

Hamiltonian coordination primitives for decentralized multiagent navigation

The International Journal of
Robotics Research
2021, Vol. 40(10-11) 1234-1254
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/02783649211037731
journals.sagepub.com/home/ijr



Christoforos Mavrogiannis¹  and Ross A. Knepper²

Abstract

We focus on decentralized navigation among multiple non-communicating agents in continuous domains without explicit traffic rules, such as sidewalks, hallways, or squares. Following collision-free motion in such domains requires effective mechanisms of multiagent behavior prediction. Although this prediction problem can be shown to be NP-hard, humans are often capable of solving it efficiently by leveraging sophisticated mechanisms of implicit coordination. Inspired by the human paradigm, we propose a novel topological formalism that explicitly models multiagent coordination. Our formalism features both geometric and algebraic descriptions enabling the use of standard gradient-based optimization techniques for trajectory generation but also symbolic inference over coordination strategies. In this article, we contribute (a) HCP (Hamiltonian Coordination Primitives), a novel multiagent trajectory-generation pipeline that accommodates spatiotemporal constraints formulated as symbolic topological specifications corresponding to a desired coordination strategy; (b) HCPnav, an online planning framework for decentralized collision avoidance that generates motion by following multiagent trajectory primitives corresponding to high-likelihood, low-cost coordination strategies. Through a series of challenging trajectory-generation experiments, we show that HCP outperforms a trajectory-optimization baseline in generating trajectories of desired topological specifications in terms of success rate and computational efficiency. Finally, through a variety of navigation experiments, we illustrate the efficacy of HCPnav in handling challenging multiagent navigation scenarios under homogeneous or heterogeneous agents across a series of environments of different geometry.

Keywords

Robot navigation, multi-robot systems, motion planning, trajectory optimization, topological methods

1. Introduction

Although several instances of centralized multiagent motion planning problems can be shown to be NP-hard (Cooper, 1990; Hopcroft et al., 1984; Spirakis and Yap, 1984), humans are remarkably effective at solving a wide range of multiagent coordination problems ranging from navigation in dense crowds to driving through street intersections in a *distributed* fashion under *no explicit communication*. Our key observation is that human effectiveness in tackling multiagent coordination tasks can be partially attributed to *uncertainty* reduction, enabled by *coordination*. Humans coordinate via sophisticated *implicit-communication* mechanisms mapping observed actions to intentions (Baker et al., 2009; Csibra and Gergely, 2007; Wiese et al., 2012). They do so by exchanging information via modalities including body language, eye gaze, path shape, and even gestures. Specifically in navigation tasks, these mechanisms enable a seamless and fluent negotiation over collision-avoidance protocols that has been identified

as the *pedestrian bargain* (Wolfinger, 1995). This type of negotiation sets the foundation for decentralized coordination among pedestrians yielding safe, socially acceptable, and comfortable co-navigation even in crowded navigation domains.

Inspired by human effectiveness in tackling crowd navigation tasks, we seek to develop mechanisms that leverage *implicit-communication* as a tool to enable mobile robots to coordinate collision-free passages in close proximity in a *decentralized* fashion. The value of implicit

¹Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA

²Department of Computer Science, Cornell University, Ithaca, NY, USA

Corresponding author:

Christoforos Mavrogiannis, Bill and Melinda Gates Center for Computer Science and Engineering, Room 259, 3800 E Stevens Way NE, Seattle, WA 98195, USA.

Email: cmavro@cs.washington.edu

communication has been recognized by lab studies involving interaction between a robot and a human agent (Carton et al., 2016; Kruse et al., 2012): *readable* motion enables smooth collision avoidance requiring low mental load from the human agent. In public crowded environments, this feature may allow robots to seamlessly integrate, enhancing human comfort and safety. From a design perspective, enabling robots to communicate their intentions *implicitly* relaxes the need to equip them with specialized communication components (e.g., screens, turn indicators, speakers, etc.). This relaxation may afford any general-purpose mobile robot to leverage the same mechanisms of coordination to navigate safely but also offer a robust alternative in the event of hardware failure for robots with specialized communication devices.

Leveraging implicit communication for distributed multiagent coordination using general-purpose mobile robots poses a number of challenges. For instance, in contrast to humans who employ a variety of modalities, most mobile robots can only employ modalities related to the motion of their base, such as path shape, speed, and acceleration; this limits the abilities of robots for generating expressive signals. Further, coordination among multiple agents in a bounded environment requires a model of the motion constraints formed among them. Finally, in the absence of explicit communication and the limited implicit communication modalities available, the burden for coordination is placed on the planning framework which also needs to account for the uncertainty inherent in multiagent domains.

To address these challenges, in this article, we introduce a mathematical formalism that explicitly models coordination across a group of multiple navigating agents using tools from low-dimensional topology. This formalism allows us to enumerate possible coordination protocols in a symbolic fashion, enabling a robot to anticipate alternative likely futures. Furthermore, we introduce a framework that may generate Cartesian trajectory primitives corresponding to desired coordination protocols using tools from computational physics (Berger, 2001a). We use this framework to design a decentralized navigation algorithm which uses the generation of coordination primitives as a tool to couple inference and control. This enables an ego-agent to anticipate the effects of its own actions on other agents and select a navigation strategy that complies with their intentions while making progress towards its destination.

In summary, we contribute: (1) a novel topological formalism that allows us to represent, enumerate and classify distinct multiagent trajectory alternatives as symbols with analytical descriptions; (2) a centralized physics-inspired mechanism, entitled *Hamiltonian Coordination Primitives* (HCP) that generates coordinated multiagent motion, driving a set of agents from a starting configuration to a goal configuration, while satisfying topological trajectory constraints; (3) *HCPnav*, an online, decentralized, reactive navigation planning algorithm that makes use of HCP as a mechanism for simultaneous prediction and generation of motion in a multiagent navigation domain; (4) empirical

results, demonstrating the effectiveness of the HCP framework in generating multiagent trajectories of desired topological specifications across a variety of scenarios and settings; (5) empirical results, demonstrating the efficacy of HCPnav in handling a variety of experimental conditions including varying numbers of agents, heterogeneity, and environments of different geometries.

This article extends and complements our WAFR'18 conference paper (Mavrogiannis and Knepper, 2020a) by incorporating: (a) an extended review of the literature covering relevant works from the interfacing areas of multiagent simulation, social robot navigation, trajectory prediction, along with a note on the benefits of topological methods and representations for robot motion planning; (b) an extended discussion describing how this framework relates to past work on topological representations and planners for multi-agent navigation; (c) a mechanism for enabling the trajectory predictions to respect workspace boundaries of known geometry; (d) a more detailed evaluation of our trajectory-optimization pipeline (HCP) including a detailed description of the CHOMP baseline and an extensive presentation of qualitative results; (e) an extended simulated evaluation of our planning algorithm (HCPnav) in scenarios involving environments with complex geometry and agents with changing intentions.

2. Related work

From the human paradigm inspiration to the use of topological tools for symbolic inference and trajectory generation, our approach interfaces with a number of distinct research communities. We review relevant literature from the fields of multiagent simulation, social robot navigation, trajectory prediction, and topological methods for robotics applications.

2.1. Multiagent simulation

The problem of simulating smooth, collision-free, multiagent navigation scenarios has been central in a number of applications, ranging from city planning to the study of evacuation scenarios, computer game design, and robot navigation.

This area is dominated by physics-inspired models, that is, models based on the principles of particle attraction and repulsion: attractive artificial potential fields drive agents to their destinations whereas repulsive ones drive them away from others. The social force model (Helbing and Molnár, 1995) has been one of the first and most influential approaches, whereas several works have employed similar models with additional considerations such as discomfort fields (Treuille et al., 2006), local predictive processes (Hoogendoorn and Bovy, 2003; Karamouzas et al., 2009), time-to-collision heuristics (Davis et al., 2020), and cognitive heuristics (Farina et al., 2017; Moussaïd et al., 2011; Warren, 2006). Some works have employed data-driven techniques to learn the parameters of human navigation in

different contexts from simulated (Henry et al., 2010) or real-world demonstrations (Karamouzas et al., 2014).

Another prevalent planning paradigm is based on the concept of *velocity obstacles* (Fiorini and Shiller, 1998), that is, regions in the velocity space of the planning agent that would result in imminent collision, under some assumed motion model of other agents. The Optimal Reciprocal Collision Avoidance (ORCA) framework (van den Berg et al., 2009) and its variants (Alonso-Mora et al., 2012; Snape et al., 2011) are based on the principle of determining efficient robot velocities that avoid the velocity obstacles, typically under the strict assumption that others follow the exactly the same strategy.

Existing physics-inspired approaches reinforce collision avoidance as an emergent property of repeated repulsive and attractive motions between agents. This mechanism is purely local in nature, often resulting in inefficient, oscillatory multiagent behaviors: agents myopically react to the local motion of each other without reasoning about the internal decision-making processes of each other. Our framework is also physics-inspired, as we borrow our main model (a point vortex) from fluid dynamics. However, our approach incorporates an inference mechanism that explicitly reasons over coordination strategies within an agent's decision-making process. This enables agents to ground local individual motion to global multiagent navigation strategies. This allows for smoother adaptation to the intentions of each other, reflected in the more efficient and safe behaviors that our agents follow.

Existing velocity-obstacle-based approaches may guarantee collision avoidance up to a limited time horizon by explicitly exploiting the bidirectional symmetry that all agents run the same exact policy knowing that they do so (van den Berg et al., 2009). In contrast, although our HCPnav framework assumes that agents are cooperative, it does not explicitly place any constraints on their decision-making. HCPnav employs HCP to generate valid trajectory representatives of global multiagent coordination strategies. The longer horizon results in proactive adaptation that enables time-efficient and safer behavior (see Section 6).

2.2. Social robot navigation

Driven by the goal of deploying robots in crowded human environments such as pedestrian scenes, researchers have proposed multiagent navigation frameworks incorporating social considerations (Mavrogiannis et al., 2021).

A number of works focus on socially aware robot navigation. For instance, Park et al. (2012) designed a model predictive control (MPC) framework aiming at generating smooth collision-avoidance maneuvers in crowded environments. Luber et al. (2012) derived a set of dynamic navigation primitives from a human trajectory dataset and made use of them for real-time trajectory prediction and generation. Some works (Shiomi et al., 2014; Truong and Ngo, 2017) employed the social force model (Helbing and Molnár, 1995) and the reciprocal velocity obstacle model

(van den Berg et al., 2009) as predictive processes to account for the generation of humanlike and socially aware collision-avoidance behaviors.

Inspired by the cooperative nature of human navigation (Wolfinger, 1995), in recent years researchers have proposed frameworks that leverage cooperation among agents to facilitate collision avoidance. For instance, Knepper and Rus (2012) proposed a sampling-based motion planner that leverages a model of the human mechanism of civil inattention (Goffman, 1966) to reinforce commitment to mutually beneficial collision-avoidance strategies. Kretzschmar et al. (2016) learned a model of human cooperative navigation behavior using inverse reinforcement learning (IRL). Chen et al. (2017) and Kim and Pineau (2016) learned models that engineer cooperation by reproducing selected social norms such as passing from the right-hand side and overtaking on the left. Lo et al. (2019) modeled collision avoidance as a stochastic game, and guaranteed safety under different models of human pedestrian decision making. From a similar perspective, Turnwald and Wollherr (2019) proposed a game-theoretic framework for humanlike motion generation that approximates the process of collision avoidance as a Nash equilibrium in a non-cooperative, static game.

Our approach is also cooperative in nature, but unlike the majority of the literature, it employs a novel, explicit model of cooperative collision avoidance. Our model represents multiagent coordination behaviors in a compact and interpretable fashion that enables both symbolic, high-level inference and geometric, low-level motion generation. The proposed framework builds upon and extends our past work on the use of explicit representations for multiagent collision avoidance. Unlike our past work that reasons about multiagent behaviors in a purely symbolic fashion (Mavrogiannis et al., 2017; Mavrogiannis and Knepper, 2019, 2020b), this work introduces a flexible, gradient-based trajectory optimization framework that bridges the gap between symbolic reasoning and trajectory generation. Our framework directly builds on our social momentum (SM) planner (Mavrogiannis et al., 2018) which leverages the heuristic of angular momentum to infer and comply with unfolding pairwise collision-avoidance intentions. In this work, we observe that the angular momentum is a heuristic that can enforce pairwise collision-avoidance maneuvers of desired topological signature. This topological signature can be modeled using the topological invariant of the winding number. Based on this insight, we leverage work from mathematical physics (Berger, 2001a) to build a machinery for generating braided multiagent trajectories of desired winding numbers.

2.3. Trajectory prediction

Over recent years, considerable attention has been paid to the problem of multiagent trajectory prediction, motivated by modern applications ranging from people tracking to autonomous driving, and social robot navigation. The

prevailing paradigm involves the use of machine learning techniques, typically recurrent neural networks (RNNs) (Graves et al., 2013), employing features modeling interaction across agents. For example, Alahi et al. (2016) introduced a social pooling layer that enables a long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) architecture to model interaction between neighboring agents. Vemula et al. (2018) incorporated a notion of attention to enable their RNN to perceive the relative importance of neighboring agents when predicting interacting behaviors. In our past work (Mavrogiannis et al., 2017), we employed braid groups (Birman, 1975) to encode interaction among neighboring agents as a sequence of symbols and capture the qualitative traits of interaction.

The increased complexity of multiagent behavior prediction has motivated a line of work that explicitly models multimodality. For instance, Gupta et al. (2018) employed generative adversarial networks (Good fellow et al., 2014) to generate diverse predictions in a pedestrian scene. Schmerling et al. (2018) employed a conditional variational autoencoder (CVAE) to generate multimodal predictions for lane-changing scenarios involving human and robot-driven vehicles. Tang and Salakhutdinov (2019) employed a RNN model to learn multimodal multiagent trajectory predictions across a variety of driving scenarios in an unsupervised fashion. The technique of generating multimodal predictions for planning problems is not new; for instance, Mainprice and Berenson (2013) incorporated multiple human motion predictions to plan collision-free robot motion in collaborative manipulation scenarios. In fact, the origins of this idea be traced at least as far back as the development of the MPC paradigm (García et al., 1989), in which the controller iteratively evaluates the quality of a set of candidate control rollouts by passing them through a dynamic model, a cost function, and a series of constraints. In a sense, the recent literature provides a more context-aligned perspective on determining valid candidate rollouts.

Although our work does not belong to the trajectory prediction literature, we propose a computational machinery that: (a) generates multiagent trajectory primitives of desired interaction patterns; (b) inherently models multimodality in a compact and interpretable fashion using tools from topology. Our proposed multiagent trajectory primitive generation pipeline could be adapted to encode data-driven bias to match desired contexts.

2.4. Topological methods for motion planning

One of the major ongoing threads in robotics these days focuses on bridging the gap between low-level robot skills and high-level, symbolic abstractions towards improving system interpretability and facilitating access to non-expert users. One of the prevailing approaches to this problem is through the use of topological methods, which have recently been employed to a wide variety of planning problems.

One of the prevalent use cases for topological methods in robotics involves the identification of qualitatively distinct planning alternatives in cluttered environments. For instance, Knepper et al. (2012) proposed a homotopy-like equivalence relation among paths. Based on this relation, they described an algorithm that reduces the time spent on collision checking by leveraging the topological similarity between locally available paths and previously collision-checked paths. The virtues of this approach are also transferable to sampling-based motion planning: leveraging topological equivalence may help bias sampling towards regions with higher potential for collision-test survival (Knepper and Mason, 2012). From a similar perspective, Denny et al. (2020) biased sampling-based motion planning algorithms towards unexplored regions using a graph structure modeling the underlying topology of the free space in a cluttered environment. Denny et al. (2018), using a Reeb-graph abstraction of the robot's configuration space, constructed a metric that approximates the homotopic similarity between paths and demonstrate its value for homotopy-aware sampling-based motion planning. Pokorny and Kragic (2015); Pokorny et al. (2016) employed persistent cohomology and persistent homology techniques, respectively, as tools for satisfying topological constraints in sampling-based motion planning over high-dimensional configuration spaces. Bhattacharya et al. (2012) incorporated topological constraints into graph-search-based motion planning using homology as a tool. Finally, Orthey et al. (2020) introduced the fiber-bundle abstraction to formalize and tackle multilevel motion planning problems involving multiple degrees of freedom (DoFs), and employ tools from Morse theory to visualize local minima (Orthey and Toussaint, 2020).

A series of works have focused on problems involving multiple agents in static or dynamic environments. For instance, Rösmann et al. (2017) incorporated a homology-based model into a trajectory-optimization framework to generate topologically distinct trajectory alternatives. Although similar in principle to our framework, their approach treats other agents as moving obstacles without modeling their dynamics or accounting for any interaction phenomena between them. In contrast, Cao et al. (2019) incorporated global homotopy reasoning into a graph-search based approach to determine safe passages in dynamic environments with human crowds. Further, Kretzschmar et al. (2016) modeled human passing preferences as distinct homotopy classes and used them as part of a feature space used to learn a reward function representing human navigation. Diaz-Mercado and Egerstedt (2017) employed topological braids (Birman, 1975) to maximize coverage in multirobot navigation problems. In a similar domain, Denny and Fine (2020) employed Reeb graphs as models of the topological structure of the workspace to maximize the throughput of a multiagent group within a partially known environment. Hu et al. (2003) used braids as prototypes of multirobot collision-avoidance maneuvers to determine low-energy conflict resolutions among

multiple coordinating agents navigating on a plane. Our past work (Mavrogiannis et al., 2017; Mavrogiannis and Knepper, 2019, 2020b) made use of topological braids as symbols describing topological events corresponding to distinct collision-avoidance maneuvers in multiagent navigation domains. Using a probability distribution over braids, our past works have employed rule-based planners to guide agents towards paths of decreased uncertainty.

The approach presented in this article is unique in that it incorporates topological specifications in a trajectory optimization framework for dynamic multiagent environments. Unlike our past work that reasons about topological outcomes at a purely symbolic level, this article ties symbolic reasoning with a trajectory generation mechanism based on the topological invariant of the winding number. This mechanism serves as a generator of multiagent coordination primitives that are used as rollouts for a cost-based motion planner. We illustrate the value of our optimization framework across a series of challenging decentralized multiagent navigation problems.

2.5. Relation to past work

Multiagent navigation of rational agents in a bounded environment has an interesting property: agents' decision making is spatiotemporally coupled as they cannot occupy the same configuration at the same time; thus, they need to reach a state of consensus with others over a joint navigation strategy in order to avoid collisions and reach their destinations efficiently. This property may be traced at the entanglement of agents' trajectories throughout the execution of a multiagent scene. Agents' trajectories may be thought of as strings that become *knitted* around each other, according to agents' navigation strategies. This knitting has topological properties which can be studied with tools from low-dimensional topology, the field of topology focusing on topological spaces of four or fewer dimensions.

A thesis of our work (Mavrogiannis, 2019) is that understanding the topological structure that underlies multiagent collision avoidance in navigation may enable artificial agents to coordinate efficiently to avoid each other, navigate in a socially compliant fashion next to each other, and even adapt robustly to the (potentially adversarial) behaviors of heterogeneous others. Overall, we argue that topological representations can offer significant benefits to modeling, perception and planning problems in robotics, including a greater potential for generalizability across diverse domains, and inherent model explainability through the introduction of rigorously derived symbolic reasoning.

This observation has motivated us to seek computational tools to leverage the powerful abstraction capabilities of topology in problems of multiagent robot navigation. Our exploration started with the use of topological braids (Birman, 1975) as symbols representing joint strategies of collision avoidance (Mavrogiannis and Knepper, 2020b). The representation of braids allowed us to build a mechanism for inferring future collective behaviors of multiple

agents from observations of their past trajectories. Incorporating such a mechanism into the planning process of non-communicating agents was shown to result in accelerated uncertainty reduction in discrete domains under challenging scenarios (Mavrogiannis and Knepper, 2019). A data-driven approximation of this mechanism from a dataset of trajectories extracted from simulated multiagent scenarios (Mavrogiannis et al., 2017) was shown to extend these findings to continuous domains. Figure 1(a) demonstrates the inference process employed in the outlined planning architecture.

In the previous works, topology was abstractly accounted for in the design of the inference mechanism. The planning agent was reasoning about the topology in a purely symbolic and discrete fashion, being essentially agnostic to the potential geometric implications from the transfer of a topological symbol to a real-world behavior. In real-world environments, it is important for a robot to understand the geometric effects of its actions on the collective system dynamics. This motivated us to seek the foundational machinery underlying the generation of a spatiotemporally braided multiagent trajectory pattern. First, we observed that the vector of angular momentum for a

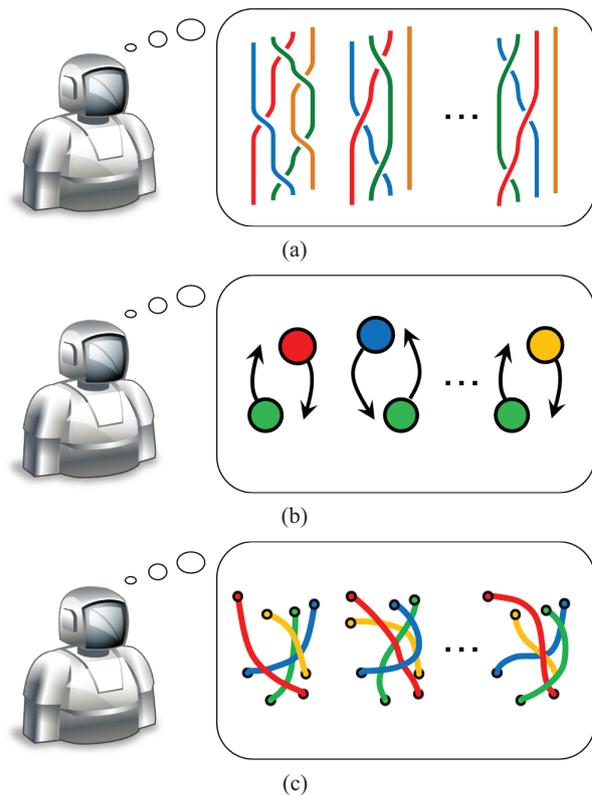


Fig. 1. Three generations of symbolic inference for multiagent navigation using topological abstractions of coupled interactions. (a) Inferring topological braids (Mavrogiannis et al., 2017; Mavrogiannis and Knepper, 2019, 2020b). (b) Inferring pairwise collision-avoidance intentions (Mavrogiannis et al., 2018). (c) Inferring coordination primitives (Mavrogiannis and Knepper, 2020a).

pair of agents that avoid a collision with each other describes their passing side intention and developed social momentum (Mavrogiannis et al., 2018), a framework that enables a navigating agent to make decisions towards maximal compliance with others in a crowded pedestrian scene, under the assumption that everyone is acting rationally. Figure 1(b) describes the inference process involved in framework.

However, in many situations, this is not the case, and it is often important for a robot to have access to alternative classes of behavior to ensure smooth and timely adaptation to potentially unexpected events or even adversarial behaviors of others. This motivated us to move a step further and build a mechanism for transitioning from symbols representing classes of multiagent behavior to Cartesian multiagent trajectory primitives (Mavrogiannis and Knepper, 2020a). This allows an agent to visualize a set of candidate futures and evaluate their second-order properties, such as their feasibility, likelihood, efficiency, and social compliance, before committing to one of them. In this article, we expand on this latter method, providing a deeper analysis and incorporating additional simulated evaluations. In contrast to the previous two main frameworks that focused on the aspects of coordination and compliance, respectively, this one targets the aspect of robustness. The inference process employed in this planner is outlined in Figure 1(c).

2.6. Contributions

In summary, we make the following contributions.

- A novel formalism of dual algebraic and geometric nature that describes the topological relationships formed across a set of trajectories in space and time. This formalism is based on the topological invariant of the winding number and can be extensible to any number of agents.
- A novel framework for multiagent trajectory generation that follows desired topological specifications. Our framework (HCP) leverages the method of Berger (2001a) for braiding two-particle trajectories into desired topological patterns across space and time.
- An extensive evaluation illustrating the ability of HCP to generate desired multiagent trajectories of desired topological specifications across a variety of challenging scenarios involving up to five agents. We show that a non-optimized implementation of our framework outperforms a baseline based on the framework of Zucker et al. (2013) in terms of topology enforcement and time efficiency.
- An online, cost-based planner (HCPnav) that leverages our trajectory generation framework, HCP, as a mechanism for the generation of valid primitives of coordination for cooperative collision-avoidance scenarios. Generating multiple alternative multiagent futures offers the potential of adaptation to unexpected

events such as the emergence of heterogeneous agents or agents with changing intentions.

- An extensive simulated evaluation illustrating the ability of HCPnav to handle a series of challenging scenarios in environments of different geometries involving up to four agents under homogeneous and heterogeneous settings. HCPnav achieves performance comparable to ORCA (van den Berg et al., 2009), a widely employed algorithm for multiagent simulation. This performance illustrates the value of HCP as a mechanism for prediction in decentralized navigation scenarios. HCP could be incorporated within a variety of modern control and planning frameworks including MPC (García et al., 1989) or reinforcement learning (Sutton and Barto, 2018).

3. Problem statement

Consider a set of $n > 1$ holonomic agents, lying at configurations $x_i \in \mathcal{X}$, $i \in \mathcal{N} = \{1, \dots, n\}$ in a planar workspace $\mathcal{X} \subseteq \mathbb{R}^2$. Agent i starts from some initial configuration $s_i \in \mathcal{X}$ and heads towards a destination d_i , $i \in \mathcal{N}$ by executing a policy $\pi_i : \mathcal{X} \rightarrow \mathcal{U}_i$, generating actions $u_i \in \mathcal{U}_i \subseteq \mathbb{R}^2$, satisfying a specification of the form:

$$u_i = \arg \min_{u_i \in \mathcal{U}_i} w_i C_d(u_i) + (1 - w_i) C_c(u_i) \quad (1)$$

where $C_d : \mathcal{U}_i \rightarrow \mathbb{R}_{\geq 0}$ represents the distance cost-to-go and $C_c : \mathcal{U}_i \rightarrow \mathbb{R}_{\geq 0}$ the collision cost of taking an action in consideration u_i , whereas $w_i \in (0, 1)$ is a weight describing agent i 's personal compromise over the two costs. Agent i is not aware of the destination d_j of agent $j \neq i \in \mathcal{N}$ but assumes that any agent $j \neq i \in \mathcal{N}$ is rational, in the sense that they also optimize for some compromise w_j between C_d and C_c . Our goal is to design a decentralized policy π_i to enable an agent i navigating under the described settings to follow a time-efficient, smooth and collision-free path.

4. HCP

Towards the goal of Section 3, in this article we contribute a novel paradigm for multiagent collision avoidance under cooperative settings (Section 3). We introduce the notion of *coordination primitives* as multiagent trajectory prototypes incorporating the property of rationality in the sense of a shared responsibility for collision avoidance across agents. In this section, we present a novel framework for generating coordination primitives in multiagent domains. Our framework HCP models pairs of interacting agents as Hamiltonian dynamical systems whose evolution follows a desired topological specification. The proposed approach is inspired by the point vortex problem (Aref, 2007) from fluid dynamics and leverages the method of Berger (2001a,b) for generating braided trajectories of multiparticle systems from topological invariants (Berger, 2001b). In the following subsections, we introduce preliminaries on

point vortex flows, review the method of Berger (2001a) and describe how we use it to generate HCPs. yielding

4.1. Hamiltonian systems

Consider a system with m DoFs lying at a state described by a set of generalized coordinates $q_j \in \mathbb{R}$ and conjugate momenta $p_j \in \mathbb{R}$, $j \in M = \{1, \dots, m\}$. Under the Hamiltonian formalism, the *state* of the system is captured into an energy function $H(q_j, p_j, t)$, $j \in M$, possibly dependent on time t , called the Hamiltonian (sum of kinetic and potential energy over all DoFs of the system). Hamilton's equations relate the evolution of the Hamiltonian to the evolution of the system state as follows:

$$\dot{q}_j = \frac{\partial H}{\partial p_j}, \quad \dot{p}_j = -\frac{\partial H}{\partial q_j}, \quad j \in M \quad (2)$$

where $\dot{(\)}$ indicates a time derivative. A dynamical system whose evolution is described by Hamilton's equations (2) is called a Hamiltonian system.

Combining the state of the j th DoF into a complex variable $z_j = q_j + ip_j$, $j \in M$, we construct an analytic function

$$F(z_1, \dots, z_m) = \Psi(z_1, \dots, z_m) + iH(z_1, \dots, z_m) \quad (3)$$

where $\Psi : \mathbb{C}^m \rightarrow \mathbb{R}$ and $H : \mathbb{C}^m \rightarrow \mathbb{R}$ is the Hamiltonian of the system assuming $\dot{H} = 0$ (i.e., the Hamiltonian is conserved). Berger (2001a) showed for this type of function, the Hamiltonian flow (2) results in motion \dot{z}_j , $j \in N = \{1, \dots, n\}$, that follows the Wirtinger derivative (Gunning and Rossi, 1965; Kaup et al., 1983) of Ψ with respect to z_j . Therefore, the collective Hamiltonian motion of all DoFs follows the direction of maximum increase of Ψ . Leveraging this construction, Berger (2001a) generated multiparticle trajectories of desired topological specifications by setting Ψ to be a *topological invariant* (Berger, 2001b), i.e., a function mapping the topological properties of the system flow to a real number. In the following sections, we build upon this method to generate braided multiagent trajectories using a topological invariant called the *winding number*.

4.2. The Winding Number

Consider a closed curve $\gamma : [0, T] \rightarrow \mathbb{C} \setminus \{0\}$ with $\gamma(0) = \gamma(T)$ and define a function:

$$\lambda(t) = \frac{1}{2\pi i} \oint_{\gamma} \frac{dz}{z} \quad (4)$$

where $z = \gamma(t)$, $t \in [0, T]$. We can express γ in polar coordinates as $\gamma(t) = r(t)e^{i\theta(t)}$, where $r(t) = \|\gamma(t)\|$ and $\theta(t) = \angle\gamma(t)$. Using the Cauchy integral formula, we can decompose (4) into the sum

$$\lambda(t) = \frac{1}{2\pi i} \int_0^t \frac{\dot{r}}{r} dt' + \frac{1}{2\pi} \int_0^t \dot{\theta} dt' \quad (5)$$

$$\lambda(t) = \frac{1}{2\pi i} \log\left(\frac{r(t)}{r(0)}\right) + \frac{1}{2\pi} (\theta(t) - \theta(0)) \quad (6)$$

The real part of this integral,

$$w = \text{Re}(\lambda(t)) = \frac{1}{2\pi} (\theta(t) - \theta(0)) \quad (7)$$

is a *topological invariant*, counting the number of times the curve γ encircled the origin in the time interval $[0, t]$. Intuitively, any continuous curves that encircle the origin the same number of times are mapped to the same winding number value. Taken across the whole curve γ from $t = 0$ to $t = T$, the imaginary part is equal to zero and thus the complex function degenerates to $\lambda(t) = w$ because the curve is closed, i.e., $r(0) = r(T)$.

By construction, the function λ fits the general form of function F from Section 4.1 because it can be expressed as a function over $z = \gamma(t) \in \mathbb{C}$. Thus, as discussed in Section 4.1, if we carefully set its imaginary part to a Hamiltonian function describing a dynamical system, the evolution of the system would maximize the growth of the real part, the winding number over the curve γ . As described by Berger (2001a), the imaginary part of the function λ indeed corresponds to the Hamiltonian function for a particular dynamical system: a *point vortex*.

4.3. Two-particle vortex motion

In fluid mechanics, a two-dimensional point vortex (Aref, 2007; Nitsche, 2006) represents an entity that induces rotational motion around an axis. It is typically described by a quantity called *vorticity*, measuring the rate of local fluid rotation. Consider two point vortices placed at positions $a = (a_x, a_y) \in \mathbb{R}^2$ and $b = (b_x, b_y) \in \mathbb{R}^2$. We define a complex function $\gamma_{ab} : [0, T] \rightarrow \mathbb{C}$ to track the relative motion of the two vortices by setting $\gamma_{ab}(t) = r_{ab}(t)e^{i\theta_{ab}(t)}$, where $r_{ab} = \|a - b\|$ and $\theta_{ab}(t) = \angle\gamma_{ab}(t)$. Assuming equal, unit vorticity, we can write the Hamiltonian for the system of vortices as

$$H = -\frac{1}{2\pi} \log r_{ab} \quad (8)$$

The equations of motion can be derived as

$$\begin{aligned} (\dot{a}_x, \dot{a}_y) &= \left(\frac{\partial H}{\partial a_y}, -\frac{\partial H}{\partial a_x} \right) \\ &= \frac{1}{2\pi} \left(-\frac{a_y - b_y}{r_{ab}^2}, \frac{a_x - b_x}{r_{ab}^2} \right) \end{aligned} \quad (9)$$

$$\begin{aligned} (\dot{b}_x, \dot{b}_y) &= \left(\frac{\partial H}{\partial b_y}, -\frac{\partial H}{\partial b_x} \right) \\ &= \frac{1}{2\pi} \left(-\frac{b_y - a_y}{r_{ab}^2}, \frac{b_x - a_x}{r_{ab}^2} \right) \end{aligned} \quad (10)$$

The two vortices rotate about each other at a constant radius in a counterclockwise direction. We may control the

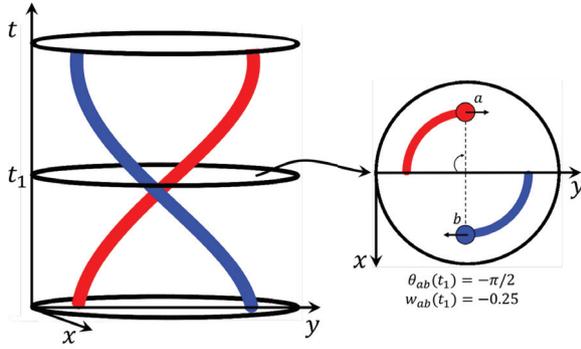


Fig. 2. Spacetime plot of the trajectories of two agents, navigating in a circular workspace (left) and projection of their trajectories until time t_1 , onto the xy plane, along with the definition of their pairwise winding angle and winding number (right).

directionality of the rotation by switching the signs in the right-hand side of (9) and (10).

Similarly to Section 4.2, we define a complex function λ_{ab} as

$$\lambda_{ab}(t) = \frac{1}{2\pi i} \log(r_{ab}) + \frac{1}{2\pi} (\theta_{ab}(t) - \theta_{ab}(0)) \quad (11)$$

The real part

$$w_{ab}(t) = \text{Re}(\lambda_{ab}(t)) = \frac{1}{2\pi} (\theta_{ab}(t) - \theta_{ab}(0)) \quad (12)$$

corresponds to the pairwise winding number of the two curves, counting the number of times the two vortices have rotated about each other (see Figure 2 for a graphic representation of the pairwise winding number). We remark that $\text{Im}(\lambda_{ab}) = H$ and, therefore, according to Section 4.1, the Hamiltonian flow for this system maximizes the growth of the real part, i.e., the corresponding winding number $\text{Re}(\lambda_{ab}) = w_{ab}$.

4.3.1. Cooperative collision avoidance as two-particle vortex motion. In this article, we propose to leverage outlined two-particle vortex dynamics to generate cooperative collision-avoidance maneuvers. Hamilton's equations (9) and (10) describe dynamics corresponding to control laws that can be used to generate coordinated collision-avoidance motion primitives for two agents. In particular, given a desired direction of collision avoidance, expressed in the sign of the winding number w_{ab} ($\text{sign}(w_{ab}) > 0$ denotes right-hand side collision avoidance and vice versa), by multiplying the right-hand sides of (9) and (10), we recover control laws that yield coordinated trajectories for a and b along the direction indicated by $\text{sign}(w_{ab})$.

Under the assumption of rationality, as outlined in Section 3, the motion primitives produced by following the outlined control laws may serve as a prediction mechanism, valuable for simultaneous inference and planning in

cooperative multiagent domains. For instance, in a hallway scenario where two agents attempt to converge on a passing side, the proposed mechanism could allow an agent to anticipate and adapt to either outcome or even attempt to enforce its own preference in uncertain situations. In the following section, we detail the outlined idea and describe how it can be adapted to domains with $n \geq 2$ agents.

4.4. Generating HCPs

Consider the problem of driving n agents from a set of initial configurations $S = (s_1, \dots, s_n) \in \mathbb{R}^{2n}$ to a set of destinations $D = (d_1, \dots, d_n) \in \mathbb{R}^{2n}$ in a collision-free fashion, while satisfying pairwise passing-side specifications as described in a vector $w = (w_{12}, w_{13}, \dots) \in \mathcal{W}$. Assuming that each agent passes each other exactly once on its way to its destination (agents do not loop around others), the magnitude of w_{ij} , $i \neq j \in N$, is not important; thus, in the remainder of the article, we abuse the notation of w_{ij} to denote $\text{sign}(w_{ij})$. The cardinality of the set of possible specifications is $|\mathcal{W}| = 2^{\binom{n}{2}}$, corresponding to all possible combinations of passing sides for all agents. It should be noted that although all combinations in \mathcal{W} are topologically possible, in practice, only a subset of them are meaningful and likely given agents' state history and under the assumption of rationality. Section 5 addresses the problem of evaluating the likelihood and the feasibility of a topological specification.

We now describe a policy $\pi_i : \mathbb{R}^{2n} \times \mathcal{W} \rightarrow \mathbb{R}^{2n}$ that can be sequentially iterated across all agents to produce a multiagent trajectory that satisfies a topological specification w . The policy leverages Hamilton's equations as described in Section 4.3.1, and we refer to it as HCP. HCP prescribes an action $u_i \in \mathbb{R}^2$ to every agent $i \in N$, synthesized from a weighted consideration of all pairwise collision-avoidance reactions between the agent and all others, towards meeting the pairwise specifications contained in w :

$$u_i = v_i \cdot k \left(u_{\text{att}}^i + u_{\text{rep}}^i \right) \quad (13)$$

where $v_i \in \mathbb{R}$ (m/sec) is an agent's desired speed, u_{att}^i , u_{rep}^i are actions (velocity vectors in meters per second) attracting the agent towards its destination and repulsing it from others, respectively, and $k \in \mathbb{R}$ (seconds per meter) is a scaling parameter. The action

$$u_{\text{att}}^i = k_{\text{att}}(d_i - q_i) \quad (14)$$

attracts the agent from its current state q_i towards its destination d_i according to an importance weight $k_{\text{att}} \in \mathbb{R}$ (per second). The action

$$u_{\text{rep}}^i = k_{\text{rep}} \sum_{j \neq i}^N c_{ij} w_{ij} v_j^i \quad (15)$$

repulses agent i from each other agent $j \in N, j \neq i$, through the velocity v_j^i (m/sec), derived from (9) and (10), with a degree of consideration equivalent to the *criticality* ^{i} of their pairwise collision avoidance, expressed by the unitless $c_{ij} \in \mathbb{R}$ (the closer two agents are, the more critical their avoidance becomes) and along the direction indicated by w_{ij} whereas $k_{rep} \in \mathbb{R}$ is a unitless importance weight. The choice of the weighting factors k_{att}, k_{rep} expresses the relative significance between goal attraction and collision avoidance, whereas the criticality term is a function of the distance between agents i and j . By sequentially executing the outlined policy, in parallel for all agents, in equal time steps of length dt , the system of agents is forced to follow the specification w . The policy is executed repeatedly until all agents reach their destinations. Note that this method does not guarantee that the desired topology will be achieved. Depending on the number of agents, their initial configurations and intended destinations, the criticality model and importance parameters, it may be challenging to balance collision-avoidance directionality with goal reaching. However, we empirically observe desirable performance at a low computational cost in a case study exploring scenarios with different numbers of agents (see Section 6).

5. Decentralized navigation with HCP

In this section, we describe a decentralized algorithm that leverages the HCP framework as a prediction mechanism to tackle the navigation problem posed in Section 3. Our algorithm is essentially a cost-based planner operating on HCPs. The algorithm comprises the following sequence of actions: (1) **predict** the destinations of other agents; (2) **generate** a set of candidate multiagent trajectories that drive agents from their current locations to their predicted destinations; (3) **evaluate** candidates with respect to a cost function; (4) **execute** the next action assigned to the planning agent from the lowest-cost candidate. In the following subsections, we describe the main components of the algorithm and provide a detailed presentation of it in pseudo-code format (see Algorithm 1).

5.1. Destination prediction

In Section 4.4, it was assumed that the planning policy has access to the destinations of other agents. In the settings we are considering (see problem posed in Section 3), no explicit communication takes place among agents, and therefore agents are not aware of each other's destinations. Thus, a planning agent needs to make a prediction about the destinations of others in order to use the HCPs framework as a prediction mechanism. However, in practice, an agent only interacts with others for as long as they lie within its sensing range, which for current robotic systems is quite limited. During this amount of time, other agents' observed behaviors may or may not be revealing about their specific

Algorithm 1 HCPnav(q, d, Ξ)

Input: map , representation of the workspace boundary; q , agent's current state; d , agent's intended destination; Ξ_{past} , state history of all agents; K , number of outcomes to consider; ϵ , desired distance-to-goal threshold.

```

1:  $AtGoal \leftarrow False$ 
2: while  $\neg AtGoal$  do
3:    $\mathcal{R} \leftarrow Get\_Reactive\_Agents(\Xi_{past})$ 
4:    $D \leftarrow Predict\_Destinations(\Xi_{past}, \mathcal{R})$ 
5:    $\mathcal{W} \leftarrow Get\_Outcomes(\mathcal{R})$ 
6:    $P \leftarrow Get\_Outcome\_Probability(\mathcal{W}, \Xi_{past})$ 
7:    $\mathcal{W}_K \leftarrow Get\_BestOutcomes(P, \mathcal{W}, K)$ 
8:    $\mathcal{Z} \leftarrow \emptyset$ 
9:   for all  $w \in \mathcal{W}_K$  do
10:     $\Xi_{pred} \leftarrow HCP(\Xi_{past}, w, D)$ 
11:     $\mathcal{Z} \leftarrow \{\mathcal{Z}, \Xi_{pred}\}$ 
12:   end for
13:    $C \leftarrow Score\_Trajectories(\mathcal{Z})$ 
14:    $u \leftarrow Get\_Best\_NextAction(C, \mathcal{Z})$ 
15:    $q \leftarrow Execute\_Action(u)$ 
16:   if  $\|q - d\| < \epsilon$  then
17:      $AtGoal \leftarrow True$ 
18:   end if
19: end while
20: return None

```

destination. In fact, detailed predictions of agents' destinations may not be sufficiently informative regarding agents' future behaviors; in crowded environments, the collision-avoidance process is a more significant influence over agents' behaviors. For this reason, we take a more practical approach, focusing on coarse predictions of agents' future locations. Alternative methods of filtering could be employed to provide more accurate destination prediction; however, this is not our focus in this article and, as will be shown in Section 6, this simplified model may yield the desired performance.

In particular, we assume that an agent's sensing range has the shape of a disk of radius R , centered at the agent's position, q_i . Any agent lying outside of this disk is not perceived by the agent whereas any agents lying behind the robot are ignored at the planning stage. For each one of the perceived and actively considered agents, we approximate their intended direction of motion by fitting a line to their recent, observed trajectory and projecting their current velocity on it. We then propagate their current speed along this direction until it intersects the boundary of the sensing disk. For our planning algorithm, that point of intersection is considered to be that agent's destination (see Figure 3). This prediction is expected to be a coarse approximation of where an agent is heading. However, because our algorithm runs in replanning cycles, this approximation provides a sufficient amount of detail for the HCP prediction mechanism. This mechanism makes use of the assumption that agents act rationally, that is, agents' behaviors are driven by an incentive of making collision-free progress towards their destinations.

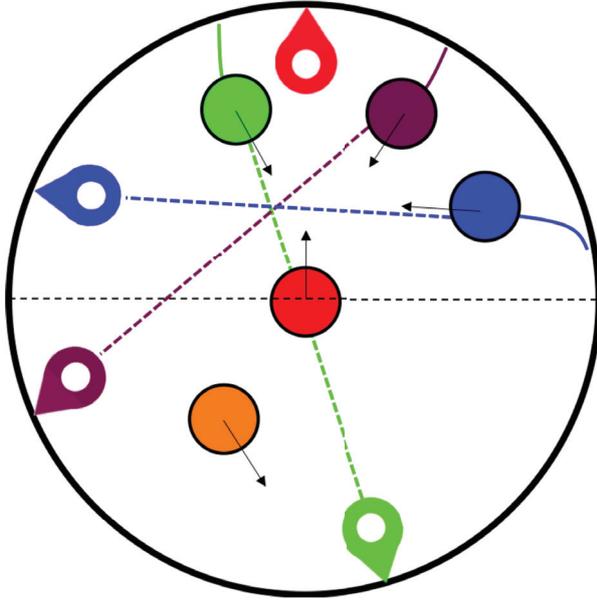


Fig. 3. The destination prediction mechanism. The red agent makes destination predictions for all agents, lying within its circular sensing disk, and in front of it.

5.2. Outcome evaluation

The set \mathcal{W} contains symbolic representations of topologically distinct *outcomes* for the system of all considered agents. Naturally, a significant question that arises is: which outcome should the planning agent trust and follow? We approach this problem with the following sequence of computations: (1) we first evaluate an outcome with respect to its likelihood; (2) we then generate trajectory representations for the set of the K most likely outcomes $\mathcal{W}_K \subset \mathcal{W}$, using the policy presented in Section 4.4; (3) finally, we evaluate these K best outcomes with respect to the physical properties of their trajectory representations.

5.2.1. Probability of an outcome. An outcome is initially encoded symbolically as a tuple \mathbf{w} that prescribes how agents avoid each other throughout the course of the scene. From a topological perspective, these symbols are independent of each other; any motion is allowed even if it is not efficient. However, from a real-world point of view, the collision-avoidance strategies that agents employ to avoid one another are coupled and modeling the complex probabilistic relationships among them is a challenging problem. For our purposes in this article, we are interested in finding a way to bias our search towards the outcomes that are more likely to occur. We do so by using the following expression:

$$P(\mathbf{w}|\Xi_{\text{past}}) = P(w_{12}, w_{13}, \dots | \Xi_{\text{past}}) \propto \frac{1}{Z} \prod_{ij} P(w_{ij}|\Xi_{\text{past}}) \quad (16)$$

where Ξ_{past} denotes agents' past trajectories and Z is a normalization constant across all $\mathbf{w} \in \mathcal{W}$. This expression was

derived by factorizing $P(w_{12}, \dots | \Xi_{\text{past}})$ using the product rule and then substituting each factor with its Bayes' rule expression. Drawing from our past work (Mavrogiannis and Knepper, 2019), we model $P(w_{ij}|\Xi_{\text{past}})$ by employing the physical quantity of angular momentum. For two agents i, j , navigating on a plane, their angular momentum L^{ij} lies along the z axis. Note that the sign of the z component of the momentum, L_z^{ij} is an indicator of agents' passing side and thus of the winding number of their trajectories w_{ij} , with $L_z^{ij} > 0$ indicating the emergence of $w_{ij} > 0$ (right-hand side collision avoidance) and $L_z^{ij} < 0$ indicating the emergence of $w_{ij} < 0$ (left hand side collision avoidance). We incorporate the momentum as a heuristic in a sigmoid model as follows:

$$P(w_{ij}|\Xi_{\text{past}}) = \frac{1}{1 + \exp(-w_{ij}k_l L_z^{ij})} \quad (17)$$

where $k_l \in \mathbb{R}$ (meters squared per second) is a normalization constant making the argument of the exponential dimensionless. The greater $|L_z^{ij}|$ is, the greater the mutual intention or preference of agents i and j over a collision avoidance along the direction of L^{ij} is.

5.2.2 Trajectory quality. We evaluate a trajectory representation $\Xi_{\mathbf{w}}$ of an outcome \mathbf{w} by computing its total energy $\mathcal{E} : \mathcal{Z}^n \rightarrow \mathbb{R}$, its required immediate acceleration $\mathcal{A} : \mathcal{Z}^n \rightarrow \mathbb{R}$ and its safety cost $\mathcal{S} : \mathcal{Z}^n \rightarrow \mathbb{R}$. The energy measure (sum of squared speeds throughout the trajectory) gives an estimate of the efficiency of an outcome whereas the acceleration measure is indicative of the aggressiveness of the maneuvers required to comply with an outcome. We model the safety cost as $\mathcal{S}(\Xi) = \exp(-k_d d_{\min})$, where $d_{\min} \in \mathbb{R}$ is the minimum distance between any pair of agents in a trajectory Ξ , and $k_d \in \mathbb{R}$ (per meter) is a normalization constant making the argument of the exponential dimensionless. Note that other cost functions could be used to incorporate different considerations such as social comfort (see, for example, Sisbot et al., 2007).

5.3. Decision making

We first rank outcomes at a symbolic level through the use of the probability distribution, presented in Section 5.2 and determine the set of the K most likely outcomes \mathcal{W}_K . Then, we determine the outcome of lowest cost:

$$C(\Xi) = \alpha_e \mathcal{E} + \alpha_a \mathcal{A} + \alpha_s \mathcal{S} \quad (18)$$

where α_e , α_a , and α_s are importance weights and, finally, extract the optimal outcome through the following optimization scheme:

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathcal{W}_K} C(\Xi_{\mathbf{w}}) \quad (19)$$

The planning agent executes the next action assigned to it from the trajectory of lowest cost $\Xi_{\mathbf{w}^*}$. Figure 4 depicts a graphic representation of the planning scheme.

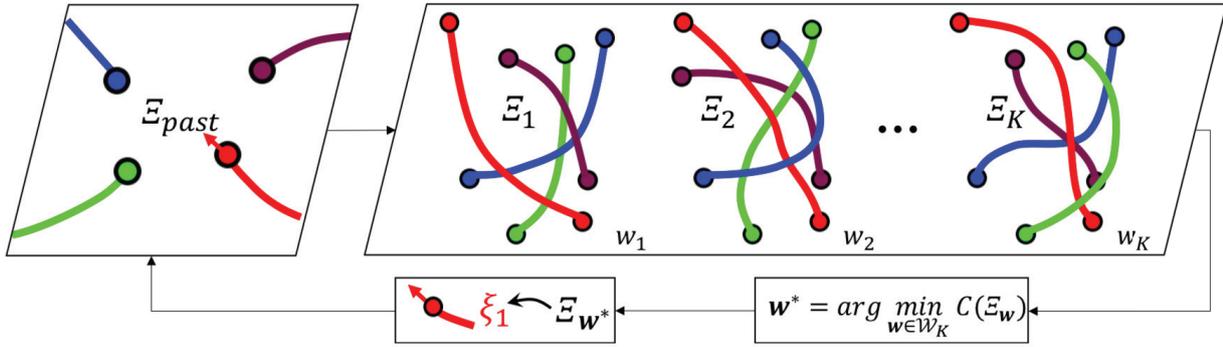


Fig. 4. Illustration of the planning scheme. At every replanning cycle, the planning agent generates a set of diverse (topologically distinct) predictions about the joint future behavior of all agents, evaluates them with respect to a cost function C , and executes the action assigned to it from the prediction of lowest cost.

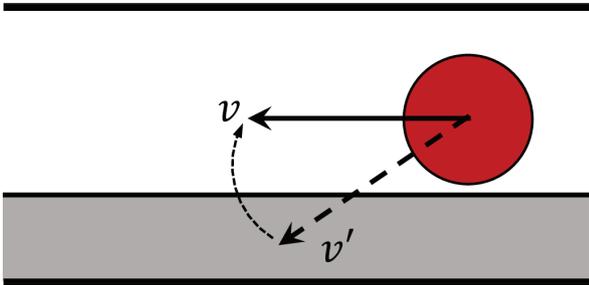


Fig. 5. Example of conforming an agent's planned velocity to a boundary constraint. The agent, represented as a red disk, has initially planned a velocity v' . If v' gets executed, it will drive the agent to a region that is too close to the boundary (gray area). Instead, the agent executes velocity v , extracted by projecting v' onto the environment boundary, directing it towards the direction of maximum increase to agent's destination (here, to the left) and setting $\|v\| = \|v'\|$.

5.4. Conforming to workspace boundary

The technique described in Section 4.4 does not take into consideration workspace boundary constraints. We account for them at execution time by projecting the planning agent's selected velocity onto the boundary, along the direction of maximum progress to goal. Figure 5 depicts an example of the mechanism. The agent has planned a velocity v' which would drive it to a region that is too close to the boundary (represented as the gray area). Instead, the agent executes a velocity v , extracted by projecting v' onto the environment boundary, directing it toward the direction of maximum increase to agent's destination (here, to the left) and keeping the same magnitude, that is, $\|v\| = \|v'\|$.

5.5. Pseudocode

Algorithm 1 summarizes the described algorithm, HCPnav, in pseudocode format. The algorithm runs in replanning cycles for as long as the Boolean variable *AtGoal* is set to

False, indicating that the agent has not reached its destination yet. At every cycle, the agent first determines a set of *reactive agents*, that is, agents that lie within the robot's sensing disk and to the front of the robot's heading (function `Get_Reactive_Agents`). Then, function `Predict_Destinations` outputs predictions for the destinations of the reactive agents and `Get_Outcomes` returns a set of topological representations for outcomes that could emerge in the remainder of the execution. Function `Get_Outcome_Probability` returns the probability for each of the outcomes considered and function `Get_Best_Outcomes` returns the K best outcomes. Function `HCP` executes the HCP policy and generates trajectory representations for these outcomes and function `Score_Trajectory` evaluates them with respect to the cost function considered. Finally, function `Get_Best_Next_Action` returns the next action for the planning agent from the trajectory of lowest cost and function `Execute_Action` executes that action. The distance between the resulting agent state and its destination is compared to the predefined threshold ϵ and the flag *AtGoal* is updated to *True* in case the agent is sufficiently close to its destination.

5.6. Complexity and practical considerations

The most computationally intense component of our algorithm is the estimation of the outcome probabilities. For n agents, this computation runs in time $O(2^{n^2})$; the remainder of the computations run in polynomial time. In practice, a replanning cycle of HCPnav on a scenario involving 4 agents and thus the evaluation of 64 topological classes with $K=5$, runs at an average of 42 ms, with the worst case being 203 ms in a non-optimized Matlab implementation on a MacBook Pro of 2015 with an Intel Core i7 processor of 2.5 GHz, running macOS High Sierra. Transfer to a faster language and optimization of parts of the code could help vastly improve performance.

Under the current design, scaling to large n is not possible. However, for a mobile robot application, we argue that

Table 1. Success rate of HCP in generating the desired trajectory topology across three conditions. For each condition, the success rate is computed within 100 randomly generated scenarios.

	Condition			
	2 agents	3 agents	4 agents	5 agents
Number of outcomes	2	6	64	1024
Success (%)	1	99.75	89.70	65.48

it is also not practical. The sensing limitations would prohibit the emergence of a large number of agents. Even if more agents are sensed, pruning them to the subset of directly reactive agents is a motivated and human-inspired way of reducing the load. Future work involves the design of an online data-driven topology-classification mechanism that would enable agents to directly estimate the most likely candidates, without brute-forcing their evaluation.

6. Evaluation

In this section, we present two main studies evaluating the main technical components of our work: the HCP trajectory generation framework and HCPnav decentralized navigation algorithm.

6.1. HCP performance

We characterize the ability of HCP to generate multiagent trajectories of desired topological specifications. We present empirical evidence illustrating the scaling properties and perform a comparative study against a trajectory-optimization framework.

6.1.1. Scaling Properties. We demonstrate the performance of HCP in generating topologically distinct, multiagent navigation trajectories. We consider 4 different conditions, corresponding to different numbers of agents (2, 3, 4, and 5 agents), navigating in a circular workspace of radius 2.5 m (agents are represented as disks of radius 0.3 m). For each condition $n \in \{2, 3, 4, 5\}$, we randomly generate 100 distinct scenarios, by assigning agents initial and final locations that lead to challenging multiagent encounters, requiring competent collision-avoidance maneuvers. We

execute each scenario, $2 \binom{n}{2}$ times, each with a distinct topological specification. We measure the success rate of the planner in generating the desired topology under all conditions considered and report it in Table 1 (a trial is considered successful if the planner was able to produce all of the distinct topologies). The planner parameters were kept constant across conditions and scenarios. It can be observed that the planner performance drops as the number of agents n increases. The method becomes more sensitive to

parameter tuning, as the effects of the chaotic nature of the vortex problem (Aref et al., 1989) become more significant.

6.1.2. Comparison with trajectory optimization. To the best of the authors' knowledge, this is the first work to consider the problem of generating multiagent trajectories following desired topological specifications across space and time in tasks involving multiple interacting agents moving between arbitrary regions of a shared workspace. To illustrate the efficacy of our approach, we present a comparative evaluation against a baseline framework based on the CHOMP (covariant Hamiltonian optimization for motion planning) trajectory-optimization framework (Zucker et al., 2013).

Although the usecases of CHOMP in the literature focus on single-robot trajectory optimization problems, we present a multiagent extension that accounts for multiagent navigation scenarios by treating agents' configurations as additional independent DoFs (decision variables in the optimization scheme). Our CHOMP baseline adapts the originally proposed *smoothness* and *obstacle* cost functionals to equivalent multiagent formulations. In addition, we introduce a novel cost functional that quantifies the violation of a specified topological specification. To ensure a fair comparison against HCP, this cost makes use of the same machinery to enforce a topological specification: the *winding number*.

We formulate the problem of multiagent trajectory generation under topological specifications as an unconstrained optimization scheme:

$$\Xi^* = \arg \min_{\Xi \in \mathcal{Z}} \mathcal{C}(\Xi) \quad (20)$$

where $\mathcal{C} : \mathcal{Z} \rightarrow \mathbb{R}$ is a real cost functional taking as input a multiagent trajectory $\Xi \in \mathcal{Z}$. The cost is defined as

$$\mathcal{C}(\Xi) = w_{\text{sm}} \mathcal{F}_{\text{sm}}(\Xi) + w_{\text{obs}} \mathcal{F}_{\text{obs}}(\Xi) + w_{\text{top}} \mathcal{F}_{\text{top}}(\Xi) \quad (21)$$

The costs \mathcal{F}_{sm} and \mathcal{F}_{obs} are adapted from the CHOMP framework (see the work of Zucker et al. (2013) for details). In particular, $\mathcal{F}_{\text{sm}} : \mathcal{Z} \rightarrow \mathbb{R}$ is a functional quantifying the trajectory smoothness cost as a sum of the individual trajectory smoothness costs:

$$\mathcal{F}_{\text{sm}}(\Xi) = \frac{1}{2} \sum_{i=1}^n \int_0^1 \left\| \frac{d}{dt} \xi_i(t) \right\|^2 dt \quad (22)$$

whereas $\mathcal{F}_{\text{obs}} : \mathcal{Z} \rightarrow \mathbb{R}$ is a functional quantifying the overall trajectory clearance, defined as a sum of the clearance costs for all pairs of agents, multiplied by their relative speed:

$$\mathcal{F}_{\text{obs}}(\xi) = \sum_{ij} \int_0^1 c(\xi_i(t), \xi_j(t)) \times \left\| \frac{d}{dt} (\xi_i(t) - \xi_j(t)) \right\| dt \quad (23)$$

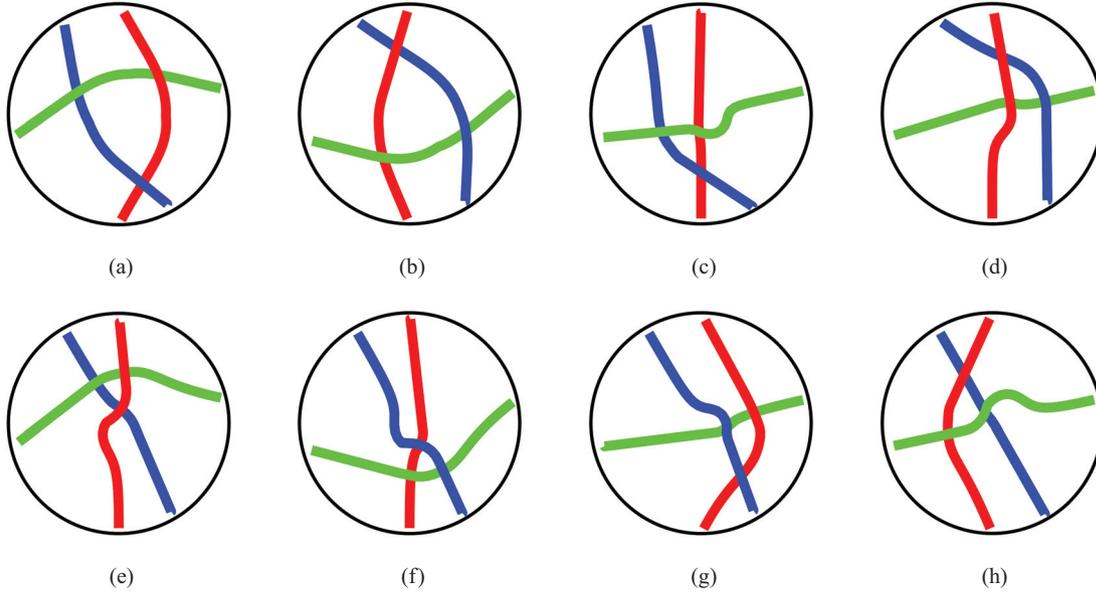


Fig. 6. Top view of trajectories generated by executing the same 3-agent scenario with all possible topological specifications. Topology tuple that was used as a specification for each execution: (a) $(1, 1, 1)$; (b) $(-1, -1, -1)$; (c) $(1, -1, 1)$; (d) $(-1, 1, -1)$; (e) $(1, 1, -1)$; (f) $(-1, -1, 1)$; (g) $(1, -1, 1)$; (h) $(-1, 1, -1)$.

Finally, we introduce a topology enforcement functional, using the definition of the winding number:

$$\mathcal{F}_{\text{top}}(\xi) = \frac{1}{2} \sum_{ij} \left(\lambda_{ij}(1 - \epsilon) - \lambda_{ij}^* \right)^2 \quad (24)$$

where $\lambda_{ij}(1 - \epsilon)$ denotes the winding number resulting from the motion of agents i and j , from time $t=0$ until time $t=1 - \epsilon$ and λ_{ij}^* denotes a specification for the winding of the trajectories of these agents (0.5 for right-hand side collision avoidance and -0.5 for left-hand side collision avoidance). The parameter ϵ is introduced to define a timing before agents' trajectories reach their final endpoints (destinations). This is done to allow for a smoother convergence during the optimization process: instead of jumping between discrete values, it allows the topology functional to take values in a continuous domain.

To illustrate the difficulty of automatically synthesizing multiagent trajectories of desired topological specifications through trajectory optimization techniques, we consider a simple case study, in which we compare the performance of HCP with the performance of the outlined CHOMP baseline (Zucker et al., 2013). We randomly generate 500 different scenarios involving 2 agents navigating towards opposing sides of a circular workspace (workspace has $5m$ diameter, starting positions are uniformly distributed along the circumference, speed normally distributed between 0.3 and 1.5 m s^{-1} for each agent). For each scenario, we randomly sample a passing side that agents should pass one another from. We execute this scenario with both HCP and CHOMP, considering the passing side as an additional specification to the problem.

Table 1 describes the performance of the two approaches, which is measured with respect to success rate and computation time (non-optimized Matlab implementation on a MacBook Pro of 2015 with an Intel Core i7 processor of 2.5 GHz, running macOS High Sierra). For CHOMP, a trial is considered successful if it generates trajectories of the desired topology within 500 iterations whereas for HCP, a trial is considered successful if the desired topology is achieved once the agents reach their destinations. It can be observed that HCP dominates with a success rate of 98.40% (corresponding to 492/500 successful trials). The computation time is comparable in terms of interactions but HCP requires almost two orders of magnitude less time in seconds. The benefits provided by HCP in terms of success rate and computation time make the consideration of multiple trajectory topologies at planning time a more practical strategy.

6.1.3. Qualitative results. We illustrate the efficacy of HCP through qualitative results acquired in challenging example scenarios. In particular, we take the following approach: we select a randomly generated multiagent navigation scenario involving n agents, and execute it $|\mathcal{W}| = 2^{\binom{n}{2}}$ times, each corresponding to a distinct topology from \mathcal{W} .

Figure 6 depicts the trajectories generated by HCP under all possible topologies (eight) for the same scenario (same starting and final configurations). Figure 7 depicts the trajectories generated by HCP for the same 4-agent scenario, executed for all possible topologies (64). We verified that for both examples, HCP was able to generate correctly all desired topologies.



Fig. 7. Top view of trajectories generated by executing the same four-agent scenario with all possible topological specifications. The subcaptions denote the topology tuple that was used as a specification for each execution.

6.2. HCPnav performance

At planning time, HCPnav uses HCP to generate multiple candidate futures represented as multiagent trajectories, evaluates them with respect to trajectory quality as discussed in Section 5, and follows the future of minimum

cost. Figure 8 shows examples of the decision making performed by HCPnav at during a navigation experiment in scenarios involving two, three, and four agents.

To demonstrate the virtues of HCPnav (Algorithm 1), we perform a series of simulation studies. We first

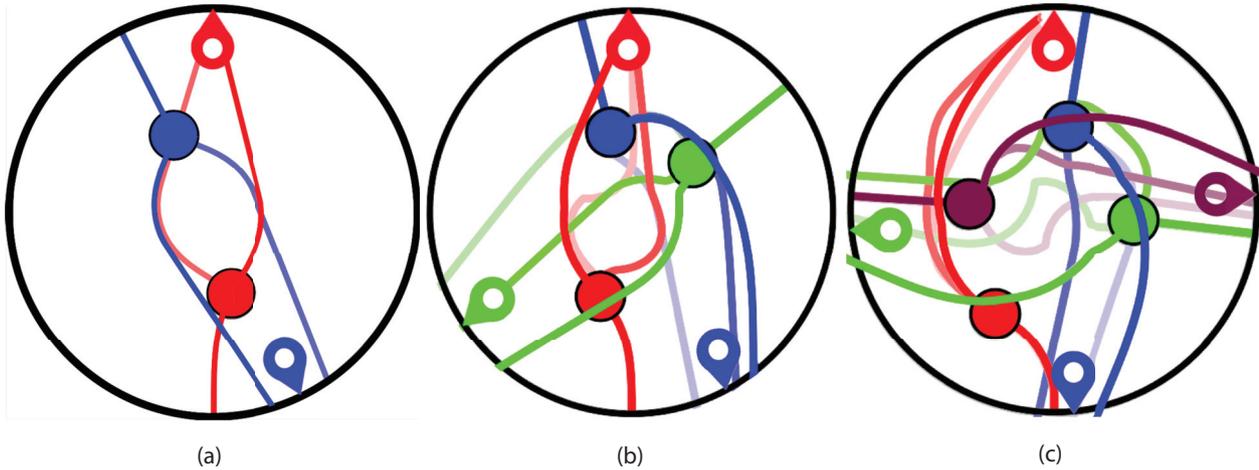


Fig. 8. Overlaid predictions made by a HCPnav agent (red color) as it navigates towards the red landmark in environments with (a) two, (b) three, and (c) four agents.

characterize the scaling properties of the algorithm, considering a series of scenarios involving different numbers of agents and execution settings. Next, we explore the applicability of HCPnav in environments with boundaries of different geometries.

6.2.1. Scaling properties. We investigate the ability of HCPnav to handle different execution settings. We focus on an environment of fixed geometry (a circle with a radius of 5 m) and consider nine different experiment configurations, defined by varying the number of agents (two, three, and four agents) and the policy profile (the mixture of policies agents are running). We consider three different policy profiles: (a) a homogeneous profile under which all agents run the same planner; (b) a heterogeneous condition under which one agent runs our planner and others are *inattentive*, moving straight to their goals without avoiding collisions; (c) a heterogeneous condition in which one agent runs our planner and others are *uncertain*, changing intentions over a destination twice, without avoiding collisions. Note that the two latter cases are particularly challenging for decentralized planners, as a typical assumption they rely heavily on is homogeneity.

For reference, we execute the same scenarios upon replacing HCPnav with a baseline: ORCA (van den Berg et al., 2009) (setting clearance and speed parameters to reproduce qualitatively similar considerations as HCPnav). We selected ORCA as it constitutes a decentralized framework for multiagent simulation with a readily available code implementation and a thorough documentation that is widely employed as a baseline in multirobot planning research.

We quantify the performance of each *reactive* agent (HCPnav or ORCA) with respect to four aspects of trajectory quality: (a) experiment time, measured as the amount of time that the last reactive agent took to reach its destination; (b) safety, measured as the minimum distance between

any two agents under homogeneous settings, and as the minimum distance between a reactive agent and any other agent under heterogeneous settings; (c) path efficiency, measured as the ratio between the length of the optimal path to goal and the length of the path that a reactive agent followed (averaged over the number of agents in the homogeneous case); (d) trajectory acceleration, measured as the average acceleration per time step per reactive agent throughout the experiment. Figure 9 depicts the performance of HCPnav and ORCA under the outlined experimental configurations in terms of the metrics considered. For each configuration (number of agents and policy profile), each planner executed the same set of 200 randomly generated scenarios.

Overall, we observe that HCPnav outperforms ORCA under homogeneous settings. In particular, HCPnav is significantly more time-efficient and safer even as the number of agents increases. These features of HCPnav could be attributed to the implicit consensus that is reached earlier through the consideration of joint strategies of collision avoidance. HCPnav accomplishes this while remaining path-efficient (>80% at all times). However, the collision-avoidance maneuvers generated by HCPnav require higher accelerations to implement (see Figure 8 for an illustration of trajectories generated by HCPnav). In parallel, ORCA dominates in path efficiency and acceleration at all times. This was expected as ORCA explicitly exploits a *bidirectional* symmetry: (a) all agents run the same exact policy and (b) they know that they do so. This enables the extraction of guarantees for collision avoidance over limited time windows under velocity and clearance constraints. In contrast, HCPnav is agnostic to the exact policies that others are following; it does perform better if others are following the same policy (as illustrated in our findings) but it does not require that others do so. Furthermore, ORCA is a local approach whereas HCPnav, leveraging the complete trajectory prediction provided by HCP, is effectively a global

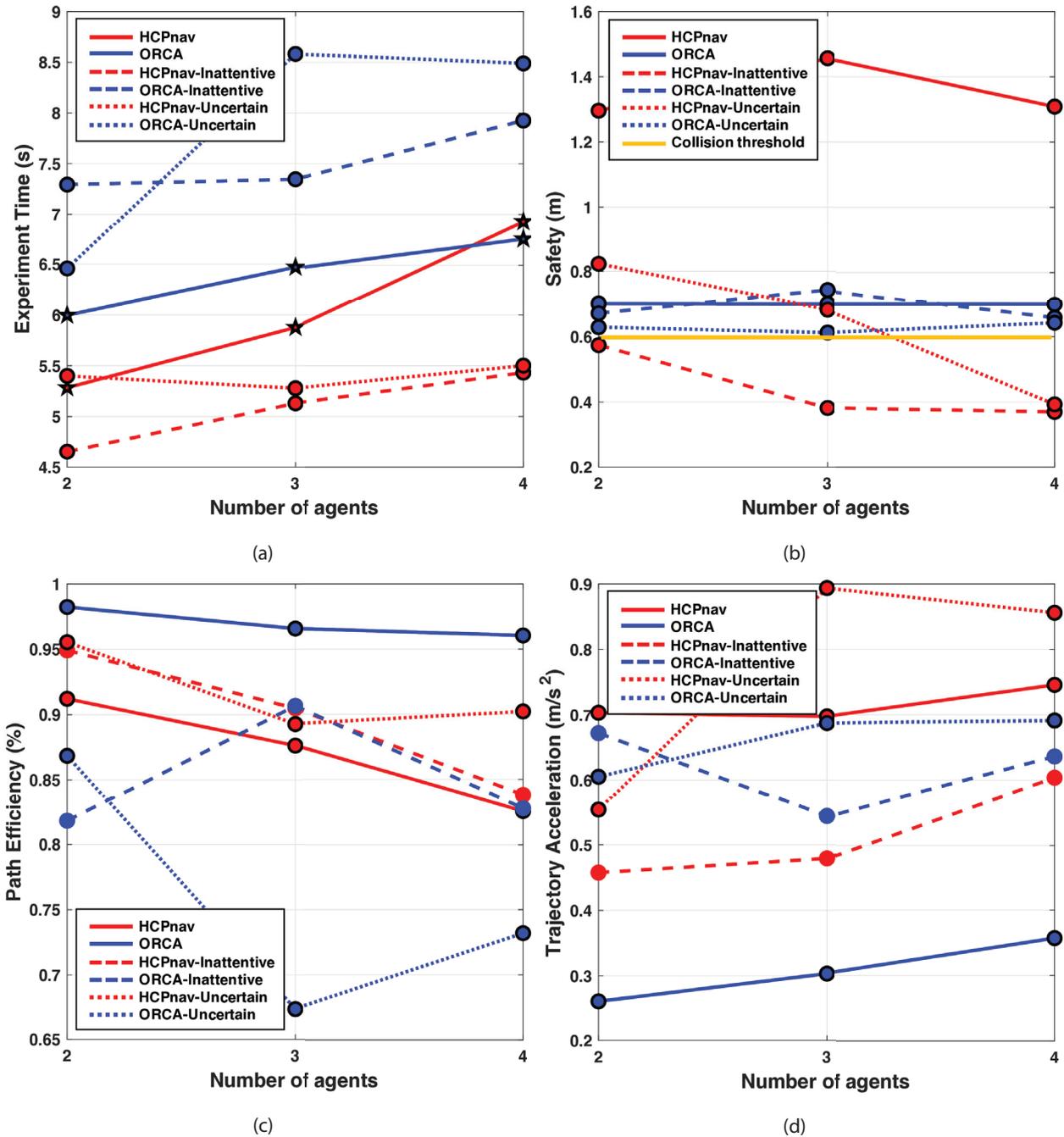


Fig. 9. Trajectory quality for all experiment configurations considered: (a) experiment time; (b) safety; (c) path efficiency; (d) trajectory acceleration. For group size, the same 200 randomly generated scenarios are executed under each of the conditions considered with both planners. For each condition and measure, we perform a paired Student's t -test to compare the populations yielded by HCPnav and ORCA. Points with black circular boundaries indicate rejection of the null hypothesis with p -value < 0.001 whereas points with star boundaries indicate rejection of the null hypothesis with p -value < 0.05 .

approach. This is in part why HCPnav is shown to have lower path efficiency and higher accelerations: HCPnav proactively adapts agents' behaviors towards an outcome of joint collision avoidance.

We also study the performance of HCPnav under heterogeneous settings. We see that under the inattentive condition, HCPnav is unsafe (< 0.6 m for all numbers of

agents) whereas ORCA is able to keep a consistently small collision-free clearance to other agents (< 0.8 m). However, we still see that ORCA's path and time efficiency drop significantly, as expected: after all, ORCA was also not meant to accommodate uncooperative agents. However, we see that under the uncertain condition, HCPnav is capable of ensuring a small threshold of safety (> 0.7 m) for

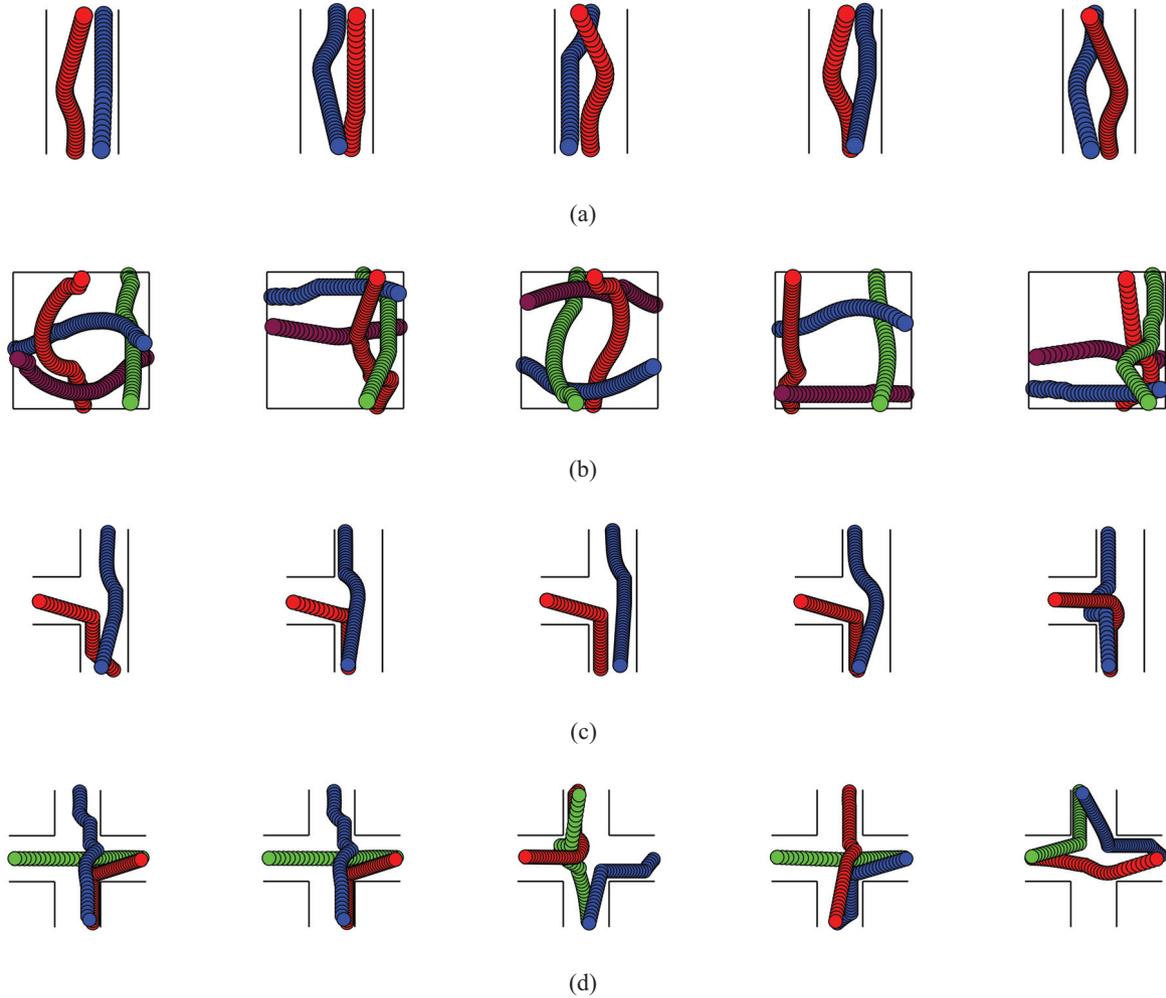


Fig. 10. Trajectories extracted by running HCPnav on workspaces of different boundary geometries: (a) corridor; (b) square; (c) T-junction; (d) crossroad.

experiments involving two and three agents whereas ORCA is at the boundary of safety (≈ 0.6 m). In these experiments, we see that HCPnav is also significantly more time efficient, but it takes longer paths and exhibits higher accelerations. ORCA handles the four-agent case but with limited time efficiency and marginal safety.

It should be noted that this evaluation is not meant as an attempt to outperform ORCA which was specifically designed to provide a computationally efficient framework for multiagent simulation. It is meant as a demonstration that HCPnav, a navigation framework leveraging our main contribution (HCP) is capable of achieving comparable performance to ORCA in a series of interesting scenarios under a variety of settings. Note also that the use of alternative cost functions within the HCPnav framework could promote or reject candidate coordination primitives in different ways which could also affect performance; the cost functions employed in the present article express important navigation specifications but their functional form or parametrization was not optimized. In general, HCPnav serves

as an example of the capabilities of HCP, which could be incorporated in a variety of alternative frameworks to provide valid predictions (e.g., MPC-based approaches).

6.2.2. Handling alternative workspace geometries. We present a series of examples, demonstrating the ability of our framework to generate motion that conforms to distinct workspace boundaries, through the use of the mechanism proposed in Section 5.4. Figure 10 demonstrates the trajectories generated by running HCPnav in four different environments: a rectangular corridor (Figure 10(a)), a square workspace (Figure 10(b)), a T-junction (Figure 10(c)), and a crossroad (Figure 10(d)). For each workspace, we uniformly sample agents' starting and ending configurations. The figures depict the top view of the scenes, with agents' trajectories represented as swept volumes (an agent crossing "over" another indicates a later passing).

Overall, we observe that the agents avoid each other while respecting the bounds of the workspace. The boundary limits the applicability of the cooperative strategies

generated by HCP but we still see that agents manage to coordinate thanks to the proactive character of the decision making induced by HCPnav.

7. Discussion

We presented a framework for generating trajectories of desired spatiotemporal entanglement for multiple holonomic agents navigating on the plane. In multiagent navigation domains, the spatiotemporal entanglement of agents' trajectories captures crucial features of the dynamics of interaction among them. In this article, we modeled this entanglement through the introduction of a mathematical formalism, based on the *winding number* topological invariant. This formalism has a dual symbolic and analytical nature, providing powerful abstraction while enabling geometric reasoning. Leveraging these properties, we introduced a trajectory generation framework, HCP, that generates multiagent trajectories of desired topological specifications. HCP achieves that by modeling agents as interacting dynamical systems with desired properties (vortices). As the vortices interact with each other, they produce multiagent trajectories that are entangled in the desired way. Through qualitative and quantitative analysis, we illustrated the efficacy of HCP in generating desired multiagent trajectory topologies in challenging scenarios. We also demonstrated the scaling properties and computational efficiency of HCP in a comparative study against a trajectory-optimization baseline. To illustrate the value of HCP for multiagent applications, we developed HCPnav, a cost-based motion planner that acts directly on HCPs, following ego-motion that corresponds to the primitive of minimum cost. HCPnav was shown to perform comparably to ORCA (van den Berg et al., 2009), a widely employed framework for multirobot navigation in a variety of scenarios. HCP could be a valuable component of alternative planning and control frameworks including, e.g., model predictive controllers (García et al., 1989).

The HCP framework introduces a series of novel features and advances that improve upon past work in the area. Past work has proposed techniques for generating topologically distinct trajectories for a single robot in a cluttered environment (Bhattacharya et al., 2012; Pokorny and Kragic, 2015; Rösmann et al., 2017). Instead, our framework focuses on global trajectory generation subject to topological constraints for multiple reactive agents, navigating between arbitrary configurations in \mathbb{R}^2 , a problem with important real-world instantiations (e.g., social robot navigation, multirobot planning, etc.). Some works in this area employ topological braids as the basis of abstracting multiagent navigation behaviors (e.g., Diaz-Mercado and Egerstedt, 2017; Mavrogiannis and Knepper, 2019). Braids require the selection of a projection plane in order to symbolically express multiagent motion primitives which are only symbolically described. In contrast, our work employs a representation based on the winding number, which

directly offers analytical descriptions of symbolically defined primitives. These descriptions enable the adoption of a variety of techniques for control design and trajectory optimization.

In this article, we built upon the framework of Berger (2001a) which constructs interacting dynamical systems for braided trajectory generation. The application of gradient-based trajectory-optimization techniques to motion planning problems in robotics often entails weighted sums of functionals, which act on trajectories locally, thus often losing sight of global, topological specifications in favor of alternative local cost improvements. In contrast, the laws of motion extracted from the construction of appropriately parametrized interacting dynamical systems incorporate a global understanding of the unfolding trajectory topology, yielding reactive motion that balances the satisfaction of topological constraints with goal-reaching maneuvers. Similar to gradient-based optimization techniques, our method cannot guarantee the attainment of global optima. However, our empirical findings illustrate that the dynamics-based machinery of the HCP outperforms the trajectory-optimization baseline across a class of problems.

Finally, our decentralized-navigation experiments showed that HCPnav is capable of performing comparably with ORCA across a number of experiments. HCPnav performs better under homogeneous settings, exhibiting significantly more time-efficient and safer motion than ORCA. This was expected because it generates motion corresponding to HCP primitives, explicitly following coordination protocols. We show that HCPnav is capable of handling scenarios within moderately crowded environments (2–4 agents) and that it can adapt to agents with changing intentions.

7.1. Limitations and directions for future work

Our work is limited in a few ways. We discuss the main limitations and offer directions for future work.

First, HCP comes with no guarantees on the satisfaction of topological constraints. Our empirical evidence showed that HCP achieves almost near-perfect performance for scenarios involving up to four agents, and satisfactory performance in scenarios involving five agents (see Table 1). Some of the failure cases correspond to topologies that were physically unattainable, i.e., topologies that would require unrealistic maneuvers to achieve as the topology representation w is agnostic to realizability. Future work

Table 2. Success rates and computation times for HCP and CHOMP over 500 randomly generated 2-agent scenarios

Planner	Success (%)	Iterations	Time (s)
CHOMP	78.80	80.3325	0.1291
HCP	98.40	86.8862	0.0048

will look into ways to quantify the realizability of topological specifications. The likelihood formalism employed under the HCPnav framework represents an indication of realizability but additional work is needed to come up with a formal model.

Furthermore, although HCP is shown to be computationally efficient compared with CHOMP (see Table 2), in practice, we observed a computational overhead that prevented us from running experiments with more than 5 agents on our non-optimized implementation (Matlab code running on a MacBook Pro with an Intel Core i7 processor at 2.5 GHz). The same observation prevented us from scaling HCPnav to experiments beyond four agents. Although these observations are in part an artifact of our limited computational resources and implementation, the exponential complexity of running computations across pairs of agents is a fundamental limitation of our approach. However, note that the goal of our framework was not to scale to arbitrary numbers of agents (such as ORCA); our goal was specifically to design a framework that can, under specific conditions, generate desired multiagent trajectory topologies. As discussed, to the best of the authors' knowledge, HCP is the first framework to generate multiagent navigation trajectories driving agents between arbitrary configurations on the plane under topological constraints on their spatiotemporal pattern. The proposed HCPnav framework constitutes a first example of how HCP could prove useful for multiagent navigation scenarios. Note that under real-world conditions, even humans tend to not consider explicitly all agents present in a navigation environment but rather tend to cluster entities together when planning paths, effectively reasoning about limited numbers of agents (Mavrogiannis et al., 2019; Wang and Steinfeld, 2020). Finally, for robots that tend to have a limited sensing radius and horizon, the scaling property does not appear to be of central concern: the number of agents that the robot would interact with at the same time would always be bounded by an area of limited coverage.

The presented frameworks (HCP and HCPnav) were introduced as novel paradigms for multiagent trajectory generation and decentralized multiagent collision avoidance. Their performance could be further improved by fine-tuning their parameters through the use of empirical or optimization-based techniques. For instance, the gain parameters k , k_{rep} or k_{att} of HCP could be fine-tuned to yield better performance. Furthermore, alternative criticality functions c_{ij} could be employed to yield more aggressive or defensive behavior. Similarly, the performance of HCPnav could be improved by considering alternative parametrizations or alternative functional form for its components. For example, the accuracy of the probability model of (16) could be improved using data-driven techniques (Mavrogiannis et al., 2017). Finally, the weights of the cost of (18) could be further tuned and the costs considered could be replaced with alternative ones. Similarly, the runtime and performance of the geometry-conforming module could be improved by considering a precomputed distance

map. Although all mentioned improvements could further improve performance, our main focus on this article was not to provide a fine-tuned implementation, but rather to contribute a general blueprint based upon which researchers and practitioners could iterate upon to get the desired performance based on their application.

Finally, we considered a deliberately simplified setup to specifically focus on the feasibility of enforcing topological constraints to multiagent trajectory generation. We considered holonomic systems, which simplify the incorporation of topological constraints to motion planning problems. However, many real-world mobile-robot systems feature non-holonomically constrained kinematics which motivates their consideration in future work. We also presented a simulated evaluation that focused on the quantification of trajectory properties under offline (HCP) and online (HCPnav) settings. Future work will address the challenges of real-world implementation, considering a non-holonomic miniature robotic racecar platform (Srinivasa et al., 2019) deployed in a lab environment.

Funding

This material is based upon work supported by the National Science Foundation (grant numbers IIS-1526035 and IIS-1563705).

ORCID iDs

Christoforos Mavrogiannis  <https://orcid.org/0000-0003-4476-1920>

Note

1. In this article, we model the *criticality* term as the inverse of a polynomial function of the distance between two agents, activated when the distance becomes lower than a threshold. Alternative options could also be employed.

References

- Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L and Savarese S (2016) Social LSTM: Human trajectory prediction in crowded spaces. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 961–971.
- Alonso-Mora J, Breitenmoser A, Beardsley P and Siegwart R (2012) Reciprocal collision avoidance for multiple car-like robots. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 360–366.
- Aref H (2007) Point vortex dynamics: A classical mathematics playground. *Journal of Mathematical Physics* 48(6): 065401.
- Aref H, Jones S, Mofina S and Zawadzki I (1989) Vortices, kinematics and chaos. *Physica D: Nonlinear Phenomena* 37(1): 423–440.
- Baker CL, Tenenbaum J and Saxe R (2009) Action understanding as inverse planning. *Cognition* 113(3): 329–349.
- Berger MA (2001a) Hamiltonian dynamics generated by Vassiliev invariants. *Journal of Physics A: Mathematical and General* 34(7): 1363.

- Berger MA (2001b) Topological invariants in braid theory. *Letters in Mathematical Physics* 55(3): 181–192.
- Bhattacharya S, Likhachev M and Kumar V (2012) Topological constraints in search-based robot path planning. *Autonomous Robots* 33(3): 273–290.
- Birman JS (1975) *Braids Links And Mapping Class Groups*. Princeton, NJ: Princeton University Press.
- Cao C, Trautman P and Iba S (2019) Dynamic channel: A planning framework for crowd navigation. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '19)*, pp. 5551–5557.
- Carton D, Olszowy W and Wollherr D (2016) Measuring the effectiveness of readability for mobile robot locomotion. *International Journal of Social Robotics* 8(5): 721–741.
- Chen YF, Everett M, Liu M and How JP (2017) Socially aware motion planning with deep reinforcement learning. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '17)*, pp. 1343–1350.
- Cooper GF (1990) The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence* 42(2): 393–405.
- Csibra G and Gergely G (2007) ‘Obsessed with goals’: Functions and mechanisms of teleological interpretation of actions in humans. *Acta Psychologica* 124(1): 60–78.
- Davis B, Karamouzas I and Guy S (2020) NH-TTC: A gradient-based framework for generalized anticipatory collision avoidance. In: *Proceedings of Robotics: Science and Systems*, Corvallis, OR.
- Denny J, Chen K and Zhou H (2018) A topology-based path similarity metric and its application to sampling-based motion planning. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6498–6505.
- Denny J and Fine BT (2020) Topology-based group routing in partially known environments. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20)*. New York, NY: Association for Computing Machinery, pp. 784–791.
- Denny J, Sandström R, Bregger A and Amato NM (2020) Dynamic region-biased rapidly-exploring random trees. In: *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*. Cham: Springer, pp. 640–655.
- Diaz-Mercado Y and Egerstedt M (2017) Multirobot mixing via braid groups. *IEEE Transactions on Robotics* 33(6): 1375–1385.
- Farina F, Fontanelli D, Garulli A, Giannitrapani A and Prattichizzo D (2017) Walking ahead: The headed social force model. *PLoS ONE* 12(1): 1–23.
- Fiorini P and Shiller Z (1998) Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research* 17(7): 760–772.
- Garca CE, Prett DM and Morari M (1989) Model predictive control: Theory and practice — a survey. *Automatica* 25(3): 335–348.
- Goffman E (1966) *Behavior in Public Places: Notes on the Social Organization of Gatherings*. Free Press.
- Goodfellow I, Pouget-Abadie J, Mirza M, et al. (2014) Generative adversarial nets. *Advances in Neural Information Processing Systems* 27: 2672–2680.
- Graves A, Mohamed A and Hinton G (2013) Speech recognition with deep recurrent neural networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649.
- Gunning R and Rossi H (1965) *Analytic Functions of Several Complex Variables*. AMS Chelsea Publishing/Prentice-Hall.
- Gupta A, Johnson J, Fei-Fei L, Savarese S and Alahi A (2018) Social GAN: Socially acceptable trajectories with generative adversarial networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264.
- Helbing D and Molnár P (1995) Social force model for pedestrian dynamics. *Physical Review E* 51: 4282–4286.
- Henry P, Vollmer C, Ferris B and Fox D (2010) Learning to navigate through crowded environments. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 981–986.
- Hochreiter S and Schmidhuber J (1997) Long short-term memory. *Neural Computation* 9(8): 1735–1780.
- Hoogendoorn S and Bovy PH (2003) Simulation of pedestrian flows by optimal control and differential games. *Optimal Control Applications and Methods* 24(3): 153–172.
- Hopcroft J, Schwartz J and Sharir M (1984) On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the “warehouseman’s problem”. *The International Journal of Robotics Research* 3(4): 76–88.
- Hu J, Prandini M and Sastry S (2003) Optimal coordinated motions of multiple agents moving on a plane. *SIAM Journal on Control and Optimization* 42(2): 637–668.
- Karamouzas I, Heil P, van Beek P and Overmars MH (2009) A predictive collision avoidance model for pedestrian simulation. In: *Motion in Games*. Berlin: Springer, pp. 41–52.
- Karamouzas I, Skinner B and Guy SJ (2014) Universal power law governing pedestrian interactions. *Physical Review Letters* 113: 238701.
- Kaup L, Kaup B, Barthel G and Bridgland M (1983) *Holomorphic Functions of Several Variables: An Introduction to the Fundamental Theory (De Gruyter Studies in Mathematics)*. Berlin: W. de Gruyter.
- Kim B and Pineau J (2016) Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics* 8(1): 51–66.
- Knepper RA and Mason MT (2012) Real-time informed path sampling for motion planning search. *The International Journal of Robotics Research* 31(11): 1231–1250.
- Knepper RA and Rus D (2012) Pedestrian-inspired sampling-based multi-robot collision avoidance. In: *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN '12)*, pp. 94–100.
- Knepper RA, Srinivasa SS and Mason MT (2012) Toward a deeper understanding of motion alternatives via an equivalence relation on local paths. *The International Journal of Robotics Research* 31(2): 167–186.
- Kretzschmar H, Spies M, Sprunk C and Burgard W (2016) Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research* 35(11): 1289–1307.
- Kruse T, Basili P, Glasauer S and Kirsch A (2012) Legible robot navigation in the proximity of moving humans. In: *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp. 83–88.
- Lo S, Alkoby S and Stone P (2019) Robust motion planning and safety benchmarking in human workspaces. In: *Workshop on Artificial Intelligence Safety, AAAI Conference on Artificial Intelligence (AAAI '19)*.

- Luber M, Spinello L, Silva J and Arras KO (2012) Socially-aware robot navigation: A learning approach. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 902–907.
- Mainprice J and Berenson D (2013) Human–robot collaborative manipulation planning using early prediction of human motion. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 299–306.
- Mavrogiannis C (2019) *Motion Planning for Socially Competent Robot Navigation*. PhD Thesis, Cornell University.
- Mavrogiannis C, Baldini F, Wang A, et al. (2021) Core challenges of social robot navigation: A survey. *arXiv preprint arXiv:2103.05668*.
- Mavrogiannis C, Hutchinson AM, Macdonald J, Alves-Oliveira P and Knepper RA (2019) Effects of distinct robotic navigation strategies on human behavior in a crowded environment. In: *Proceedings of the ACM/IEEE International Conference on Human–Robot Interaction (HRI)*.
- Mavrogiannis CI, Blukis V and Knepper RA (2017) Socially competent navigation planning by deep learning of multi-agent path topologies. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '17)*, pp. 6817–6824.
- Mavrogiannis CI and Knepper RA (2019) Multi-agent path topology in support of socially competent navigation planning. *The International Journal of Robotics Research* 38(2–3): 338–356.
- Mavrogiannis CI and Knepper RA (2020a) Decentralized multi-agent navigation planning with braids. In: *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*. Cham: Springer, pp. 880–895.
- Mavrogiannis CI and Knepper RA (2020b) Multi-agent trajectory prediction and generation with topological invariants enforced by Hamiltonian dynamics. In: *Algorithmic Foundations of Robotics XIII*. Cham: Springer, pp. 744–761.
- Mavrogiannis CI, Thomason WB and Knepper RA (2018) Social momentum: A framework for legible navigation in dynamic multi-agent environments. In: *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI '18)*, pp. 361–369.
- Moussaïd M, Helbing D and Theraulaz G (2011) How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences* 108(17): 6884–6888.
- Nitsche M (2006) Vortex dynamics. In: Francoise J-P, Naber GL and Tsun TS (ed.) *Encyclopedia of Mathematics and Physics*. Amsterdam: Elsevier Science Ltd.
- Orthey A, Akbar S and Toussaint M (2020) Multilevel motion planning: A fiber bundle formulation. *arXiv preprint arXiv:2007.09435*.
- Orthey A and Toussaint M (2020) Visualizing local minima in multi-robot motion planning using multilevel morse theory. In: *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.
- Park JJ, Johnson C and Kuipers B (2012) Robot navigation with model predictive equilibrium point control. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 4945–4952.
- Pokorny FT and Kragic D (2015) Data-driven topological motion planning with persistent cohomology. In: *Proceedings of Robotics: Science and Systems (RSS '15)*.
- Pokorny FT, Kragic D, Kavraki LE and Goldberg K (2016) High-dimensional winding-augmented motion planning with 2D topological task projections and persistent homology. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '16)*, pp. 24–31.
- Rösmann C, Hoffmann F and Bertram T (2017) Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems* 88: 142–153.
- Schmerling E, Leung K, Vollprecht W and Pavone M (2018) Multimodal probabilistic model-based planning for human-robot interaction. In: *Proceedings IEEE Conference on Robotics and Automation*, pp. 3399–3406.
- Shiomi M, Zanlungo F, Hayashi K and Kanda T (2014) Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model. *International Journal of Social Robotics* 6(3): 443–455.
- Sisbot EA, Marin-Urias LF, Alami R and Siméon T (2007) A human aware mobile robot motion planner. *IEEE Transactions on Robotics* 23(5): 874–883.
- Snape J, van den Berg J, Guy SJ and Manocha D (2011) The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics* 27(4): 696–706.
- Spirakis P and Yap CK (1984) Strong NP-hardness of moving many discs. *Information Processing Letters* 19(1): 55–59.
- Srinivasa SS, Lancaster P, Michalove J, et al. (2019) MuSHR: A low-cost, open-source robotic racecar for education and research. *CoRR* abs/1908.08031.
- Sutton RS and Barto AG (2018) *Reinforcement Learning: An Introduction*. Cambridge, MA: A Bradford Book.
- Tang C and Salakhutdinov RR (2019) Multiple futures prediction. *Advances in Neural Information Processing Systems* 32: 15398–15408.
- Treuille A, Cooper S and Popovic Z (2006) Continuum crowds. *ACM Transactions on Graphics* 25(3): 1160–1168.
- Truong XT and Ngo DN (2017) Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model. *IEEE Transactions on Automation Science and Engineering* 14(4): 1743–1760.
- Turnwald A and Wollherr D (2019) Human-like motion planning based on game theoretic decision making. *International Journal of Social Robotics* 11(1): 151–170.
- van den Berg J, Guy SJ, Lin MC and Manocha D (2009) Reciprocal n -body collision avoidance. In: *Proceedings of the International Symposium on Robotics Research (ISRR '09)*, pp. 3–19.
- Vemula A, Muelling K and Oh J (2018) Social attention: Modeling attention in human crowds. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '18)*, pp. 1–7.
- Wang A and Steinfeld A (2020) Group split and merge prediction with 3D convolutional networks. *IEEE Robotics and Automation Letters* 5(2): 1923–1930.
- Warren WH (2006) The dynamics of perception and action. *Psychological Review* 113(2): 358–389.
- Wiese E, Wykowska A, Zwicker J and Müller HJ (2012) I see what you mean: How attentional selection is shaped by ascribing intentions to others. *PLoS ONE* 7(9): 1–7.
- Wolfinger NH (1995) Passing moments: Some social dynamics of pedestrian interaction. *Journal of Contemporary Ethnography* 24(3): 323–340.
- Zucker M, Ratliff N, Dragan A, et al. (2013) CHOMP: Covariant Hamiltonian optimization for motion planning. *The International Journal of Robotics Research* 32(9–10): 1164–1193.