

MIT Open Access Articles

*Locally-optimal multi-robot navigation under
delaying disturbances using homotopy constraints*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

Citation: Gregoire, Jean, Michal Čáp, and Emilio Frazzoli. "Locally-Optimal Multi-Robot Navigation Under Delaying Disturbances Using Homotopy Constraints." *Autonomous Robots* 42, no. 4 (December 14, 2017): 895–907. doi:10.1007/s10514-017-9673-6.

As Published: <http://dx.doi.org/10.1007/s10514-017-9673-6>

Publisher: Springer US

Persistent URL: <http://hdl.handle.net/1721.1/116601>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Locally-Optimal Multi-Robot Navigation under Delaying Disturbances using Homotopy Constraints

Jean Gregoire* · Michal Cap* · Emilio Frazzoli

Received: date / Accepted: date

Abstract We study the problem of reliable motion coordination strategies for teams of mobile robots when any of the robots can be temporarily stopped by an exogenous disturbance at any time. We assume that an arbitrary multi-robot planner initially provides coordinated trajectories computed without considering such disturbances. We are interested in designing a control strategy that handles delaying disturbance such that collisions and deadlocks are provably avoided, and the travel time is minimized. The problem is analyzed in a coordination space framework, in which each dimension represents the position of a single robot along its planned trajectory. We demonstrate that to avoid deadlocks, the trajectory of the system in the coordination space must be homotopic to the trajectory corresponding to the planned solution. We propose a controller that abides this homotopy constraint while minimizing the travel time. Besides being provably deadlock-free, our experiments show that travel time is significantly smaller with our method than with a reactive method.

Keywords autonomous robots · planning · coordination · control · homotopy classes

*Equal contribution

J. Gregoire
MIT Senseable City Lab
MIT, Boston, USA.
E-mail: jmmg@mit.edu

M. Cap
Artificial Intelligence Center
Czech Technical University, Prague, Czech republic.
E-mail: michal.cap@agents.fel.cvut.cz

E. Frazzoli
Dep. of Mechanical and Process Eng.
ETH Zurich, Zurich, Switzerland.
E-mail: emilio.frazzoli@idsc.mavt.ethz.ch

1 Introduction

The problem of coordinating multiple mobile robots sharing a common environment finds numerous applications including, e.g. warehouse managements systems (Guizzo, 2008) or autonomous urban transportation systems (Dresner and Stone, 2008). There exist two main paradigms to tackle this complex problem: the *reactive paradigm* – robots continuously sense their environment and react according to what they sense – and the *deliberative paradigm* – robots first sense and compute a plan, and then execute the computed plan.

In the *reactive paradigm*, the robot follows the shortest path to its current destination, monitors positions and velocities of other robots in its vicinity, and attempts to resolve collision situations as they appear, locally.

Many reactive control strategies have been proposed within the last three decades. Among the first ones has been the cocktail party model (Lumelsky and Harinarayan, 1997), while more recently, the techniques based on velocity-obstacle paradigm have become popular (Van den Berg et al, 2008; Guy et al, 2009; Alonso-Mora et al, 2014; Van Den Berg et al, 2011). The key benefit of the reactive approaches is their computational efficiency. As noticed in (Lumelsky and Harinarayan, 1997; Pallottino et al, 2007), their main drawback is the inability to provably avoid deadlocks, especially in crowded and cluttered environments.

In the *deliberative paradigm*, coordinated collision-free trajectories from the origin to the destination of each robot are computed before execution. Even the simplest formulations of multi-robot motion planning problem are however known to be NP-hard (Spirakis and Yap, 1984). Therefore, heuristics such as as prioritized motion planning (Erdmann and Lozano-Pérez, 1987; Čáp et al, 2015a) are often used in practice to compute the coordinated trajectories. Curiously, if the operational environment is appropriately structured,

prioritized techniques are known to always provide a solution in polynomial time (Čáp et al, 2015b,a).

This work focuses on the coordination of *labeled* robots, i.e., each robot is given a specific goal to reach. For completeness, we should mention that there exists an alternative *unlabeled* formulation, where the task is to navigate a set interchangeable robots to a set of goal location such that all locations are occupied. The complexity of the unlabeled version of the problem has been studied by Solovey and Halperin (2016) and efficient solution algorithms for this formulation have been also developed, see, e.g., Turpin et al (2014) and Solovey et al (2015).

The key benefit of deliberative planning is that once a solution is found and precisely executed, all robots are guaranteed to reach their destination in finite time, i.e. they will not get stuck in a deadlock. It is important to note that these guarantees only apply if the robots execute the planned trajectories precisely both in space and in time.

However, in environments where the robots share space with humans, such as warehouses or roads, precise temporal execution of the planned trajectory is often impossible to ensure. For example, a robot in a warehouse must yield to a human crossing its path, even if its planned trajectory dictates to move forward at that instant. Even a small disturbance that delays a robot may have a strong impact on the system. See the example scenario in Figure 1: If one of the robots is delayed, and the other simply continues advancing along its trajectory, the robots will end up in a deadlock.

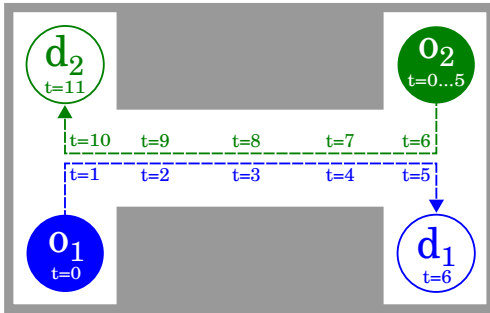


Fig. 1 Deadlock example induced by a delaying disturbance. Robots 1 desires to move from origin o_1 to destination d_1 , the second robot desires to move from o_2 to d_2 . A possible solution found by a multi-robot trajectory planner is indicated by the dashed line, where labels $t = \dots$ denote the time evolution. The plan dictates that the robot 1 goes through the corridor first. Robot 2 waits until robot 1 leaves the corridor to proceed through the corridor towards its destination. Now suppose that robot 1 is subject to a disturbance and enters the corridor with a delay of 5 time units. If robot 2 continues moving according to the original plan, both robots engage in a heads-on collision or a deadlock, depending on whether some low-level safety mechanism prevents the robots from physically crashing, in the center of the corridor.

To retain the benefits of the deliberative approaches while being reactive to unexpected events, hybrid deliberative/reactive architectures have been proposed. For example, Kowshik

et al. designed a hybrid controller (Kowshik et al, 2011) that computes a plan for all robots and updates it constantly as new information comes in. However, the above method needs to replan every time an unexpected event impacts the progress of any of the robots.

In this paper, we show that deadlock avoidance can be achieved without replanning, by designing a reactive controller that ensures the path of the system in the coordination space remains in the homotopy class of the path corresponding to the planned multi-robot trajectory. Therefore, our approach should not be seen as an alternative to deliberative methods but rather as a complementary technique allowing to reliably handle disturbance while executing the plan output by a multi-robot planner.

Previous work in motion planning already noticed that planning a set of homotopic paths allows to react to unexpected events by taking advantage of the freedom of action within the homotopy class (see, e.g. the work on elastic strips in (Quinlan and Khatib, 1993; Brock and Khatib, 2002)). In the context of multi-robot coordination, homotopy classes have been studied by Ghrist and Lavelle (Ghrist et al, 2005; Ghrist and Lavelle, 2006) to design a locally-optimal trajectory planner, deforming any trajectory into a locally Pareto optimal trajectory. Gregoire et al. proved that these homotopy classes can be encoded by so-called priority graphs (Gregoire, 2014). In contrast to the above work, the proposed method serves as an additional layer. It takes as input coordinated trajectories planned by an arbitrary multi-robot motion planner, and it derives a homotopy class of safe and deadlock-free solutions. The system is then reactively controlled to remain in this homotopy class.

The contribution of this paper is twofold. Firstly, we show that coordinated trajectories computed by an arbitrary multi-robot motion planner can be used to derive a maximal set of homotopic solutions – a homotopy class – by translating the problem into the coordination space. The set of solutions is composed of all paths in the coordination space that are homotopic to the diagonal path δ . Given coordinated trajectories in the physical space, we derive a whole region χ^δ in the coordination space which is both collision-free and deadlock-free. Robots can safely navigate within this region with the guarantee to reach their goal at the end. To our knowledge, there is no previous work in multi-robot coordination that, given a particular solution, derives a maximal set of local solutions around the initial solution (this maximal set of local solutions is the derived homotopy class).

The second contribution of the paper is a controller that guarantees the system remains within χ^δ , i.e. it remains within the derived maximal set of local solutions, despite any delaying disturbance affecting the advancement of individual robots, while minimizing delay. The application of the proposed controller allows to guarantee that all robots will eventually reach their destination even when some of the robots

are temporarily stopped by a disturbance. It is remarkable that a mathematical property – the existence of homotopy classes in the coordination space – can be exploited to design a controller that provides a robustness property at the system level, i.e. the ability to handle unpredicted disturbances while retaining the liveness of the system. Interestingly, as the homotopy class is a maximal set of local solutions, we have the guarantee that to perform better than our controller, there is no alternative but to replan, i.e., to switch to an alternative homotopy class of solutions. In other words, our controller does everything that can be done to handle disturbance without replanning. It must be noted that under major disturbances, the system may perform much better by replanning, but at the cost of a high computational complexity : replanning should be considered as the last alternative.

This article is a mathematical extension of the conference paper (Čáp et al, 2016). The conference paper was mainly focused on the algorithmic aspect of the proposed controller, while this article focuses on the construction of the homotopy constraint and how this mathematical property can be used to design a controller allowing to navigate within the constructed homotopy class. It is structured as follows. Section 2 shows how to use initial coordinated trajectories computed by an arbitrary planner to obtain a planned homotopy class. Section 3 then provides a controller ensuring locally optimal navigation within a planned homotopy class. Finally, Section 4 evaluates our control scheme through simulations and Section 5 concludes the paper.

2 From planned coordinated trajectories to a homotopy constraint in the coordination space

In this section we formulate the problem in the coordination space and show several structural properties of paths in the coordination space. Firstly, we show that in the coordination space the homotopy class of the planned path can be equivalently defined as a class of paths taking values in a well-defined subregion of the coordination space χ^δ . Secondly, we show that if the system stays within the region χ^δ , then it is guaranteed to avoid deadlocks and eventually reach the goal. Intuitively, the homotopy class should be seen as a set of solutions to the coordination problem such that the relative order between robots with respect to each intersection area remains the same as initially planned. It seems intuitive that keeping the same relative order, i.e. staying within the homotopy class, is a deadlock-free strategy. This intuition is proved by the following results, and homotopy theory gives their underlying mathematical origin.

2.1 Problem formulation in the coordination space

Consider 2-d or 3-d workspace $\mathcal{W} \subseteq \mathbb{R}^d$, with $d \in \{2, 3\}$, populated by n robots indexed $1, \dots, n$. The configuration space of robot i is denoted \mathcal{C}_i and the region of the workspace that robot i occupies when it is at configuration c is denoted $R_i(c)$. Let π_1, \dots, π_n be feasible collision-free trajectories from the desired origin configuration to the desired destination configuration obtained from an arbitrary multi-robot trajectory planner. The continuous-time trajectory of robot i is a function $\pi_i : [0, T] \rightarrow \mathcal{C}_i$, where T denotes the time when the last robot reaches its destination.

The state of the multi-robot system is described in terms of position $x_i \in [0, T]$ of each robot i along its trajectory π_i . That is, if a robot is in state $x_i \in [0, T]$, then it is at configuration $\pi_i(x_i) \in \mathcal{C}_i$ and it occupies the spatial region $R_i(\pi_i(x_i)) \subset \mathcal{W}$.

We construct a coordination space χ for the multi-robot system as an n -dimensional cube $\chi := [0, T]^n$ (Kant and Zucker, 1986; O'Donnell and Lozano-Perez, 1989; LaValle, 2006). A point in the coordination space encodes the state of all robots, i.e. the position along their trajectories. Robots i and j are said to be in collision in system configuration $x = (x_1, \dots, x_n)$ if

$$R_i(\pi_i(x_i)) \cap R_j(\pi_j(x_j)) \neq \emptyset. \quad (1)$$

The set of all couples (x_i, x_j) such that i and j are in collision is defined as

$$C_{ij} := \{(x_i, x_j) \mid R_i(\pi_i(x_i)) \cap R_j(\pi_j(x_j)) \neq \emptyset\} \quad (2)$$

The obstacle region χ^{obs} in the coordination space of n robots is defined as :

$$\chi^{\text{obs}} := \{(x_1, \dots, x_n) \mid \exists i, j \ i \neq j : (x_i, x_j) \in C_{ij}\}. \quad (3)$$

We let $\chi^{\text{free}} := \chi \setminus \chi^{\text{obs}}$ denote the obstacle-free region in the coordination space.

A (collision-free) solution to the coordination problem in the coordination space is defined as a component-wise non-decreasing¹ path $\varphi : [0, 1] \rightarrow \chi^{\text{free}}$ such that :

$$\varphi(0) = (0 \dots 0) \text{ and } \varphi(1) = (T \dots T). \quad (4)$$

The requirement on the path to be non-decreasing is a standard assumption adopted by many works in the field (e.g, (Kant and Zucker, 1986; O'Donnell and Lozano-Perez, 1989; Ghrist et al, 2005; Ghrist and Lavalle, 2006)). It makes even more sense in our particular scheme, as our goal is to execute the plan, i.e. to move forwards along the planned trajectories. While deadlocks can be trivially resolved by reversing the progress of some of the robots at a cost of increased travel time, our goal is to reach destination in minimum time, without the need to move backwards and backup to the initial plan.

¹ Each component $\varphi_i : [0, 1] \rightarrow [0, T]$ has to be non-decreasing.

We let Φ denote the set of solutions, equipped with the topology of pointwise convergence. We say that two solutions φ^1 and φ^2 are homotopic if and only if there exists a continuous map $H : [0, 1] \rightarrow \Phi$ such that $H(0) = \varphi^1$ and $H(1) = \varphi^2$.

Observe that, if planned trajectories are precisely followed by all robots (in the absence of disturbance), the path of the multirobot system in the coordination space is a diagonal line segment joining $(0, \dots, 0)$ to (T, \dots, T) . This path will be referred to as the planned path, denoted δ , and defined component-wise as $\delta_i : \tau \in [0, 1] \mapsto \tau T$. It is remarkable that this path, δ , is a (collision-free) solution by construction:

Property 1 (Diagonal collision-freeness) The diagonal path δ is a (collision-free) solution : $\delta \in \Phi$.

We built the coordination space such that for all $\tau \in [0, T]$, $(\pi_1(\tau) \dots \pi_n(\tau)) = \pi(\tau)$ is a point of the collision-free planned trajectory. As a result, $(\tau \dots \tau) \notin \mathcal{X}^{\text{obs}}$ for all τ . All the results of the paper are based on the above property.

We are interested in the existence of a subset $\mathcal{X}^\delta \subset \mathcal{X}^{\text{free}}$ that satisfies $\forall \varphi : [0, 1] \rightarrow \mathcal{X}^\delta : \varphi$ is homotopic to δ . Intuitively, we're looking for a region of the coordination space such that the relative order between robots remains the same.

2.2 Construction of the homotopic constraint

To construct the maximal set \mathcal{X}^δ , we proceed very intuitively by completing the collision regions on the North-West and South-East directions, forming a collision-free and deadlock-free corridor around the diagonal path as depicted in Figure 2. The mathematical construction is detailed in Appendix A. We obtain a completed obstacle region, denoted $\mathcal{X}^{\text{obs}\delta}$ as follows:

$$\mathcal{X}^{\text{obs}\delta} := \{x \in \mathcal{X} : \exists i, j, (x_i, x_j) \in (C_{ij}^{\text{NW}} \cup C_{ij}^{\text{SE}})\}. \quad (5)$$

We define \mathcal{X}^δ as the complement of the completed obstacle region : $\mathcal{X}^\delta := \mathcal{X} \setminus \mathcal{X}^{\text{obs}\delta}$. Interestingly, by simple geometric considerations, the set \mathcal{X}^δ is invariant through min and max operators as illustrated in Figure 2 (these operators are defined in Appendix A).

We can now present the main result of this section and the first contribution of this paper:

Theorem 1 *A solution $\varphi \in \Phi$ is homotopic to δ if and only if φ takes values in \mathcal{X}^δ .*

The proof can be found in Appendix B and follows the same scheme as the ones in (Gregoire, 2014) in a much more general setting as there is no need to introduce the notion of priority in our setting².

² Note that we can see δ as a representative of the homotopy class, while in (Gregoire, 2014) the homotopy class is uniquely represented by its priority graph.

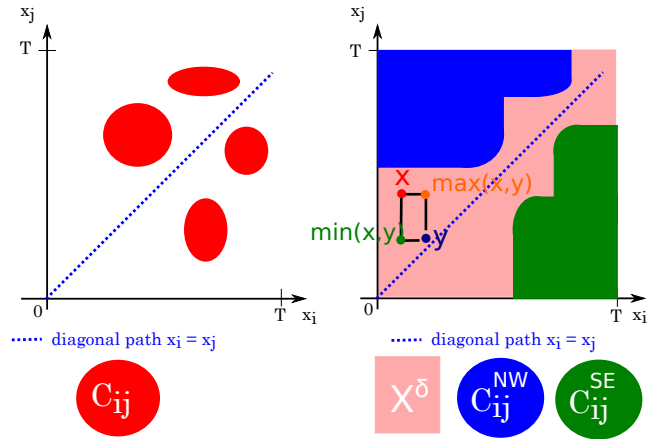


Fig. 2 Illustration of the completion of the obstacle region to constrain the trajectory in the coordination within the same homotopy class as the planned path. All connected regions above (resp. below) the diagonal segment are completed in the North-West (resp. South-East) direction.

The above result means that \mathcal{X}^δ can be seen as a homotopy constraint: a set of configurations that correspond to values taken by solutions within the same homotopy class. It allows to design a maximally permissive feedback control policy to remain within the homotopy class in the presence of delaying disturbances. All we have to do to remain within the assigned homotopy class is to take care not to cross the boundary of the constructed corridor \mathcal{X}^δ .

Generally speaking, some configurations in \mathcal{X}^δ could be deadlock configurations requiring robots to move backwards to reach their goal. The theorem below states that \mathcal{X}^δ is actually a deadlock-free space, i.e. from any point in \mathcal{X}^δ : there exists a non-decreasing collision-free path to the goal configuration (T, \dots, T) . This is an important property, as it means that as long as a controller guarantees the system stays in \mathcal{X}^δ , then deadlocks are provably avoided.

Theorem 2 *\mathcal{X}^δ is a deadlock-free space, i.e. for all $x \in \mathcal{X}^\delta$, there exists a solution φ^x going through x .*

Proof Take $x \in \mathcal{X}^\delta$. And consider the following path:

$$\varphi_i^x(\tau) := \begin{cases} \min(x_i, 2\tau T) & \text{for } \tau \in [0, 1/2] \\ \max(x_i, 2(\tau - 1/2)T) & \text{for } \tau \in (1/2, 1] \end{cases} \quad (6)$$

By Property 2, $x \in \mathcal{X}^\delta$ and δ taking values in \mathcal{X}^δ implies that φ^x takes values in \mathcal{X}^δ . Moreover, it starts at $(0 \dots 0)$, it ends at $(1 \dots 1)$ and it is non-decreasing, so that φ^x is a solution to the coordination problem going through x (at time 1/2). \square

Note that Theorem 2 is on its own sufficient for the design of the proposed controller. However, Theorem 1 provides an important insight that allows one to interpret the proposed control rule as a strategy that ensures that the trajectory of the system is in the homotopy class of the planned solution, i.e the proposed rule is an instance of navigation under homotopy constraints.

3 Locally-optimal navigation within a homotopy class under disturbances

In this section, we will describe a control rule for controlling the advancement of individual robots along their trajectories in response to some of the robots being delayed by a disturbance such that the path of the multi-robot system in the coordination space remains in the homotopy class of the planned coordination-space path. As a consequence of the properties discussed in the previous section, the fact that the system remains in this homotopy class ensures that all robots reach their goals without getting trapped in a deadlock.

3.1 Problem formulation

For simplicity and practical considerations, we will describe the controller in discrete time. The discrete-time trajectory of robot i is a function $\pi_i(t) : \{0, 1, \dots, T\} \rightarrow \mathcal{W}$, where T denotes the time step when the last robot reaches its destination. The state of the system is described in terms of position $x_i \in \{0, \dots, T\}$ of each robot i along its trajectory π_i , i.e. if robot i is in state x_i , the robot is at spatial position $\pi_i(x_i)$. The control variables are $a_i(t) \in \{0, 1\}$ for each robot i . If $a_i(t) = 1$, robot i is trying to proceed; if $a_i(t) = 0$, robot i willingly stops. Robots control variables are subject to an exogenous multiplicative disturbance $d_i(t) \in \{0, 1\}$, where $d_i(t) = 0$ models the situation when the robot is forced to stop. Then, the discrete-time system dynamics is governed by the equation:

$$\forall t \in \mathbb{N}, x_i(t+1) = x_i(t) + a_i(t) \cdot d_i(t).$$

The state $x_i(t)$ of robot i can be intuitively interpreted as a position in the plan π_i at time t . Note that this position is measured in time units. Our model is illustrated in Figure 3.

Our objective is to design a multi-robot controller $G(x_1, \dots, x_n)$ that takes the current position of each robot and returns control variables (a_1, \dots, a_n) such that from the initial state $\mathbf{x}_0 = (0, \dots, 0)$ at time $t = 0$, the system is collision-free, all robots reach destination following a path in the coordination space that belongs to the same homotopy class as δ , and the travel time for each robot to reach their goal is locally minimal, i.e. it is minimal given the assigned homotopy class.

In the sequel, we will present a control scheme that satisfies the above requirements. We remark that the presented scheme combines advantages of planning and reactive approaches in that it guarantees liveness and in the same time retains freedom of action allowing robots to deviate from the planned trajectory and handle the disturbance thanks to the freedom of action within the homotopy class.

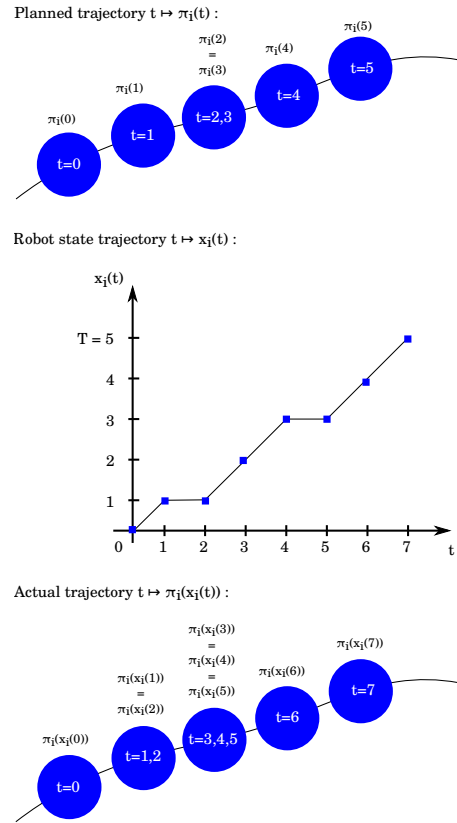


Fig. 3 Top: An example output of the planner for robot i . The balls indicate the time evolution of the planned trajectory $t \mapsto \pi_i(t)$ for robot i . The curve represents the geometric path in \mathbb{R}^2 followed by robot i . **Middle:** State trajectory $t \mapsto x_i(t) \in [0, 5]$ encodes the time evolution of position of robot i along its planned trajectory. We can see that during time intervals $[1, 2]$ and $[4, 5]$ the robot makes no progress in its plan. Also, the fact that the state trajectory is below the diagonal can be interpreted as the robot lagging behind its plan. **Bottom:** The actual trajectory of the robot is $t \mapsto \pi_i(x_i(t))$. One can see that robot i will follow the same geometric path in \mathbb{R}^2 as planned. However, the time evolution along this geometric path is different. In particular, robot i will reach its goal at time 7 instead of 5.

3.2 Control law

We now introduce a control law, referred to as robust multi-robot trajectory tracking (RMTRACK), that ensures that the actual trajectory in the coordination space under disturbances remains in the homotopy class of planned path δ . As stated in Theorem 1, this is equivalent to remaining within \mathcal{X}^δ , which is a deadlock-free space (cf. Theorem 2).

The multi-robot control law $G(x_1, \dots, x_n)$ is decomposed into collection of control laws $\{G_i(x_1, \dots, x_n)\}$, each governing the advancement of robot i . The control law for single robot i is defined as

$$G_i(x_1, \dots, x_n) := \begin{cases} 0 & \text{if } x_i = T \text{ or if } [\exists j : x_i > x_j \text{ and} \\ & C_{ij} \cap (\{x_i + 1\} \times \{x_j, \dots, x_i + 1\}) \neq \emptyset] \\ 1 & \text{otherwise.} \end{cases} \quad (7)$$

As we can see, the control law allows robot i to proceed only if the line segment from $(x_i + 1, x_j)$ to $(x_i + 1, x_i + 1)$ does not go through C_{ij} for every other robot j . The mechanism is illustrated in Figure 4. The effect of the application of such a control law is illustrated in Figure 5.

Intuitively, the reader can interpret the control scheme as follows : let every robot proceed at any time, unless it is about to cross an intersection area before another robot that was supposed to cross the area first according to the initial plan. The hierarchy of the initial plan is conserved. In the example of Figure 1, it means in particular that robot 2 should not enter the corridor before robot 1 exits the corridor, which clearly avoids the presented deadlock situation. This research work provides this intuitive approach with a theoretical framework and proofs.

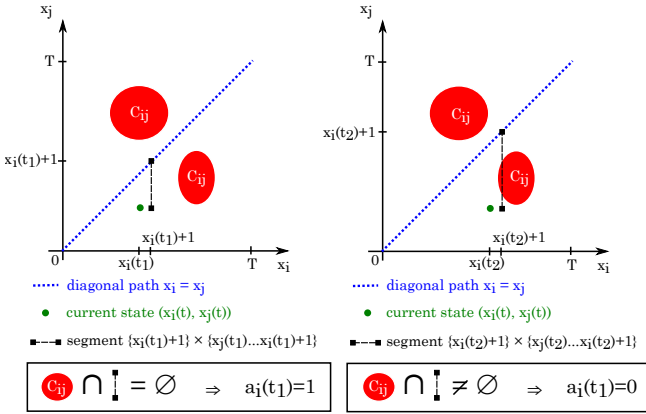


Fig. 4 Illustration of the control law computation for robot i with respect to single other robot j . **Left:** The segment $\{x_i + 1\} \times \{x_j, \dots, x_i + 1\}$ is collision-free with C_{ij} at time t_1 , therefore the robot i is commanded to proceed, i.e. $a_i(t_1) = 1$. **Right:** The segment is not collision-free with C_{ij} at time t_2 , therefore the robot i is commanded to stop, i.e. $a_i(t_2) = 0$.

In the sequel, we show that under certain technical assumptions, the RMTRACK control law satisfies collision-freeness and liveness properties, i.e. the robots are guaranteed not to collide and eventually reach their goal.

3.3 Properties

3.3.1 Collision-freeness

In order to show collision-freeness we make the following technical assumption. We assume that there is a 1-margin between the diagonal path and the obstacle region in the coordination space, i.e.

$$\forall t \in \{0, \dots, T-1\}, \forall i \neq j, \begin{cases} (t+1, t) \notin C_{ij} \\ \text{and} \\ (t, t+1) \notin C_{ij} \end{cases} \quad (8)$$

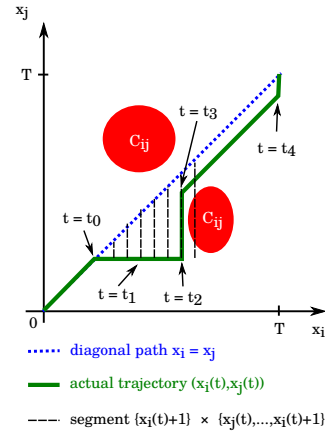


Fig. 5 Trajectory of the robots in the coordination space under control law G . At time $t = t_0$, a disturbance makes robot j stop ($d_j(t_0) = 0$). The system leaves the diagonal path, and the control law allows robot i to proceed until time $t = t_2$ where robot i is commanded to stop, i.e. $a_i(t_2) = 0$ (see Figure 4). Finally, disturbance for robot j goes away, and robot j proceeds. It's only at time $t = t_3$ that the control law allows robot i to proceed. Robot i reaches its goal at time $t = t_4$ and robot j right after. No collision occurred and note how the control law ensured that the actual trajectory in the coordination space remains on the same side as the diagonal path with respect to each connected component of the obstacle region. This is what enables to guarantee liveness, as changing the homotopy class may lead to a deadlock configuration with respect to other robots.

This assumption is typically negligible as the geometric distance traveled by a robot in one time step is small. It can also be satisfied by planning the reference trajectories with robots modeled with slightly larger bodies than actual robots (to ensure a sufficient safety margin).

The following lemma is a preliminary to prove that the control strategy is collision-free.

Lemma 1 Under control law G and assuming that Equation (8) holds, we have for all $i, j \in \{1, \dots, n\}$ and for all $t \in \mathbb{N}$ s.t. $x_i(t) \geq x_j(t)$:

$$C_{ij} \cap (\{x_i(t)\} \times \{x_j(t), \dots, x_i(t)\}) = \emptyset \quad (E_{i,j,t})$$

Proof Initially, we have $x_1(0) = x_2(0) = \dots = x_n(0) = 0$ and the state of robots does not belong to C_{ij} , so that $(E_{i,j,0})$ holds for all $i, j \in \{1, \dots, n\}$.

Now, assume that $(E_{i,j,t})$ holds at some arbitrary time step $t \in \mathbb{N}$ for all $i, j \in \{1, \dots, n\}$ s.t. $x_i(t) \geq x_j(t)$.

For each $i, j \in \{1, \dots, n\}$, consider two options:

- If $x_i(t) = x_j(t)$, then $(E_{i,j,t}) \equiv (E_{j,i,t})$ hold and consider three options:
 - If $x_i(t+1) = x_j(t+1)$, $(E_{i,j,t+1}) \equiv (E_{j,i,t+1})$ will be satisfied as planned trajectories are collision-free.
 - If robot i moves one step forward while robot j is stopped, then by Equation (8), we have $(x_i(t+1), x_j(t+1)) = (x_i(t) + 1, x_j(t)) \notin C_{ij}$, and as planned trajectories are collision-free $(x_i(t+1), x_i(t+1)) \notin C_{ij}$, so

that $\{x_i(t+1)\} \times \{x_j(t+1), x_i(t+1)\} \cap C_{ij} \neq \emptyset$, so that $(E_{i,j,t+1})$ is satisfied with $x_i(t+1) > x_j(t+1)$.

- If robot i moves one step forward while the other one j is stopped, we use the symmetric reasoning to obtain that $(E_{j,i,t+1})$ holds.
- If $x_i(t) > x_j(t)$ then we have $x_i(t+1) \geq x_j(t+1)$ and consider three options:
 - If neither of robots moves, $(E_{i,j,t+1})$ will still be obviously satisfied as $(E_{i,j,t+1}) \equiv (E_{i,j,t})$ which holds.
 - If robot i does not move, then we have:

$$\begin{aligned} & \{x_i(t+1)\} \times \{x_j(t+1), \dots, x_i(t+1)\} \\ &= \{x_i(t)\} \times \{x_j(t+1), \dots, x_i(t)\} \\ & \quad \subseteq \{x_i(t)\} \times \{x_j(t), \dots, x_i(t)\} \end{aligned}$$
 which does not intersect C_{ij} as $(E_{i,j,t})$ holds, so that $(E_{i,j,t+1})$ is satisfied.
 - If robot i moves, then by construction of G and because $x_i(t) > x_j(t)$, we have:

$$C_{ij} \cap (\{x_i(t)+1\} \times \{x_j(t), \dots, x_i(t)+1\}) = \emptyset \quad (9)$$
 Taking into account that $x_i(t+1) = x_i(t)+1$ and $x_j(t+1) \in \{x_j(t), x_j(t)+1\}$, we obtain:

$$\begin{aligned} & \{x_i(t+1)\} \times \{x_j(t+1), \dots, x_i(t+1)\} \\ &= \{x_i(t)+1\} \times \{x_j(t+1), \dots, x_i(t)+1\} \\ & \quad \subseteq \{x_i(t)+1\} \times \{x_j(t), \dots, x_i(t)+1\} \end{aligned}$$
 which does not intersect C_{ij} by Equation (9), so that $(E_{i,j,t+1})$ holds.

By induction, we conclude that $(E_{i,j,t})$ is satisfied for all $t \in \mathbb{N}$ and $i, j \in \{1, \dots, n\}$ s.t. $x_i(t) \geq x_j(t)$. \square

Theorem 3 *Under control law G and assuming that Equation (8) holds, the trajectory in the coordination space is collision-free, i.e.*

$$\forall t \in \mathbb{N}, (x_1(t), \dots, x_n(t)) \notin C.$$

Proof Take an arbitrary time step $t \in \mathbb{N}$. Assume that $(x_1(t), \dots, x_n(t)) \in C$. Then, there exists i, j such that $(x_i(t), x_j(t)) \in C_{ij}$ and we can assume without loss of generality that $x_i(t) \geq x_j(t)$. This is in contradiction with Lemma 1. \square

3.3.2 Liveness

First, we characterize our assumptions on disturbances. Clearly, it is possible to construct a disturbance function that will prevent the system from reaching configuration (T, \dots, T) under any control law. For example, if for a given robot $i \in \{1, \dots, n\}$, we have $\forall t \in \mathbb{N}, d_i(t) = 0$, then it is impossible for robot i to reach its goal. Therefore, in the following analysis, we assume that disturbances do not prohibit any of the robots from reaching its goal, i.e. we consider systems in which disturbances may delay any given robot for arbitrarily long, but the robot will eventually be able to reach the goal. Formally, we say that disturbances are *non-prohibitive* if for

any controller that satisfies

$$\forall t \in \mathbb{N}, \begin{cases} x_1(t) = x_2(t) = \dots = x_n(t) = T \\ \text{or} \\ \exists i \in \{1, \dots, n\} : x_i(t) < T \text{ and } a_i(t) = 1 \end{cases},$$

the system will eventually reach the goal, i.e. there exists $t_f \in \mathbb{N}$ such that $(x_1(t_f), \dots, x_n(t_f)) = (T, \dots, T)$. In other words, as long as the controller lets at least one unfinished robot proceed at any point of time, all robots will eventually reach their goal. The time of goal achievement t_f might be affected by disturbances, but disturbances will not prevent goal achievement in finite time.

Under *non-prohibitive* disturbances, the control law G guarantees liveness:

Lemma 2 *Under control law G , there is at least one robot proceeding at any point of time, i.e.*

$$\forall t \in \mathbb{N}, \begin{cases} x_1(t) = x_2(t) = \dots = x_n(t) = T \\ \text{or} \\ \exists i \in \{1, \dots, n\} : x_i(t) < T \text{ and } a_i(t) = 1 \end{cases}$$

Proof Define $I(t) \subseteq \{1, \dots, n\}$ as follows:

$$I(t) := \arg \min_i x_i(t) = \{i : \forall j \neq i \quad x_j(t) \geq x_i(t)\}$$

$I(t)$ is non-empty and finite as it is an argmin over a finite set. By construction of G , we have:

$$\forall i \in I(t) \quad a_i(t) = G_i(x_1(t), \dots, x_n(t)) = 1 \text{ or } x_i(t) = T$$

There are two scenarios: a) Either for all $i \in I(t)$, $x_i(t) = T$, then $\min_i x_i(t) = T$, so that we have $x_1(t) = x_2(t) = \dots = x_n(t) = T$. b) Or there exists some $i \in I(t)$ such that $x_i(t) < T$ and $a_i(t) = G_i(x_1(t), \dots, x_n(t)) = 1$. This concludes the proof. \square

Theorem 4 *Control law G ensures liveness under non-prohibitive disturbances, i.e.*

$$\exists t_f : x_1(t_f) = x_2(t_f) = \dots = x_n(t_f) = T$$

Proof This is a direct consequence of the preceding lemma and of the assumption made on disturbances. \square

3.3.3 Local optimality

The proposed control strategy is locally optimal: it minimizes the travel time for each robot to reach their destination among solutions within the assigned homotopy class, i.e. among solutions respecting the same relative order between robots at intersection areas. This result can be interpreted as the equivalent in our feedback control approach to the locally optimal left-greedy solution of (Ghrist et al, 2005; Ghrist and Lavelle, 2006). It is intuitive given that the control strategy consists in moving forward unless the configuration of robots would cross the boundary of the "homotopy corridor" χ^δ , which is a greedy strategy.

Theorem 5 *Control law G is locally optimal, i.e. it minimizes travel time for each robot to reach their destination among solutions within the assigned homotopy class.*

Proof This result is a direct adaptation of the proof of "optimality under assigned priorities in the absence of kinodynamic constraints" that can be found in (Gregoire, 2014). \square

3.4 Extension under disturbances on the planned geometric paths

The results presented in this paper consider that robots are still able to precisely follow the initially planned geometric paths and that disturbances only affect the time execution of the plan. A major possible extension of this work is to handle small disturbances affecting the geometric paths as well, and we provide here some insights on the matter.

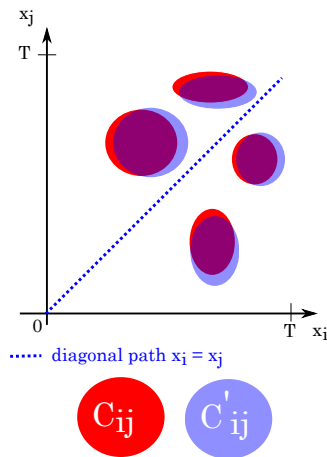


Fig. 6 Under small disturbances on the geometric paths, the coordination space has the same topological structure.

For sufficiently small disturbances on the geometric paths and assuming the diagonal path is not in direct contact with the obstacle region (some security margin exists), then the new coordination space corresponding to the new geometric paths will have the same topological structure as depicted in Figure 6. As a consequence, a new corridor χ'^{δ} can be derived and locally-optimal navigation can be performed within this corridor. We believe that our theoretical framework can be extended to obtain provably locally-optimal navigation under space-time disturbances on the initial plan, which is beyond the scope of this paper and will be the focus of our future work.

4 Experimental Evaluation

In this section we will discuss the results of experimental comparison of RMTRACK approach against a baseline liveness preserving method ALLSTOP and a reactive method ORCA using multi-robot simulation. The naive strategy ALLSTOP used as a baseline merely consists in stopping all robots in the team every time any single robot is forced to stop, which is a very conservative and non optimal approach.

Experiment Setup

The comparison was performed in three environments: Empty hall, Office corridor and Warehouse as shown in Figure 7. A single problem instance in one of the environments consists of n robots attempting to move from randomly generated origins to randomly generated destinations. We first find collision-free multi-robot trajectories from the origins to the destinations and then let each robot follow the given trajectory while randomly disturbing its advancement with specified intensity. More precisely, every second we decide with probability corresponding to the disturbance intensity whether during the following second the robot will be prevented from moving.

To ensure that the initial coordinated trajectories can be found in a reliable and tractable fashion, the test environments satisfy so-called well-formed infrastructure property³ (Čáp et al, 2015a). This allowed us to use revised prioritized planning approach (Čáp et al, 2015a) to efficiently find the initial trajectories for the robots to follow.

In Empty hall and Warehouse environment we generated 10 instances with 10 robots and 10 instances with 50 robots; In Office corridor environment we generated 10 instances with 10 robots and 10 instances with 35 robots. Note that a single instance represents a specific assignment of origins and destinations to the robots.

An illustrative video, the source codes of all algorithms and benchmark instances are available at :

<http://agents.fel.cvut.cz/~cap/rmtrack/>.

Comparison of RMTRACK and ALLSTOP

First, we compare two liveness-preserving control laws to handle disturbances: 1) ALL-STOP: the baseline law that makes entire multi-robot team stop whenever a single robot is disturbed and 2) RMTRACK: the law proposed in Section 3.2. For each instance and disturbance intensity ranging from 0 % to 50 %, we run both algorithms and measured the time it took for each robot to reach its destination.

³ In a well-formed infrastructure a start and destination of each robot is constrained to lie at a position where it does not completely prevent other robots from reaching their goals - most man-made infrastructures, e.g. a national road network system, satisfy the property.

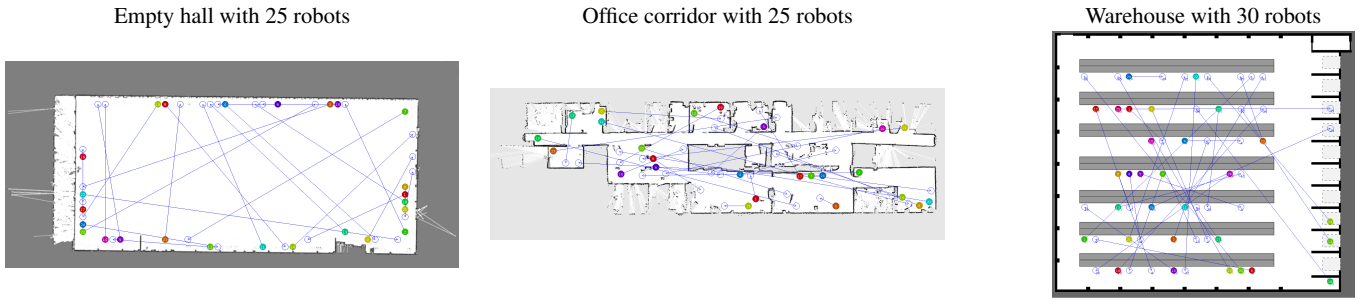


Fig. 7 Maps used for experimental comparison. The figures show an example problem instance in each environment. The filled circles represent robots. The arrows indicate the desired destination of each robot.

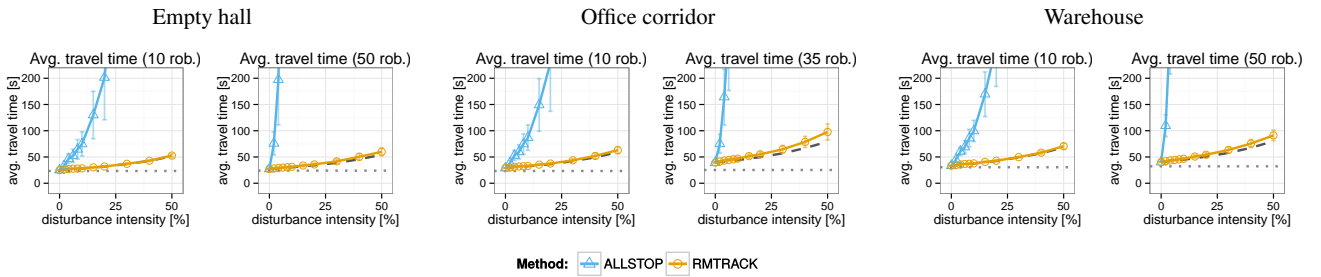


Fig. 8 Experimental comparison of ALL-STOP strategy with RMTRACK. Each datapoint is an average travel time of a single robot from its origin to its destination under the given disturbance intensity using one of the two evaluated control strategies. The dashed line represents the average lower bound on the travel time under the given disturbance intensity. The dotted line represents the average travel time from origin to destination assuming no disturbance and no need for coordination between robots. The bars represent standard deviation of the difference between the travel time under the evaluated algorithm and the lower bound travel time.

In order to isolate the effect of each control law on the travel time from the effect of disturbances and the effect of the quality of the initial plan, we compute a lower-bound on the travel time of each robot assuming fixed disturbance and fixed initial plan. Such a lower bound is obtained by simulating the robot such that the inter-robot collisions are ignored and thus all robots always command to proceed at maximum advancement rate along their initial trajectory. Then the average advancement rate of the robot and consequently the travel time is affected solely by disturbances. In fact, for uniformly distributed random disturbance with equal intensity q for all robots, this corresponds to robots advancing on expectation at $1 - q$ fraction of the original advancement rate 1. Thus, the lower bound on expected travel time under disturbance intensity q can be also computed as $E(t_f)/(1 - q)$, where $E(t_f)$ denotes expected travel time in the absence of disturbance.

It is easy to see that this lower bound represents the best possible travel time that can be achieved by RMTRACK, for instance, the travel time when the paths of the robots do not overlap. On the other hand, it is not difficult to construct a combination of problem instance and disturbances for which the behavior of RMTRACK degenerates to that of ALLSTOP. Curiously, since ALLSTOP proceeds only when none of the robots is disturbed, which for uniformly dis-

tributed disturbance with intensity q at each robot happens with probability $(1 - q)^n$, the expected travel time for ALLSTOP strategy can be consequently computed as $E(t_f)/(1 - q)^n$, where $E(t_f)$ is again the expected travel time without disturbance and n is the number of robots in the system.

Consequently, we expect the average traveltime under RMTRACK strategy to be bounded from below by the lower-bound travel time and by the ALLSTOP travel time from above. The actual travel time under RMTRACK will then depend on the "interdependency" of initial trajectories and the level of disturbance. Given these two bounds, an interesting question is how will RMTRACK strategy perform in characteristic real-world environments.

The results of performance comparison of RMTRACK with respect to ALLSTOP and the lower bound travel time for the three test environments are shown in Figure 8. We can see that consistently over all test environments and for different numbers of robots, the baseline strategy ALLSTOP quickly becomes impractical when the disturbance intensity is high. In contrast, the average travel time under RMTRACK remains reasonable even for high intensities of disturbance. Further, it is encouraging that for all three environments we tested on, the average travel time under RMTRACK remains close to the lower-bound travel time.

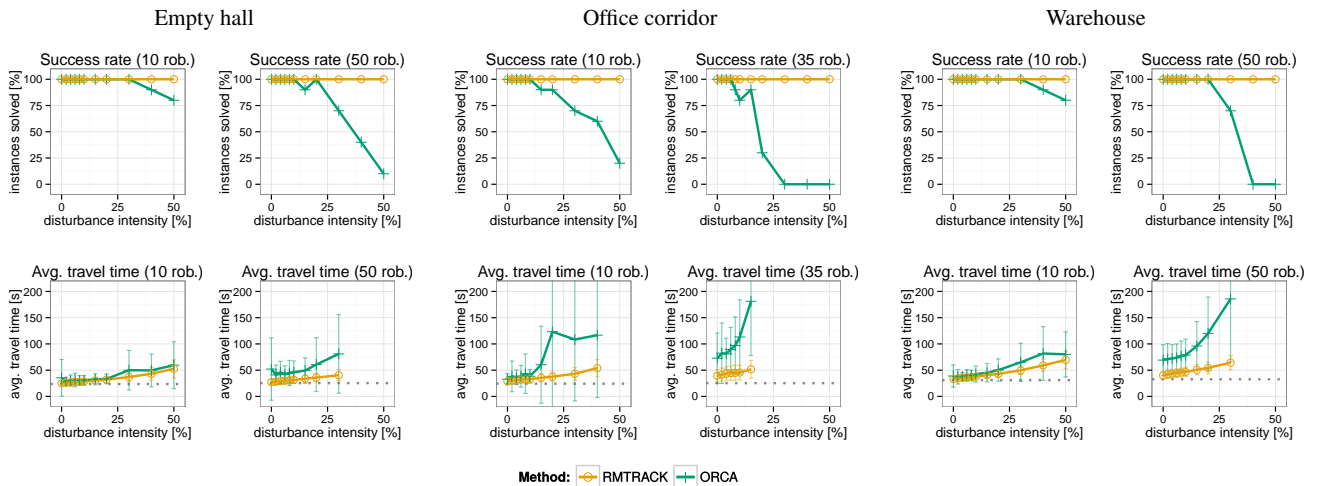


Fig. 9 Experimental comparison of ORCA and RMTRACK. The average travel time is computed if there is at least five instances for given number of robots and disturbance intensity successfully solved by both evaluated algorithms. The dotted line represents the average travel time from origin to destination assuming that no disturbance and no need for coordination with other robots. The bars represent standard deviation of the difference between the travel time under the evaluated algorithm and the average travel time from origin to destination ignoring collisions with other robots and disturbance.

Comparison of RMTRACK and ORCA

Next, we compared RMTRACK strategy with a reactive collision avoidance technique ORCA, which is a characteristic representative of a family of popular collision avoidance algorithms based on the reciprocal velocity obstacle paradigm (Van Den Berg et al, 2011). Given the current velocities of all robots in the neighborhood and the desired velocity vector, it attempts to compute the closest velocity vector to the current desired velocity that does not lead to future collision with other robots, assuming that they will continue moving at their current velocity. In our implementation, the desired velocity at each time instance follows the shortest path to destination. For each instance we run ORCA and RMTRACK techniques for different disturbance intensities ranging from 0% to 50%. During the experiment, we often witnessed ORCA leading robots to dead-lock situations during which the robots either moved at extremely slow velocities or even stopped completely. Therefore, if the robots failed to reach their destination within 10 minutes⁴, we considered the run as failed.

Figure 9 summarizes the results of the comparison. We can see that the success rate of ORCA deteriorates with increasing disturbance intensity. This is perhaps surprising since reactive methods are believed to be particularly well suited for unpredictable environments. Among the reasons behind this phenomenon seems to be that the reciprocal reactive algorithms rely on all robots executing the same algorithm and consequently on "splitting" the collision avoidance effort. This assumption is however violated if one of the robots is

disturbed and does not execute the velocity command that the algorithm computed. The plots in the bottom row show comparison of performance of RMTRACK and ORCA. We can see that even when ORCA solves a given instance, the expected travel time for a robot is on expectation longer, especially so in cluttered environments and for high disturbance intensities.

5 Conclusion

The ability to guarantee safe and dead-lock free motion coordination for autonomous multi-robot systems in unpredictable environments is a standing challenge. To tackle this challenge, this work proposed a new paradigm based on navigation under homotopic constraints. We use the output of any multi-robot trajectory planner to construct a maximal set of homotopic solutions to the coordination problem and show that remaining in this homotopy class is equivalent to constraining robots to navigate within a subset χ^δ of the constructed coordination space. Then, we propose a feedback controller which ensures that robots respect the constructed homotopic constraint.

With the proposed control scheme, collision avoidance and deadlock avoidance are guaranteed even if unexpected events require some or even all robots to stop for an arbitrary (yet finite) time. Moreover, the control scheme is locally-optimal as it minimizes travel time among solutions homotopic to the initial solution. The delay could not be smaller, unless a completely new plan is computed. The obtained control scheme is particularly useful for autonomous intralogistics systems where robots and humans share a common

⁴ Average travel time between origin and destination ignoring collisions and without disturbance is around 25 second

environment. It is remarkable that a quite theoretical result on the existence of homotopy classes in the coordination space can be leveraged to provide such guarantees at the system level.

The experimental evaluation shows that the proposed method is both more reliable and more efficient than the state-of-the-art reactive collision-avoidance methods.

There are two main perspectives for future work. First of all, as detailed in Subsection 3.4, we would like to handle space-time disturbances, i.e. disturbances affecting not only the time execution of the plan but also requiring robots to follow a slightly different geometric path. Future work should extend our theoretical results to obtain a provably locally-optimal controller under space-time disturbances. Under major space-time disturbances, local approaches guaranteeing the system remains within the initially planned homotopy class will prove inefficient. As a result, future work should also study replanning in our homotopy class paradigm. In the presented approach, the system is bound to remain in the homotopy class of the initial solution. The system would be able to reach the goal faster if it finds a new solution from the current state and start navigating in its homotopy class. It would be interesting to gain better understanding on when the change in the homotopy class is unnecessary, when it is significantly beneficial, and how to do it in a computationally efficient manner.

References

- Alonso-Mora J, Gohl P, Watson S, Siegwart R, Beardsley P (2014) Shared Lumelsky1997control of autonomous vehicles based on velocity space optimization. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, pp 1639–1645
- Van den Berg J, Lin M, Manocha D (2008) Reciprocal velocity obstacles for real-time multi-agent navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, pp 1928–1935
- Brock O, Khatib O (2002) Elastic strips: A framework for motion generation in human environments. *The International Journal of Robotics Research* 21(12):1031–1052
- Čáp M, Gregoire J, Frazzoli E (2016) Provably safe and deadlock-free execution of multi-robot plans under delaying disturbances. In: Proceedings of the IEEE Conference on Intelligent Robots and Systems, pp 5113–5118, DOI 10.1109/IROS.2016.7759750
- Dresner K, Stone P (2008) A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research* pp 591–656
- Erdmann M, Lozano-Pérez T (1987) On multiple moving objects. *Algorithmica* 2:1419–1424
- Ghrist R, LaValle SM (2006) Nonpositive curvature and pareto optimal coordination of robots. *SIAM Journal on Control and Optimization* 45:1697–1713
- Ghrist R, O’Kane JM, LaValle SM (2005) Computing pareto optimal coordinations on roadmaps. *The International Journal of Robotics Research* 12:997–1012
- Gregoire J (2014) Priority-based coordination of mobile robots. arXiv preprint arXiv:14100879
- Guizzo E (2008) Three engineers, hundreds of robots, one warehouse. *Spectrum, IEEE* 45(7):26–34
- Guy SJ, Chhugani J, Kim C, Satish N, Lin M, Manocha D, Dubey P (2009) Clearpath: Highly parallel collision avoidance for multi-agent simulation. In: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, New York, NY, USA, SCA ’09, pp 177–187
- Kant K, Zucker SW (1986) Toward efficient trajectory planning: The path-velocity decomposition. *International Journal of Robotics Research* 5(3):72–89
- Kowshik H, Caveney D, Kumar P (2011) Provable systemwide safety in intelligent intersections. *IEEE Transactions on Vehicular Technology* 60(3):804–818
- LaValle SM (2006) *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., available at <http://planning.cs.uiuc.edu/>
- Lumelsky VJ, Harinarayan K (1997) Decentralized motion planning for multiple mobile robots: The cocktail party model. In: *Robot colonies*, Springer, pp 121–135
- O’Donnell P, Lozano-Perez T (1989) Deadlock-free and collision-free coordination of two robot manipulators. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp 484–489 vol.1
- Pallottino L, Scordio VG, Bicchi A, Frazzoli E (2007) Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Transactions on Robotics* 23(6):1170–1183
- Quinlan S, Khatib O (1993) Elastic bands: Connecting path planning and control. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, pp 802–807
- Solovey K, Halperin D (2016) On the hardness of unlabeled multi-robot motion planning. *The International Journal of Robotics Research* 35(14):1750–1759, DOI 10.1177/0278364916672311
- Solovey K, Yu J, Zamir O, Halperin D (2015) Motion planning for unlabeled discs with optimality guarantees. In: Proceedings of Robotics: Science and Systems, Rome, Italy, DOI 10.15607/RSS.2015.XI.011
- Spirakis PG, Yap CK (1984) Strong np-hardness of moving many discs. *Information Processing Letters* 19(1):55–59
- Turpin M, Mohta K, Michael N, Kumar V (2014) Goal Assignment and Trajectory Planning for Large Teams of Interchangeable Robots. *Autonomous Robots* 37(4):401–

415, DOI 10.1007/s10514-014-9412-1

Van Den Berg J, Guy S, Lin M, Manocha D (2011) Reciprocal n-body collision avoidance. *Robotics Research* pp 3–19

Čáp M, Novák P, Kleiner A, Selecký M (2015a) Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Transactions on Automation Science and Engineering* 12(3):835–849, DOI 10.1109/TASE.2015.2445780, URL <http://dx.doi.org/10.1109/TASE.2015.2445780>

Čáp M, Vokřínek J, Kleiner A (2015b) Complete decentralized method for on-line multi-robot trajectory planning in well-formed infrastructures. In: *Proceedings of the International Conference on Automated Planning and Scheduling*, pp 324–332

A Completion of the obstacle region

In this section, we construct the maximal set $\chi^\delta \subset \chi^{\text{free}}$. We start by defining Δ^{NW} and Δ^{SE} as follows:

$$\Delta^{\text{NW}} := \{(a, b) \in [0, T]^2 : b > a\} \quad (10)$$

$$\Delta^{\text{SE}} := \{(a, b) \in [0, T]^2 : b < a\}. \quad (11)$$

Then, we build the completed obstacle region $\chi^{\text{obs}\delta}$ as follows:

$$C_{ij}^{\text{NW}} := ((C_{ij} \cap \Delta^{\text{NW}}) + (\mathbb{R}_- \times \mathbb{R}_+)) \cap [0, T]^2 \quad (12)$$

$$C_{ij}^{\text{SE}} := ((C_{ij} \cap \Delta^{\text{SE}}) + (\mathbb{R}_+ \times \mathbb{R}_-)) \cap [0, T]^2 \quad (13)$$

where $A + B$ denotes the set of $a + b$ with $(a, b) \in A \times B$, $\mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$ and $\mathbb{R}_- := \{x \in \mathbb{R} : x \leq 0\}$. The completion process is depicted in Figure 2.

We use $\min(x, y)$ and $\max(x, y)$ operators on two points x, y in the coordination space, defined component-wise as :

$$(\min(x, y))_i := \min(x_i, y_i) \quad (14)$$

$$(\max(x, y))_i := \max(x_i, y_i). \quad (15)$$

Property 2 (Invariance through min and max operators) For all $x, y \in \chi^\delta$, we have $\min(x, y) \in \chi^\delta$ and $\max(x, y) \in \chi^\delta$.

B Proof of Theorem 1

Proof (Necessary condition) We first prove that taking values in χ^δ is a necessary condition for being homotopic to δ by contradiction. Consider a solution φ . Assume that φ is homotopic to δ , but it does not take only values in χ^δ . As a consequence, it takes some value $x^c \in \chi^{\text{obs}\delta}$ at some point τ^c such that $(\varphi_i, \varphi_j)(\tau^c) = (x_i^c, x_j^c) \in C_{ij}^{\text{NW}}$ for some $i, j \in \{1 \dots n\}$ (or, equivalently, $(x_j^c, x_i^c) \in C_{ji}^{\text{SE}}$). By construction of C_{ij}^{NW} , there exists $(x_i^0, x_j^0) \in C_{ij}$ distinct from (x_i^c, x_j^c) (because φ is collision-free) such that $x_i^0 \geq x_i^c$ and $x_j^0 \leq x_j^c$. Consider the maximal segment $\Sigma \subset [0, T]^2$ going through points (x_i^0, x_j^0) and (x_i^c, x_j^c) . As φ is assumed to be homotopic to δ , there exists a continuous transformation $H : [0, 1] \rightarrow \Phi$ such that $H(0) = \varphi$ and $H(1) = \delta$. For all $\alpha \in [0, 1]$, $H(\alpha)$ intersects Σ . $H(\alpha) = \varphi$ intersects at configuration (x_i^c, x_j^c) and $H(1)$ intersects Σ at a configuration on the image of δ . As a consequence, by continuity, $H(\alpha)$ goes through (x_i^0, x_j^0) for some α , which is absurd as $H(\alpha)$ should be a solution for all $\alpha \in [0, 1]$ (solution $H(\alpha)$ should be collision-free in particular). \square

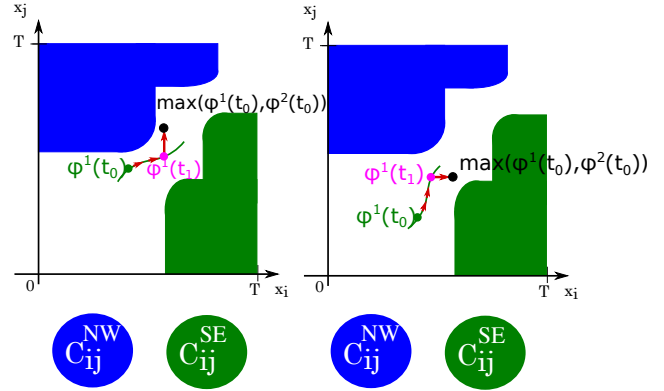


Fig. 10 Illustration of the transformation of φ^1 into $\max(\varphi^1, \varphi^2)$. Two cases may appear, and in both cases, the transformation remains within χ^δ .

Proof (Sufficient condition) Now, we prove that taking values in χ^δ is a sufficient condition. Consider two arbitrary solutions $\varphi^1, \varphi^2 \in \Phi$ taking values in χ^δ and the following continuous transformation H defined as follows for all $\alpha \in [0, 1]$, $\tau \in [0, 1]$:

$$H(\alpha)(\tau) := \min(\varphi^1(\tau + \alpha), \max(\varphi^1(\tau), \varphi^2(\tau))) \quad (16)$$

where $\varphi^1(\tau + \alpha) \equiv (T \dots T)$ if $\tau + \alpha > 1$ by convention.

We have $H(0)(\tau) = \min(\varphi^1(\tau), \max(\varphi^1(\tau), \varphi^2(\tau))) = \varphi^1(\tau)$, so that $H(0) = \varphi^1$. Moreover,

$$H(1)(\tau) = \min(\varphi^1(1 + \tau), \max(\varphi^1(\tau), \varphi^2(\tau))) \quad (17)$$

$$= \min(T, \max(\varphi^1(\tau), \varphi^2(\tau))) \quad (18)$$

$$= \min(T, \max(\varphi^1(\tau), \varphi^2(\tau))) \quad (19)$$

$$= \max(\varphi^1(\tau), \varphi^2(\tau)). \quad (20)$$

As a result, $H(1) = \max(\varphi^1, \varphi^2)$. Finally, by Property 2 (see also Figure 10), φ^1 and φ^2 taking values in χ^δ implies that for all $\alpha \in [0, 1]$, $H(\alpha)$ takes values in χ^δ . Moreover, $H(\alpha)$ is non-decreasing as min and max operators do not affect that property. As a result, H continuously transforms φ^1 into $\max(\varphi^1, \varphi^2)$ while remaining in Φ . By symmetry of the roles played by φ^1 and φ^2 , there also exists a continuous transformation transforming φ^2 into $\max(\varphi^1, \varphi^2)$ while remaining in Φ . As a result, φ^1 and φ^2 are both homotopic to $\max(\varphi^1, \varphi^2)$, so that φ^1 and φ^2 are homotopic solutions. In particular, choosing for φ^1 an arbitrary solution taking values in χ^δ and $\varphi^2 \equiv \delta$, we obtain that any solution taking values in χ^δ is homotopic to δ . \square