Decentralized Hybrid Control for Multi-Agent Motion Planning and Coordination in Polygonal Environments

Mason Sutorius * Dimitra Panagou**

* Robotics Institute, University of Michigan, Ann Arbor, MI, USA (e-mail: sutorimb@umich.edu)
** Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA (e-mail: danagou@umich.edu)

Abstract: This paper presents a decentralized hybrid control scheme for the motion planning and coordination of teams of mobile agents in known obstacle environments with both convex and non-convex obstacles. A mathematical analysis using tools from switched systems theory is carried out to establish the convergence of the system trajectories under certain modeling assumptions on the surrounding environment. The design resolves a class of deadlock situations arising in earlier work, and allows for a wider class of obstacles (both convex and non-convex) to be considered in the environment. Simulation results demonstrate the efficacy of the algorithm.

Keywords: Autonomous Mobile Robots; Decentralized Control and Systems; Multi-vehicle systems; Trajectory and Path Planning; Autonomous Vehicles

1. INTRODUCTION

Planning and control for multi-agent systems, e.g., mobile robots, is a popular research topic with a plethora of methodologies that this paper can not cite in its entirety. As a common ground, centralized approaches often offer stronger performance, yet they demand increased computation time as the number of agents increases, while requiring communication between all agents and a centralized controller. As an alternative, decentralization is advantageous in terms of flexibility, robustness, and computational efficiency, and can be applied to various multi-agent control problems, including consensus, formation control, collision avoidance, navigation among waypoints, deployment/coverage, see Cao et al. (2013); Yao and Gupta (2011); Zelazo et al. (2012); Pei et al. (2013); Schwager et al. (2011); Jin et al. (2015); Roussos and Kyriakopoulos (2013) for some indicative approaches. In addition, decentralized multi-agent systems in which the agents rely only on local sensing to gather information on their surroundings (other agents, physical obstacles) eliminate the need for network communications, which are often subject to drop-outs and delays, and thus might be advantageous in terms of realistic implementation.

Collision avoidance is of top priority in motion planning and coordination problems, and becomes particularly challenging in environments with multiple agents and static physical obstacles. Various methods to solve the decentralized navigation problem exist, and can be broadly classified as prescribed Lalish et al. (2008); Ferrera et al. (2013); Hwang et al. (2007); Jin et al. (2015), optimization-based Camponogara et al. (2002); Ferrari-Trecate et al. (2009), or reactive (potential or vector-field generated) methods Panagou (2014); Hernández-Martínez and Aranda-Bricaire (2011); Dimarogonas et al. (2006), while more recently decentralized controllers that satisfy linear temporal specifications as well as connectivity constraints have received increased interest Guo et al. (2014).

In this paper we consider the decentralized motion planning and coordination for multiple mobile agents in a known obstacle environment, and propose a solution within a hybrid control framework. The problem is defined as designing a hybrid control system that drives the agents to their respective goal locations while avoiding collisions with each other and with static polygonal objects. The motivation behind this work lies in the synthesis of local controllers that satisfy the global objective (convergence to the goal destinations) and the local objective (collision avoidance) in a scalable, computationally-efficient manner. In general, obtaining stability guarantees for hybrid multi-agent systems is a nontrivial task, as the identification of Zeno trajectories, i.e., of trajectories leading to switching among modes infinitely often, is highly dependent on the structure of the system, i.e., on the adopted dynamics, modes, and transitions. Here we are primarily driven by our recent work in Hegde and Panagou (2016), where a class of vector fields is constructed to steer each agent towards its goal location while avoiding known static obstacles. The extension of this work to the multi-agent case has been done using a decentralized nonlinear model predictive control (NMPC) technique to account for the inter-agent safe separation constraints, under the assumption that inter-agent communication is not available, i.e., that agents have local sensing information only. However, NMPC is computationally expensive, especially for high-dimensional nonlinear systems, thus not always applicable.
to realistic setups. Hence, synthesizing local reactive controllers into a hybrid system offers merits from both the theoretical and the practical point-of-view.

In Ferrera et al. (2013), the authors define a hybrid system of five modes to ensure the collision-free motion for a team of $N$ agents. Their algorithm yields promising results for concave obstacles, but deadlock situations arise when object contours are convex, or the environment is such that agents are not able to surround each other (see Fig. 1).

Thus, we seek to extend the prior work of Ferrera et al. (2013) to enable collision avoidance for a group of $N$ agents in an known environment with both convex and non-convex static obstacles, while establishing convergence properties under certain assumptions. Specifically, this paper addresses the problem of convergence in an 2-D environment that contains both convex and concave obstacles, without inter-agent communication. We design a hybrid system utilizing the modes in Ferrera et al. (2013) and implement a high-level convex contour detection algorithm to ensure convergence and collision avoidance.

The paper is organized as follows: Section 2 gives the problem formulation and Section 3 describes the hybrid collision avoidance algorithm. Section 4 provides simulation results, and Section 5 summarizes the methodology and conclusions of the work.

2. PROBLEM FORMULATION

Given a team of $N$ agents in a known 2-D environment, we seek to drive each agent to its goal location while avoiding collisions with static polygonal obstacles and other agents. Let $\mathcal{O}_s = \{v_i | i \in \{1, \ldots, v_N\}\}$ be the set of vertices of all of the $M$ static polygonal obstacles and $v_N$ be the total number of vertices or corners. Let $\mathcal{O}_e$ be the set of edges of all of the $M$ static polygonal obstacles.

Consider a group of $N$ agents with positions $r_i(t) = [x_i(t), y_i(t)]^T$, $i \in N$ and orientations $\theta_i(t) \in [0, 2\pi]$. Let $r_i(0) = [x_i(0), y_i(0)]^T$ and $\theta_i = [x_{gi}, y_{gi}]^T$ be the initial and goal position of agent $i$, respectively. Each agent $i$ has a sensing region $\mathcal{S}_i$ of radius $R_{si}$ centered at $r_i(t)$ given as

$$\mathcal{S}_i = \{r \in \mathbb{R}^2 \mid ||r - r_i|| < R_{si}\},$$

in which it can detect polygonal obstacles and other agents to avoid collisions. Let $\mathcal{C}_i$ be the set of objects in the sensing region $\mathcal{S}_i$ of agent $i$ such that $\mathcal{C}_i = \{j \in N : j \neq i, ||r_j - r_i|| \leq R_{si}\} \cup \{j \in \mathcal{O}_s : ||v_j - r_i|| \leq R_{si}\} \cup \{i, j \in \mathcal{O}_e : ||(v_i - v_j) - r_i|| \leq R_{si}\}$. We assume that:

- The geometry of the environment is known.
- Each agent can locate itself in the global environment.
- Each agent is circular with radius $\rho_i$, and its motion is modeled using unicycle kinematics:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ 0 \end{bmatrix} \Rightarrow \dot{q}_i = f(q_i, \nu_i), \quad (1)$$

where $q_i = [r_i^T, \theta_i]^T = [x_i, y_i, \theta_i]^T$ is the state vector of agent $i$, comprising position coordinates and orientation relative to the global frame $\mathcal{G}$, and $\nu_i = [u_i, \omega_i]^T$ is the vector of control inputs comprising the linear velocity $u_i$ and the angular velocity $\omega_i$ of the agent $i$ about the $z\mathcal{G}$ axis.

- The agents can be heterogeneous in size $\rho_i$, maximum capable speed $V_{max,i}$, and sensing radius $R_{si}$; however, the linear speed limit $k_{ui}$ of each agent $i$ must adhere to the physical limitations of the entire team:

$$k_{ui} \leq \min_{j \in \mathcal{C}_i} \frac{R_{si} + R_{sj} - \rho_i - \rho_j}{\frac{dT}{dT}}.$$  \quad (2)

These constraints ensure that agents detect potential collisions with other agents and obstacles before the collisions can occur.

- An agent can malfunction during operation, causing loss of mobility. However, the agent cannot malfunction within a set threshold distance around the goal position of some other agent, or within a set threshold distance of a static obstacle so that it blocks a feasible path, or causes a deadlock situation.

- No inter-agent communication exists; each agent knows only the positions of neighboring agents and obstacles within its sensing region $R_{si}$.

The adopted modeling for an agent is depicted in Fig. 2.

3. HYBRID COLLISION AVOIDANCE ALGORITHM

The collision avoidance problem described above is solved using a hybrid approach. The overall architecture comprises a high-level convex edge-detection to resolve deadlock situations caused by convex object contours, and a low-level reactive mode-based algorithm that ensures collision avoidance between agents.
3.1 Reactive Algorithm for Collision Avoidance

We build upon the decentralized collision avoidance algorithm for a team of $N$ agents in Ferrera et al. (2013). The motion of each agent is governed by a hybrid system comprising five modes. Let us define the following terms:

\[
\text{Goal Reached: } \begin{cases}
\|r_i - r_{gi}\| < \theta_{th} \\
\text{Free: } N_{Ai} \neq \emptyset \\
\text{Blocked: } |\theta_i - \phi_{ai}| \leq \theta_{th} \\
\text{Rendezvous: } \phi_{ai} \in C_i \\
\text{Recontre: } \phi_{ai} \notin C_i
\end{cases}
\]

**Navigable Area $N_{Ai}$:** Let $L_{ij}$ be a line that passes through the intersection points of the boundaries of the sensing regions $S_i$ and $S_j$ (see Fig. 4). For a given agent $i$, the semi-plane defined by $L_{ij}$ that contains element $j \in C_i$ is the forbidden area $F_{Ai}$, for agent $i$ w.r.t. agent $j$. After agent $i$ processes all of the objects within $C_i$, the total forbidden area is defined as $F_{Ai} = \bigcup_{j \in C_i} F_{Ai}$, and the navigable area $N_{Ai}$ is the remaining area. If $N_{Ai}$ is infinite in size, the area is open (see also Fig. 5).

**Avoidance Heading $\phi_{ai}$:** A heading that will allow agent $i$ to surround all agents $j \in C_i$ in the counter-clockwise direction. If $r_{gi} \notin N_{Ai}$, then no such avoidance heading $\phi_{ai}$ exists. For each line $L_{ij}$, two avoidance headings exist which are parallel to the line. The agent evaluates each line and selects the avoidance heading that (1) drives the agent towards the navigable area (2) enables the agent to surround the obstacles within its sensing region in the counter-clockwise direction.

In what follows we describe the five modes of the hybrid system as defined in Ferrera et al. (2013), while the logic governing the transitions among modes is shown in Fig. 3. The modes are described below and depicted in Fig. 5.

**Free Mode:** If $C_i \neq \emptyset$, then the agent is in the free mode and moves towards the goal location $r_{gi}$ under the control law:

\[
u_i = k_{ui} \tanh(|r_i - r_{gi}|), \quad \omega_i = -k_{wi}(\theta_i - \phi_i),
\]

where $\phi_i$ is the line-of-sight between the agent and the goal location, i.e., the angle between the $x_G$ axis of the global frame and the vector $r_{gi} - r_i$; see also Fig. 2.

**Recontre Mode:** If $C_i \neq \emptyset$, and $r_{gi} \in N_{Ai}$, then an avoidance heading $\phi_{ai}$ is commanded. If $|\theta_i - \phi_{ai}| > \theta_{th}$, then the agent is in the reconitre mode and immediately stops linear movement to avoid collision. The agent rotates until its heading is within $\theta_{th}$ of $\phi_{ai}$ with the following control law:

\[
u_i = 0, \quad \omega_i = -k_{ai}(\theta_i - \phi_{ai}).
\]

**Rendezvous Mode:** If $C_i \neq \emptyset$, and $r_{gi} \notin N_{Ai}$, then an avoidance heading $\phi_{ai}$ is commanded. If $|\theta_i - \phi_{ai}| \leq \theta_{th}$, then the agent is in the rendezvous mode, and acts with

\[
u_i = k_{ui} \tanh(|r_i - r_{gi}|), \quad \omega_i = -k_{wi}(\theta_i - \phi_{ai}).
\]

**Blocked Mode:** If $C_i \neq \emptyset$, and $r_{gi} \notin N_{Ai}$, then the agent is in the blocked mode, and the control law is given as:

\[
u_i = 0, \quad \omega_i = 0.
\]

This system guarantees collision-free movements and that agents will reach their respective goal locations provided that: (i) Agents do not malfunction within a set distance of the goal location of another agent. (ii) The goal location of
the agents do not block the path of other agents. (iii) The obstacles are concave. (iv) The environment is constructed such that agents are able to surround each other. See Figure 1 for a depiction of a counterexample.

3.2 Convex Contour Detection & Deadlock Resolution

We extend the system in Ferrera et al. (2013) to include a high-level controller to resolve deadlock situations caused by convex obstacle contours. When a convex obstacle corner is within the attractive-to-the-goal heading of the agent, an alternate waypoint is provided to escape this deadlock situation. When the situation is resolved, the original goal location is provided to the agent.

Mathematically, let $A = [v_{c-1} - r_{gi}; v_{c+1} - r_{gi}]$, and $B = [v_{c-1} - v_c; v_{c+1} - v_c]$, where $v_c$, $v_{c-1}$, and $v_{c+1}$ are the coordinates of the convex corner, the next corner in the clockwise direction, and the next corner in the counterclockwise direction, w.r.t. the center of the obstacle. Let $\mathcal{R}_s = \{r_{ai} \in \mathbb{R}^2 \mid \|r_{ai} - r_i\| < \rho_i\}$ be the area in which the robot is located. If $\exists r_{ai} \in \mathcal{R}_s | A \cdot r_{ai} > 0, B \cdot r_{ai} > 0, A \cdot v_c > 0$, where $\cdot$ is a component-wise inequality, and the agent is in the restricted free mode, then a new waypoint is given as

$$r_{gi}^* = R(\psi)R_{ai} \frac{v_{c+1} - v_c}{\|v_{c+1} - v_c\|} + v_{c+1},$$

where $R$ is a rotation matrix. This situation is depicted in Figure 6. When $\exists r_{ai} \in \mathcal{R}_s | A \cdot r_i \neq 0, B \cdot r_i \neq 0$, the original goal location $r_{gi}$ is given to the agent. If a convex contour causes a deadlock for agent $i$ and $r_{gi}$, then another alternate waypoint $r_{gi}^*$ is passed to the agent. This process occurs recursively, until the latter deadlock situation is resolved. It should be noted that the waypoint $r_{gi}^*$ can be located anywhere within the environment-it may lie inside or on the boundary of other agents or static obstacles, or even within a set distance of an agents goal location.

3.3 Stability Analysis of the Hybrid System

The proposed hybrid control policy gives rise to switched closed-loop system dynamics for each agent, described as:

$$\dot{q}_i(t) = f_k(q_i(t)), \quad (7)$$

where $f_k(\cdot)$ the vector equation denoting the closed-loop system dynamics under the $k$-th mode, $k \in \mathcal{K} = \{1, 2, 3, 4, 5, 6\}$.

For analyzing the stability of the closed-loop system, we follow the results in Branicky (1998); Zhao and Hill (2008). Let us consider a strictly increasing sequence of times:

$$T = \{t_0, t_1, \ldots, t_n, \ldots\}, \quad n \in \mathbb{N},$$

the set: $I(T) = \bigcup_{t \in \mathbb{N}} [t_2n, t_{2n+1}]$ denoting the interval completion of the sequence $T$, and the switching sequence:

$$\Sigma = \{q_0; (k_0, t_0), (k_1, t_1), \ldots, (k_n, t_n), \ldots\},$$

where $t_0$ is the initial time, $q_0$ is the initial mode, $\mathbb{N}$ is the set of nonnegative integers and $k_n \in \mathcal{K}$. The switching sequence $\Sigma$ along with (7) completely describes the trajectory of the switched system according to the following rule: $(k_p, t_p)$ means that the system evolves accordingly to:

$$\dot{q}_{kp}(t) = f_{kp}(q_{kp}(t), t)$$

for $t_p \leq t < t_{p+1}$. Equivalently, for $t \in [t_p, t_{p+1})$ one has that the $k_p$-th subsystem is active. We assume that the switching sequence is minimal in the sense that $k_p \neq k_{p+1}$. For any $k \in \mathcal{K}$, denote:

$$\Sigma_p \{ k \} = \{ t_{k_1}, t_{k_1+1}, t_{k_2}, t_{k_2+1}, \ldots, t_{k_n}, t_{k_n+1}, \ldots \}$$

the sequence of switching times when the $k$-th subsystem is switched on or switched off. Hence, $I(\Sigma | k)$ is the set of times that the $k$-th subsystem is active.

Denote $E(T) = \{ t_0, t_2, \ldots, t_n, \ldots \}$ the even sequence of $T$; then, $E(\Sigma | k) = \{ t_{k_1}, t_{k_2}, \ldots, t_{k_n}, \ldots \}$ denotes the sequence of the switched on times of the $k$-th subsystem.

Theorem 1: (Zhao and Hill (2008)) Consider the switched system (7) and assume that for each $k \in \mathcal{K}$ there exists a positive definite generalized Lyapunov-like function $V_k(x)$ w.r.t. $f_k(x(t))$ and the associated trajectory. Then:

(i) The origin of the switched system is stable if and only if there exist class $\mathcal{K}$ functions $\alpha_k(\cdot)$ satisfying

$$V_k(q(t_{km+1}))) - V_k(q(t_{km})) \leq \alpha_k(||q_0||), \quad (8)$$

$m \geq 1, \ k = 1, 2, \ldots, K$.

(ii) The origin of the switched system is asymptotically stable if and only if (8) holds and there exists $k$ such that $V_k(q(t_{km+1}) \to 0$ as $m \to \infty$.

The theorem essentially states that stability of the switched system is ensured as $V_k(q(t_{km+1}))) - V_k(q(t_{km}))$, which is the change of $V_k$ between any “switched on” time $t_{km+1}$ and the first active time $t_{km}$, is bounded by a class $\mathcal{K}$ function, regardless of where $V_k(q(t_{km}))$ is. With this result at hand, the stability of the proposed hybrid control scheme under the stated assumptions is now straightforward:

Theorem 2: The position trajectories $r(t)$ of the switched system $\dot{q} = f_k(q)$, where $k \in \mathcal{K}$, under the proposed switching logic, are asymptotically stable.

Proof: The correctness of the proposed theorem can be verified by a direct application of Theorem 1. For each
subsystem $k \in \mathcal{K}$, consider the generalized Lyapunov-like function $V_k(r) = \|r - r_{gi}\|$. At any “switched on” time instant $t_{kn}$ with $n > 1$, (that is, for any “switched on” time instant after the first switch has occurred at $t_{k1}$), one has that $V_k(r(t_{kn})) \leq V_k(r(t_{k1}))$, for $k \in \{1, 2, 3, 4, 5\}$, i.e., the agent is either moving towards the goal or stationary. Also, when in mode $k = 6$, one has $V_k(r(t_{kn})) - V_k(r(t_{k1})) \leq \alpha_k(\|r_0\|)$, that is, the distance to the goal location $r_{gi}$ might grow while in mode 6 due to the motion around the obstacle, with the class $\alpha(\cdot)$ function dependent on the geometry of the obstacle, yet any growth of each $V_k$ is always bounded (assuming that the obstacles are compact sets). Hence the switched system is stable. Furthermore, note that for the modes $k = 3$ and $k = 5$ where there is no decrease of the generalized Lyapunov function $V_k$ during the “switched on” time intervals, i.e., for the modes where the linear velocity is $u_i = 0$, it holds that:

(1) Each agent $i$ stays in mode $k = 5$ (blocked mode) for finite amount of time, i.e., even if agent $i$ is temporarily blocked by surrounding agents $j \neq i$, the agents $j$ will be in different modes and hence move towards their goal locations, freeing space around agent $i$ and therefore causing him switch to a different mode; this is due to the assumption on the sparsity of the goal locations.

(2) Each agent $i$ stays in mode $k = 3$ (rendezvous mode) for finite amount of time, i.e., until the orientation error $\theta_i - \phi_{oi}$ drops below the prescribed threshold, after which it switches to mode $k = 4$ (rendezvous mode), where the linear velocity $u_i$ is non-zero.

(3) After exiting mode $k = 6$ the agent switches to mode $k = 2$ (free mode) until it reaches its goal location (goal reached mode $k = 1$), or until it encounters other obstacles/agents (mode $k = 3$ and then mode $k = 4$).

(4) Under the assumption on the sparsity of the goal locations, mode $k = 4$ eventually leads to switching to mode $k = 1$. Hence the position trajectories are asymptotically stable.

4. SIMULATION RESULTS

We validate the efficacy of the proposed hybrid motion planning and coordination algorithm through multiple simulations while varying the number, size, and shape, of the obstacles. A simulation scenario involving $N = 10$ agents is presented in the sequel. The system is discretized with a sampling time of $T_s = 0.01s$. The sensing radius $R_s$ and agent radius $\rho$ were selected as 0.064 and 0.02, with a maximum physical velocity $v_0$ of 1.2. Algorithm 1 was implemented in MATLAB to validate the system. $d_{th}$ and $\theta_{ch}$ were selected as 0.01 and $1^\circ$, respectively.

The initial deployment of the agents in the environment, along with the goal locations are depicted in Fig. 7, where initial locations are denoted by ‘+’, and goal locations by ‘*’. The resulting paths are shown in Fig. 8, while Fig. 9 shows the mode each agent is in over time. Fig. 10 illustrates the minimum distance between agents over time, demonstrating that the inter-agent distance is always greater than $\rho_i + \rho_j$, i.e., that the agents do not collide. All the modes are activated at some point in time for at least one agent, and the convex edge-detection is activated for this given agent as well, as demonstrated in Fig. 9. It is worth noticing that agent 3 stays in the “blocked mode” for a finite amount of time only (cyan color).

5. CONCLUSION

We presented a hybrid motion planning and coordination algorithm for teams of mobile agents operating in known obstacle environments with both convex and non-convex obstacles, along with certain stability guarantees. A mathematical analysis using tools from switched systems theory was carried out to establish the convergence of the system trajectories under the adopted modeling assumptions on the surrounding environment. The design resolves a class of deadlock situations arising in earlier work, and allows
FIG. 10. Minimum distance between any two agents.

Algorithm 1 Motion Planning for Dynamic Agents

Input: \( u_i, \omega_i \)

Output: \( u_i, \omega_i \)

1: Initialization : \( O_e, O_i, r_{gi}, S_i \)
2: LOOP For all \( N \) agents
3: while \( r_i \neq r_{ig} \) do
4: Calculate \( N_A \)
5: if \( A \cdot r_i > 0 \) and \( B \cdot v_i > 0 \) and \( A \cdot v_i > 0 \) then
6: \( r_{gi} = r_{gi} \)
7: end if
8: if \( r_{gi} \in N_A \) then
9: Free Mode
10: Calculate \( u_i, \omega_i \) using (3)
11: else if \( N_A = \emptyset \) then
12: Blocked Mode
13: Calculate \( u_i, \omega_i \) using (6)
14: else
15: Calculate \( \phi_{ai} \)
16: if \( |\theta_i - \phi_{ai}| \leq \theta_{th} \) then
17: Rendezvous Mode
18: Calculate \( u_i, \omega_i \) using (5)
19: else
20: Rencontre Mode
21: Calculate \( u_i, \omega_i \) using (4)
22: end if
23: end if
24: end while
25: Goal Reached Mode

for a wider class of obstacles to be considered. Our current work focuses on improving the proposed design by establishing less conservative assumptions on the goal locations of the agents, resolving thus additional possible deadlocks.

REFERENCES

