}) = g_{\text{f},i}. \tag{3}$$

Note that if agents are removed from the environment upon arrival to their target (e.g., upon landing), the liveness condition stated above can be applied recursively, to ensure that all agents will eventually reach their targets.

III. The proposed motion coordination policy

In this section, we will propose a spatially decentralized feedback control policy. The policy is based on a number of discrete *modes of operation*, and as such the closed-loop system is a hybrid system; we will analyze its properties in Section IV. In order to introduce our control policy, we need to define some of its elements.

A. Reserved region

This policy is based on a concept of *reserved region*, over which each active agent claims exclusive ownership. Let the map $c : SE(2) \to \mathbb{R}^2$, $(x, y, \theta) \mapsto (x^c, y^c)$ associate to the configuration of an agent the center of the circle it would describe under the action of a constant control input $\omega = -1$. In other words,

$$(x^c, y^c) = c(x, y, \theta) = (x + \sin(\theta), y - \cos(\theta))\,;$$

refer to figure 1.

The reserved region for the i-th agent is defined as a disc of radius $1 + d_s/2$ centered at $c(g_i)$:

$$R_i(t) = \{(x, y) \in \mathbb{R}^2 : \|(x, y) - c(g_i(t))\|_2 \leq 1 + d_s/2\}. \tag{4}$$

The motion of the point (x_i^c, y_i^c) is described by the following equations:

$$\begin{aligned} \dot{x}_i^c(t) &= (1 + \omega_i(t)) \cos\theta_i(t) \\ \dot{y}_i^c(t) &= (1 + \omega_i(t)) \sin\theta_i(t). \end{aligned} \tag{5}$$

Our policy is based on the following basic observation: the model described by (1) and (5) is such that the reserved region (i) can be stopped at any time, by setting $\omega = -1$, and (ii) once stopped, it can be moved in any direction, provided one waits long enough for the heading θ to reach the appropriate value. As a consequence, for example, the center of the reserved region can follow any continuous path within an arbitrarily small tolerance, unlike model (1). Note that it is always possible to keep the reserved region fixed, with the corresponding agents moving along a minimum-radius circle entirely contained within it, see Figure 1.

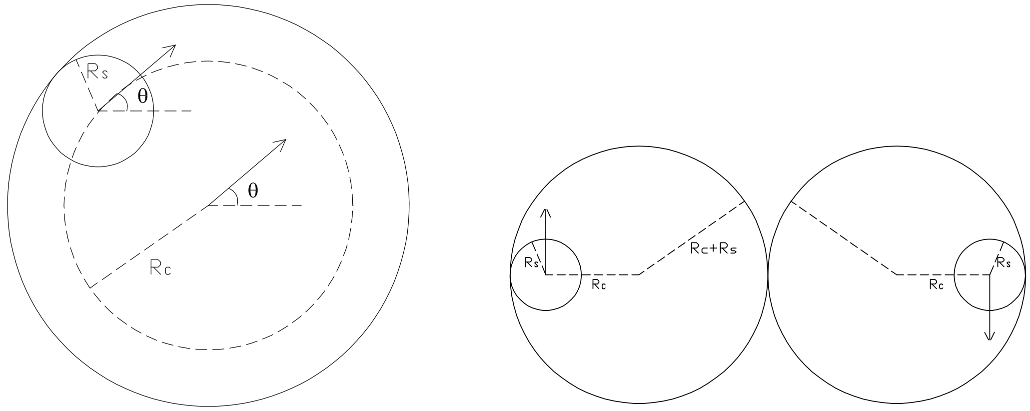

Figure 1. Left: The reserved region of a nonholonomic vehicle. Right: worst case scenario for the choice of the alert distance. In our case $R_C = 1$ and $R_S = \frac{d_s}{2}$.

B. Constraints

A sufficient condition to ensure safety is that the interiors of reserved regions are disjoint at all times; if such a condition is met, conflicts can be avoided if agents hold their reserved regions fixed, and move within them (by setting $\omega = -1$). As a consequence, each point of contact between reserved regions defines a constraint on further motion for both agents involved. More precisely, if the reserved region of agent i is in contact with the reserved regions of agents with indices in $J_i \subset \{1, \ldots, n\}$, the motion of the agents is constrained as follows

$$\dot{x}_i^c (x_i^c - x_j^c) + \dot{y}_i^c (y_i^c - y_j^c) \geq 0, \quad \forall j \in J_i. \tag{6}$$

In other words, the velocity of the i-th reserved region is constrained to remain in the convex cone determined by the intersection of a number of closed half-planes.

Note that the full set of constraints can be computed assuming that each agent is aware of the configuration of all agents within an alert distance $d_a = 4 + d_s$; refer to figure 1. In addition, the amount of information needed to compute the bound is uniformly bounded, independent from the total number of agents in the system: in fact, the maximum number of agents whose reserved region is in contact with the reserved region of the computing agents is six.

Let us define the set-valued map $\Theta : SE(2) \times 2^{SE(2)} \to 2^{S^1}$, associating to the configurations of an agent and of its neighbors the set of allowable directions in which the reserved region of the computing agent can translate without violating the constraints (6). For a connected, non-empty set $B \subset S^1$, $B \neq \emptyset$, let us define $\max(B)$ and $\min(B)$ as the elements on the boundary of B, respectively in the positive and negative direction with respect to the bisectrix of B. Finally, define the map $\Theta^-(g, \bar{g}) = \Theta(g, \bar{g}) \setminus \min(\Theta(g, \bar{g}))$. In other words, the output of Θ^- is an open set, obtained removing the boundary in the clockwise direction of the cone of feasible reserved region translations. Whenever Θ is a proper subset of S^1, $\max(\Theta)$, $\min(\Theta)$, and Θ^- are well defined. If $\Theta = \emptyset$, or $\Theta = S^1$, we set $\Theta^- = \Theta$.

C. Holding

As previously mentioned, setting $\omega = -1$ causes an immediate stop of an agent's reserved region's motion. We will say that when $\omega = 1$, the agent is in the `hold` state.

D. Right-turn-only steering policy

Our concept for decentralized conflict-free coordination is based on maintaining the interiors of reserved regions disjoint. Assuming that no constraints are violated, an agent will attempt to steer the center of its own reserved region towards the position it would assume at the target configuration. In a free environment,

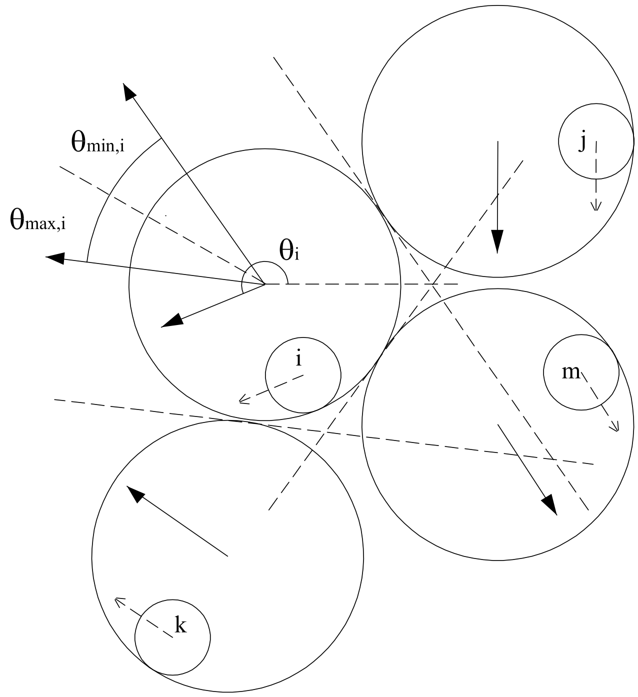

Figure 2. The set of allowable directions in which the center of the *i*-th reserved region can move, generated by the contact with the reserved regions of vehicles *j*, *m* and *k* respectively.

this can be accomplished switching between the `hold` state and a `straight` state:

$$\omega = \begin{cases} 0 & \text{if } \|\Delta_\mathrm{f}\|_2 > 0 \text{ and } \theta = \phi(\Delta_\mathrm{f}) \\ -1 & \text{otherwise} \end{cases} \tag{7}$$

where $\Delta_\mathrm{f} = c(g_\mathrm{f}) - c(g)$, and $\phi : \mathbb{R}^2 \setminus 0 \to S^1$ is a function returning the polar angle of a vector. Note that reserved region move along straight lines according to (7); clearly, such a policy is not optimal (in a minimum-time or minimum-length sense), but it does provide a simple feasible path for the agent from the current configuration to its target.

E. Rolling on a stationary neighboring reserved region

If the path of the reserved region to its position at the target is blocked by another reserved region, a possible course of action is represented by rolling in a pre-specified direction (in our case, the *positive* direction) on the boundary of the blocking region. Since in our setup agents communicate only information on their states, not on their future intentions, care must be exercised in such a way that the interiors of reserved regions remain disjoint

Let us start by assuming that the reserved region of the neighboring agent remains stationary; in order to roll on such region, without violating safety constraints, the control input must be set to

$$\omega = \begin{cases} (1 + d_s/2)^{-1} & \text{if } \Theta^-(g, \bar{g}) \neq \emptyset \text{ and } \theta = \max(\Theta) \\ -1 & \text{otherwise} \end{cases} \tag{8}$$

The above policy is obtained by switching between the `hold` state and a `roll` state; note that when in the `roll` state, the agent is not turning at the maximum rate.

Note that (8) also addresses the case in which the agent's motion is constrained by more than one contact with other agents' reserved regions. The only case in which the agent will not transition to the `roll` state, is the degenerate case in which Θ is a singleton, and Θ^- is empty.

F. Non-stationary neighbors

In general, the reserved region of an agent will not necessarily remain stationary while an agent is rolling on it. While it can be recognized that the interiors of the reserved regions of two or more agents executing (8) will always remain disjoint, it is possible that contact between two agents is lost unexpectedly (recall that the control input of other agents, their constraints, and their targets, are not available). In this case, we introduce a new state, which we call `roll2`, in which the agents turns in the positive direction at the maximum rate, i.e., $\omega = +1$, unless this violates the constraints. The rationale for such a behavior is to attempt to recover contact with the former neighbor, and to exploit the maximum turn rate when possible. The `roll2` state can only be entered if the previous state was `roll`.

G. Generalized Roundabout Policy

We are now ready to state our policy for cooperative, decentralized, conflict resolution; we call it Generalized Roundabout (GR) policy. The policy followed by each vehicle is based on four distinct *modes of operation*, each assigning a constant value to the control input ω. As a consequence, the closed-loop behavior of an individual agent can be modeled as a hybrid system.

We now introduce the hybrid system modeling the dynamics of a single agent. We define a hybrid system as a tuple

$$\mathcal{S} = (Q,\, X,\, U,\, \Phi, \Delta,\, \mathrm{Inv},\, \mathrm{Init}),$$

where Q is a set of discrete states, X is the continuous state space, U is a set of exogenous inputs, $\Phi : Q \times X \times U \to TX$ is a function describing the continuous dynamics of the system, Δ is a relation describing discrete transitions, and Inv, Init denote the invariant and initial conditions set, respectively. We refer the reader to the relevant literature for a more in-depth discussion of the hybrid systems formalism (e.g., [16–19] and references therein).

More in detail, the model for an individual agent can be specified as follows:

$$\begin{aligned}\mathcal{A} =& (\{\texttt{roll}, \texttt{roll2}, \texttt{hold}, \texttt{straight}\}, SE(2) \times \mathbb{R}, 2^{SE(2)},\\ & \Phi_{\mathrm{GR}}, \Delta_{\mathrm{GR}}, \mathrm{Inv}_{\mathrm{GR}}, \mathrm{Init}),\end{aligned}$$

- The discrete states $Q = \{\texttt{roll}, \texttt{roll2}, \texttt{hold}, \texttt{straight}\}$ correspond to constant inputs $\omega_{\texttt{roll}} = (1 + d_s/2)^{-1}$, $\omega_{\texttt{roll2}} = +1$, $\omega_{\texttt{hold}} = -1$, and $\omega_{\texttt{straight}} = 0$, respectively.
- $X = SE(2) \times \mathbb{R}$: in addition to its own configuration, each agent can keep track of time through a clock τ.
- $U = 2^{SE(2)}$: The exogenous input is a set $\bar{g} \subset SE(2)$ summarizing the available information about other agents. Since we are dealing with a decentralized policy, $\bar{g}$ can be restricted to contain solely neighbors within an alert distance $d_a = 4 + d_s$.
- The map Φ_{GR} is derived from (1), substituting the appropriate value for ω, based on the discrete mode, and by the clock rate $\dot{\tau} = 1$, i.e., it can be written in coordinates as follows:

$$\begin{aligned}\dot{x} &= \cos(\theta)\\ \dot{y} &= \sin(\theta)\\ \dot{\theta} &= \omega_q, \quad q \in Q\\ \dot{\tau} &= 1.\end{aligned} \tag{9}$$

- The initial set for each agent is unrestricted, i.e., the hybrid state $z = (q, x)$ can take any value in $Q \times X$.
- We do not explicitly write down the GR policy and its transition relations, guards, and invariants, but we refer the reader to Figure 3, which should provide the necessary detail in a clearer fashion.

The multiple-vehicle system we are considering is the parallel composition of n agents

$$\mathcal{S}_{\mathrm{GR}} = \mathcal{A}_1 | \mathcal{A}_2 | \ldots | \mathcal{A}_n \tag{10}$$

coupled through their configurations, communicated through the individual agents' exogenous inputs; $\mathcal{S}_{\mathrm{GR}}$ does not have exogenous inputs itself. (We do not define the operation of parallel composition here; see, e.g., [20] for details.)

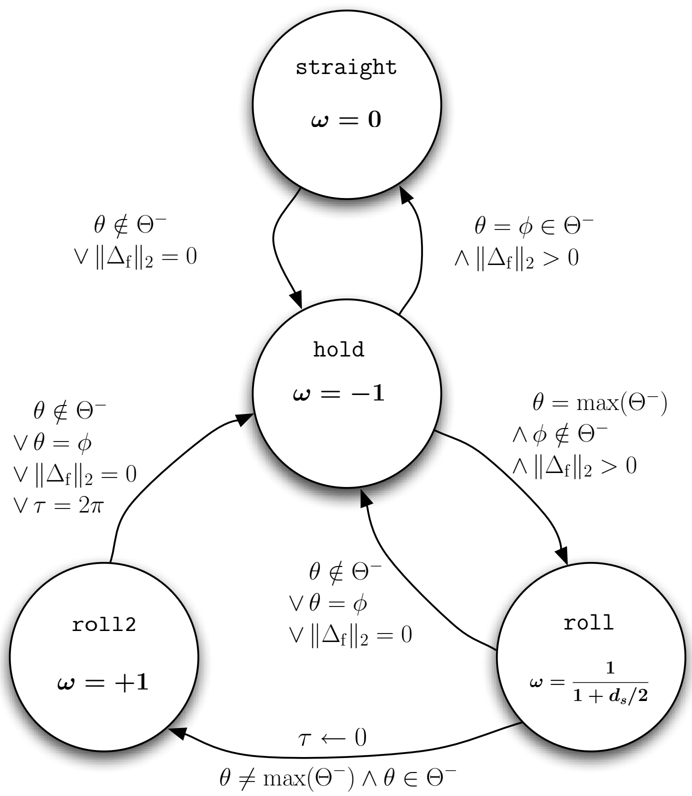

Figure 3. A hybrid automaton describing the Generalized Roundabout policy.

IV. Analysis

In this section, we will analyze the properties of the closed-loop hybrid system $\mathcal{S}_{\mathrm{GR}}$ defined in the previous section 10.

A. Well-posedness

The first step in our analysis of $\mathcal{S}_{\mathrm{GR}}$ is to verify that it is a well posed dynamical system, i.e., a solution exists and is unique, for all initial conditions within a given set. Indeed,

Theorem 1 *The hybrid system $\mathcal{S}_{\mathrm{GR}}$ is well posed, for all initial conditions in which the interiors of reserved disks are disjoint, i.e., $\|c(g_i) - c(g_j)\| \geq 2 + d_s$, $\forall i, j \in \{1, \dots, n\}$.*

Proof: The map (9) is globally Lipschitz in the state and in the control input; moreover, control inputs are constant within a discrete mode. The parallel composition of n copies of the continuous dynamics (9) is also globally Lipschitz. Hence, in order to establish well posedness of it is sufficient to show that there is no accumulation point of switching times, i.e., the number of switches in an open time interval is bounded, and the control input signal ω is piecewise continuous.

First of all, note that the number of instantaneous switches is bounded by three: the specification of invariant conditions in Figure 3 prevents infinite loops without time advancement. This can be verified by inspection of the invariants.

Let t_0 denote the time at which a switch in the discrete state has occurred, we need to show that there exists a $t' > t_0$ such that there are no switches in the open interval (t_0, t'). For simplicity, assume that the discrete state at time t_0 is the terminal state of the sequence of instantaneous switches occurring at t_0.

In the following, we will consider the i-th agent, and compute bounds on the time separation between switches, based on the current state of all agents. We have the following cases:

Case 1: $q_i(t_0) = \texttt{hold}$. A switch can be triggered by the following:

- $\Delta_{\mathrm{f},i} = 0$, $\theta_i = \theta_{\mathrm{f},i}$: The agent reaches its final configuration, and is removed from the system.
- $\Delta_{\mathrm{f},i=0} > 0$, $\theta_i = \phi_i \in \Theta_i^-$: the agent transitions to the discrete state `straight`.
- $\Delta_{\mathrm{f},i} > 0$, $\theta_i = \max(\Theta_i)$: the agent transitions to the discrete state `roll`.

None of the three above events can occur in the time interval $(t_0, t_0\delta_{1,i})$, with $\delta_{1,i} = \min\{\theta_{\mathrm{f},i} - \theta_i, \phi_i - \theta_i, \max(\Theta_i) - \theta_i\}$; the angle differences are meant to be counted in the direction of angular motion of the agent, modulo 2π.

Case 2: $q_i(t_0) = \texttt{straight}$. A switch can be triggered by the following:

- $\Delta_{\mathrm{f},i} = 0$: the reserved disk has been steered to its final configuration, and the agent transitions to the discrete state `hold`.
- $\phi_i \notin \Theta_i^-$: a new constraint on the motion of the reserved disk is activated, as the consequence of a contact with another agent's reserved disk

Neither of the two above events can occur in the time interval $(t_0, t_0+\delta_{2,i})$, with $\delta_{2,i} = \min\{\|\Delta_{\mathrm{f},i}\|_2, \min_{j\neq i}\{\|c(g_i) - c(g_j)\|_2 - (2 - d_s)\}\}$.

Case 3: $q_i(t_0) = \texttt{roll2}$. A switch can be triggered by events that have already been considered above, plus the time-out condition $\tau < 2\pi$. Hence no switches can occur in time interval $(t_0, t_0 + \delta_{3,i})$, where $\delta_{3,i} = \min\{\delta_1, \delta_2, 2\pi - \tau\}$.

CASE 4: $q_i(t_0) = \texttt{roll}$. This is the only delicate case, as instantaneous transitions can be triggered by other agents' actions. Let us indicate with j the index of the agent generating the constraint corresponding to $\max(\Theta_i)$. If $q_j = \texttt{hold}$, then the invariant $\theta_i = \max(\Theta_i)$ is preserved as the reserved disk of the i-th agent rolls on the reserved disk of the j-th agent; switches can be triggered by events considered above. If $q_j \neq \texttt{hold}$, the reserved disks of the two agents will detach at time zero—thus triggering a transition of the discrete state of the i-th agent to `roll2`; however, since the motion of the j-th agent is constrained by agent i, in such a way that the envelope of the reserved disk of agent j forms an angle $\chi_{ij} > 0$ (since Θ_j^- has been defined as an open set), the time at which the next switch can occur in this case is no sooner than $t_0 + 2\sin(\chi_{ij}/2)$. Hence, an additional switch cannot happen in the interval $(t_0, \delta_{4,i})$, with $\delta_{4,i} = \min\{\delta_2, (\phi_i - \theta_i)(1 + d_s/2), 2\sin(\chi_{ij}/2)\}$.

Summarizing, for the whole system, if t_0 is a switching time for at least one of the agents, no other agents can switch within the interval $(t_0, t_0 + \delta)$, where $\delta = \min_i\{\delta_{1,i}, \delta_{2,i}, \delta_{3,i}, \delta_{4,i}\} > 0$. ■

B. Safety

Theorem 2 *For all initial conditions for which the interiors of the agent's reserved disks are disjoint, i.e.,* $\|c(g_i) - c(g_j)\| \geq 2 + d_s$, $\forall i, j \in \{1, \ldots, n\}, i \neq j$, *the GR policy is safe, that is,* $\forall t \geq 0$, $d(g_i(t), g_j(t)) > d_s$, $\forall \in \{1, \ldots, n\}, j \neq i$.

Proof: The proof of the theorem follows directly from the fact that trajectories $g_i(t)$, $i = 1, \ldots, n$ are continuous functions of time. Moreover, within each state the feedback control policy has been chosen so that reserved discs never overlap: a transition is always enabled to the `hold` state, which stops the reserved disk istantaneously. Since the agents are always contained within their reserved disk, at a distance $d_s/2$ from its boundary, safety is ensured. ■

C. Liveness

We are still unable to provide a general result in terms of liveness for an arbitrary number of systems; however, we can now provide a sufficient condition on the target location for liveness in the simple case $n = 2$.

Theorem 3 *Consider two vehicles such that the center of the reserved disc in final configurations are at distance larger than* $2d_s + 4$. *The* GR *policy allows the vehicles to reach their final destinations in finite time, from all initial conditions such that the interiors of the reserved disks are disjoint.*

Proof: If the reserved disks of the two vehicles do not touch each other the two vehicles will reach their goal with the sequence of controls $\omega = -1$, $\omega = 0$, $\omega = -1$. Otherwise, when a contact between the reserved discs occurs, we have six different cases:

Case 1 $q_1 = q_2 = \texttt{straight}$.

Case 2 $q_1 = \texttt{straight}$, $q_2 = \texttt{hold}$.

Case 3 $q_1 = \texttt{straight}$, $q_2 = \texttt{roll}$.

Those cases, reported in figure 4 are such that the contact will be immediately lost. In the first case no other contacts will be generated and the goals will be reached with a sequence of transitions `straight`, `hold` for both vehicles. In the second case the first vehicle will reach its final destination with a sequence `straight`, `hold`, while the second one will maintains control `hold` until it is no longer blocked by the first vehicle, and can move towards its goal; the reserved disks will no longer touch. In the third case, the second agent will transition to the `roll2` state as soon as contact is lost. The reserved disks will not touch again. The second agent will reach its final destination with a sequence `roll2`,`hold`,`straight`,`hold`, or `roll2`,`straight`,`hold`, depending on the initial and final configurations.

Case 4 $q_1 = q_2 = \texttt{hold}$.

Case 5 $q_1 = \texttt{hold}$, $q_2 = \texttt{roll}$.

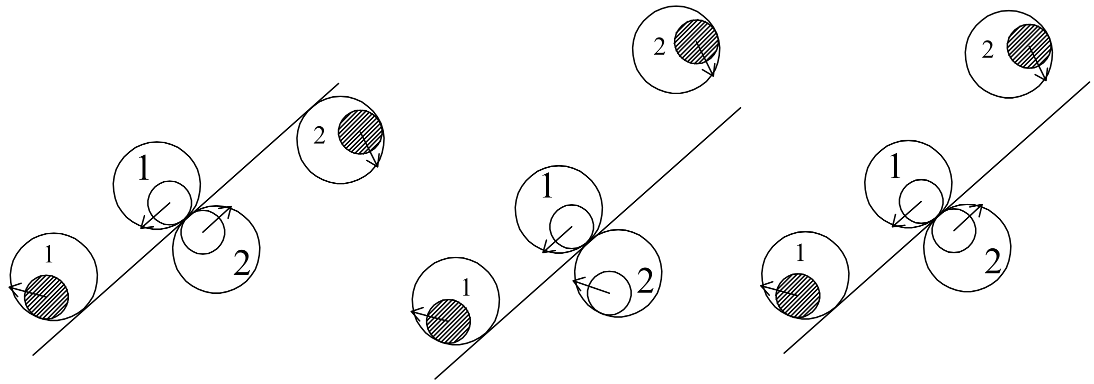

Figure 4. Three possible situation for two agents with reserved discs in contact, agent 1 is such that q_1 = `straight`. On the left q_2 = `straight`, in the middle q_2 = `hold`, on the right q_2 = `roll`

It is sufficient to discuss the second case, since if both vechicles are in state `hold` they will reach a configuration that is equivalent to the second case unless one of them can move through its final configuration without contacts of the reserved disks. If this occurs, one of the vehicles will be in state `straight` and this is the case 2 discussed above.

In the second case the second vehicle will turn on the left so that the second reserved discs will slide along the first one until one of the two vehicle are able to move through the goal or they reach the configuration of case 6 (that will be discussed below).

Case 6 $q_1 = q_2 =$ `roll`.

In this case the contact will be lost immediately, and both vehicles will switch to `roll2`; reserved disks may touch again. If a new contact occurs, the point of contact between the reserved discs has moved counterclockwise in the first vehicle's frame and clockwise on the second one. After this new contact both vehicles are in the `hold` state. If one of the vehicle can move through its final configuration by switching to the `straight` state, the configuration is equivalent to Case 2. Otherwise this procedure is repeated. But after enough time if the distance between target configurations is larger than $2d_s + 4$, one of the two vehicles will be able to move through its goal since at least one of the goals is not covered by the cluster movements. In this case for one vehicle $\omega = 0$ and the configuration is equivalent to one of the previous cases.

A similar proof can be provided for the three-vehicle case, but is not reported here since it does not provide any additional insight into the problem, and is quite laborious. Unfortunately, we are still unable to develop an inductive step to extend the result to a general number of agents, or provide a proof that is valid in the general case. Current work is aimed at addressing the issue.

V. Simulation and experimental results

This section provides some simulation results, highlighting how the proposed policy works. Furthermore, some details on experimental results on a testbed using small computer-controlled cars are reported.

A. Simulations

For the simulations, we start considering a case with seven agents that have to navigate from an initial to a final configuration, avoiding collisions. Initial configurations form a cluster in which the reserved disks

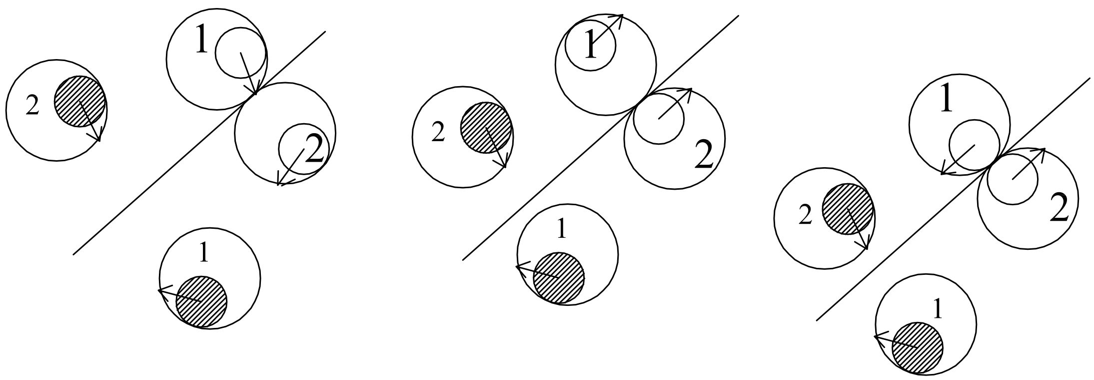

Figure 5. Three possible situation for two agents with reserved discs in contact, on the left and in the middle, agent 1 is such that $q1 = \texttt{hold}$, while agent 2 $q_2 = \texttt{hold}$ and $q_2 = \texttt{roll}$. On the right $q_1 = q_2 = \texttt{roll}$.

of six agents are tangent to the seventh one, while final configurations lie on a circumference. The overall trajectories simulating this particular case are plotted on the bottom right of figure 6.

Starting from initial configuration, after an initial transient, during which agents are far from each other and can navigate according to the "right-turn-only" strategy, some reserved regions come into contact. The most important events in the simulation are shown in figure 6 and reported below.

1. Referring to figure 6 top-left: for agents 1 , 2 , 3 , 4 , 5 and 7, `hold` is active since their relative final configurations do not belong to relative sets Θ^- and their headings are such that $\theta_i \neq \theta_{\max,i}$. For vehicle 6 `roll` is active since its final configuration is not contained into its own Θ^- set, and $\theta_6 = \theta_{\max,6}$.

2. Referring to figure 6 top-right: no agent is able to move toward the relative final configuration. More precisely, agents 2 ,3 ,5 ,7 are in `hold`, while agents 1 , 4 and 6 are in `roll`.

3. Referring to figure 6 bottom-left: Agents 1 , 3 , 4 and 5 execute `hold`. For Agents 2 and 6 `roll` is active. Notice that agent 7 is the only agent that can move toward its target since its final configuration is contained in the set of its admissible directions.

We tested our algorithm for a large number of agents, in different configuration, and results have been successful with respect to safety and liveness. Obviously cluster configuration are the most challenging from the traffic congestion point of view. For this reason we report cases of 19 and 37 agents represented in figures 7 and 8 respectively. As it can be seen from the figures[a], all agents reach their final targets in finite time despite the initial congestion.

B. Experimental results

The proposed policy has also been tested on an experimental testbed including three Super Perfection Micro RC Machine© (see figure 9) which admit four discrete inputs: Forward, Reverse, Left, Right. For our purposes only Forward, Left and Right inputs have been used.

Agents are identified by 2 inch by 4 inch labels, each having a different number of black dots to identify the specific car (see figure 9). A single Logitech QuickCam Sphere© camera is used to detect the labels and compute the position and orientation of each vehicle.

[a] Animations of the agents' motion in the mentioned scenarios will be made accessible from the following URL: `http://rigoletto.seas.ucla.edu/download/gnc05`.

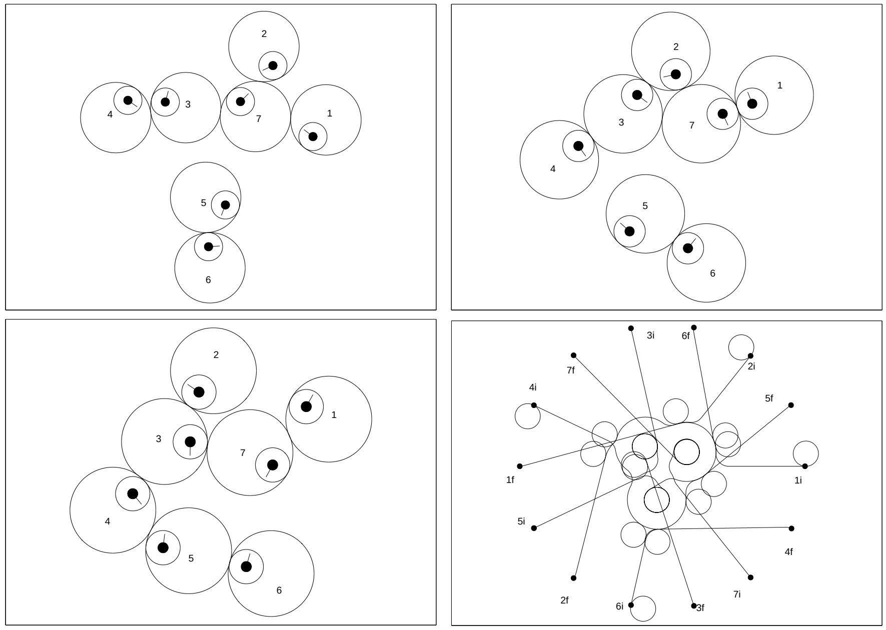

Figure 6. Significant instants and whole agents trajectories (down-right) of a simulation with seven agents)

In our experiments, a single server is used for all the three cars; the server has access to the configuration of all vehicles through the camera. In order to simulate a decentralized scenario, for each vehicle i the server uses the information on position and orientation of any car within a certain distance from i. It then computes the control based on this partial information and sends it back to agent i.

Experiments replicate the results found in simulations, within small errors due to the sensing and actuation systems. Some frames of an experiment video are reported for reader convenience in figure 10[b]

VI. Conclusions and future work

In this paper, we have outlined a novel spatially decentralized, cooperative policy for conflict-free motion coordination of non-holonomic vehicles. The policy gives rise to a hybrid system, which can be shown to be well posed, and safe, if the initial conditions satisfy a rather non-restrictive (but possibly conservative) condition. Moreover, we showed simulation results with large numbers of vehicles in a congested environment, which not only confirm safety of the policy, but also suggest its liveness, in the sense that all agents reached their target in finite time. Note that the number of agents in our example is much larger than the number of agents that other existing algorithms can handle maintaining safety guarantees. Finally, all of the computations involved in the proposed policy are spatially decentralized, and their complexity is bounded regardless of the number of agents, thus making the policy scalable to large-scale systems. Current work is aimed at determining conditions formally guaranteeing the liveness of the proposed control law for an arbitrary number of aircraft.

Acknowledgments

This research was partially supported by NSF grant 0133869, EC grants IST 2001-37170 "RECSYS", IST-004536 "RUNES", FP6-IST-511368 NoE "HYCON", and MIUR grant FIRB RBAU01RY47. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the supporting organizations.

[b]A video of the experiment will be made accessible from the following URL: `http://rigoletto.seas.ucla.edu/download/gnc05`.

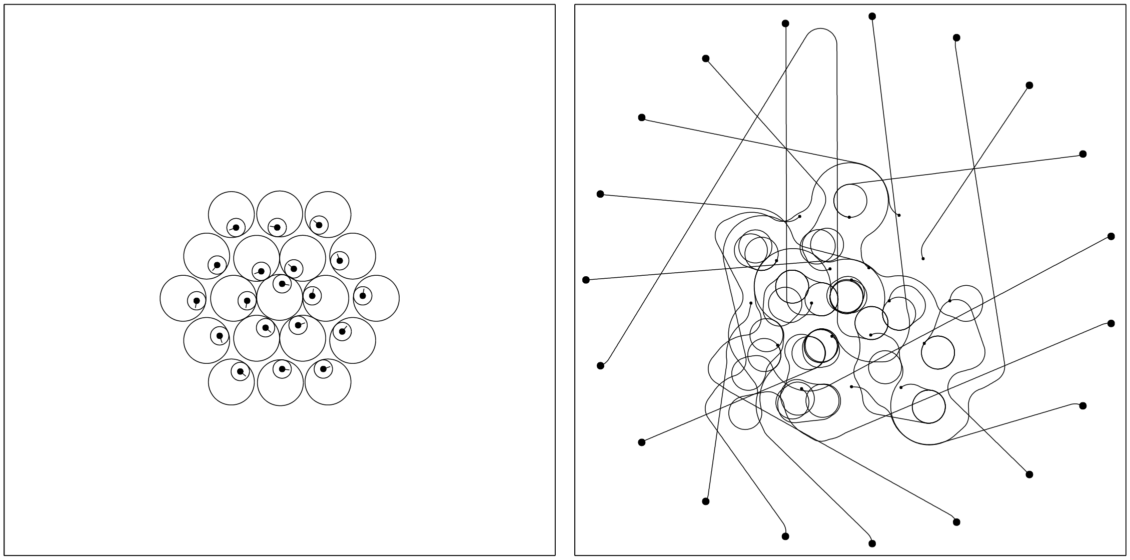

Figure 7. Evolution of nineteen vehicles (right) starting from the cluster configuration (left).

References

[1]Dubins, L. E., "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, Vol. 79, 1957, pp. 497–516.

[2]LaValle, S. M. and Hutchinson, S. A., "Optimal motion planning for multiple robots having independent goals," *IEEE Trans. on Robotics and Automation*, Vol. 14, No. 6, 1998, pp. 912–925.

[3]Peng, J. and Akella, S., "Coordinating multiple robots with kinodynamic constraints along specified paths," *Proc. Fifth International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Nice, France, 2002.

[4]Gerkey, B. P. and Matariĉ, M. J., "Sold!: Auction methods for multi-robot coordination," *IEEE Trans. Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 758–768.

[5]Chitsaz, H. and J. M. O'Kane, S. M. L., "Exact Pareto-Optimal Coordination of Two Translating Polygonal Robots on an Acyclic Roadmap," *Proc. IEEE Conf. on Robotics and Automation*, 2004.

[6]Lumelsky, V. J. and Harinarayan, K. R., "Decentralized motion planning for multiple mobile robots: the cocktail party model," *Autonomous Robots*, Vol. 4, No. 1, 1997, pp. 121–35.

[7]Klavins, E., "Communication Complexity of Multi-Robot Systems," *Proc. Fifth International Workshop on the Algorithmic Foundations of Robotics*, Nice, France, 2002.

[8]Inalhan, G., Stipanovic, D. M., and Tomlin, C. J., "Decentralized Optimization, with Application to Multiple Aircraft Coordination," *Proc. IEEE Conf. on Decision and Control*, 2002.

[9]Jadbabaie, A., Lin, J., and Morse, A. S., "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Aut. Control*, Vol. 48, No. 6, June 2003, pp. 988–1001.

[10]Tanner, H., Jadbabaie, A., and Pappas, G. J., "Flocking agents with varying interconnection topology," Submitted to Automatica.

[11]Olfati-Saber, R., "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," Submitted to the IEEE Trans. on Automatic Control (Technical Report CIT-CDS 2004-005).

[12]Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems," *IEEE Trans. Aut. Control*, Vol. 43, 1998, pp. 509–521.

[13]Ghosh, R. and Tomlin, C. J., "Maneuver design for multiple aircraft conflict resolution," *Proc. American Control Conf.*, Chicago, IL, 2000.

[14]Pallottino, L., Scordio, V., and Bicchi, A., "Decentralized Cooperative Conflict Resolution Among Multiple Autonomous Mobile Agents," *Proc. IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, 2004.

[15]Tomlin, C., Mitchell, I., and Ghosh, R., "Safety Verificaiton of Conflict Resolution Maneuvers," *IEEE Trans. Intelligent Transportation Systems*, Vol. 2, No. 2, 2001.

[16]Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S., "The Algorithmic Analysis of hybrid systems," *Theoretical Computer Science*, Vol. 138, 1995, pp. 3–34.

[17]Branicky, M. S., *Studies in Hybrid Systems: Modeling, Analysis, and Control*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[18]Lygeros, J., *Hierarchical Hybrid Control of Large Scale systems*, Ph.D. thesis, University of California, Berkeley, CA, 1996.

[19]Alur, R., Dang, T., Esposito, J., Hur, Y., Ivančić, F., Kumar, V., Lee, I., Mishra, P., Pappas, G. J., , and Sokolsky, O., "Hierarchical modeling and analysis of embedded systems," *Proceedings of the IEEE*, Vol. 91, No. 1, 2003, pp. 11–28.

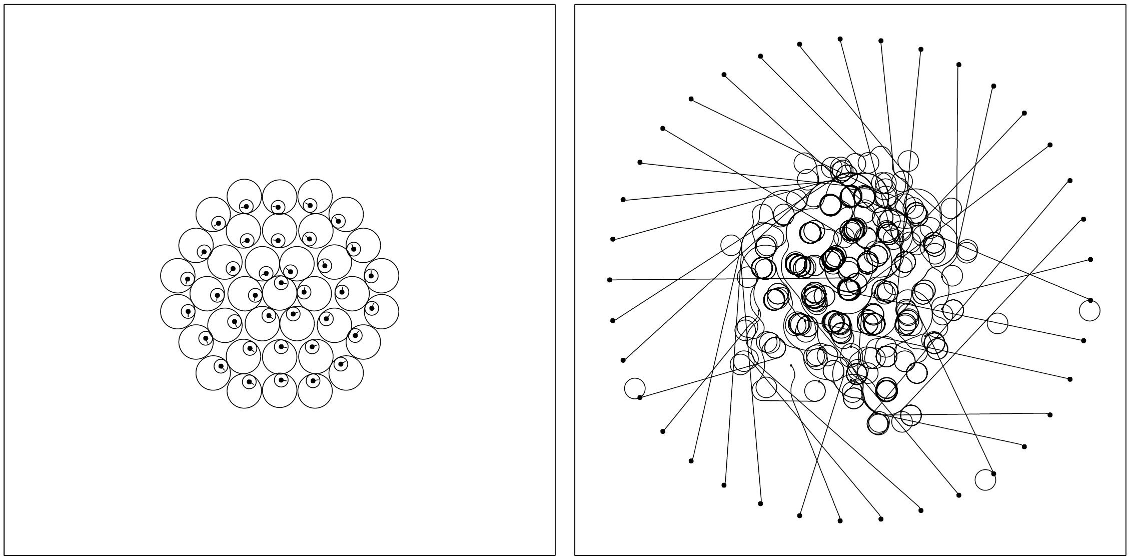

Figure 8. Evolution of 37 vehicles (right) starting from the cluster configuration (left).

[20]Lynch, N., Segala, R., and Vandraager, F., "Hybrid I/O Automata Revisited," *Hybrid Systems IV: Computation and Control*, edited by M. Di Benedetto and A. Sangiovanni-Vincentelli, Vol. 2034 of *Lecture Notes in Computer Science*, Springer-Verlag, 2001, pp. 403–417.

Figure 9. Cars used in the experiments with an example of label.

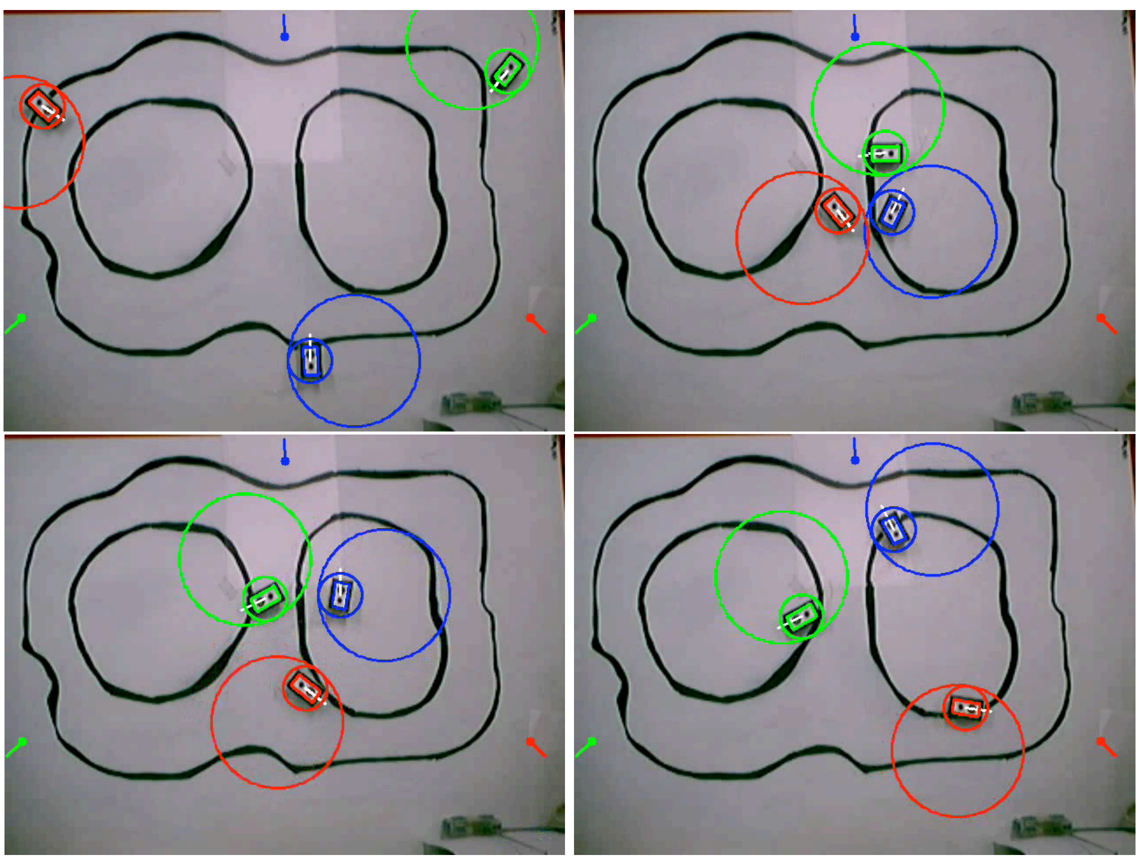

Figure 10. Experimental results on a three vehicles system, each vehicle has a label for the camera recognition. Both safety and reserved disc for each vehicle are plotted. In top left the vehicle initial configurations is represented while final configurations are the coloured dots and segments.