A Novel Relationship Between Dynamics and Complexity in Multi-agent Collision Avoidance

Jeffrey Kane Johnson
School of Informatics and Computing
Indiana University
Bloomington, Indiana 47408
Email: jj56@indiana.edu

***Abstract*—This paper examines the relationship between system dynamics and problem complexity of collision avoidance in multi-agent systems. Motivated particularly by results in the field of automated driving, a variant of the reciprocal n-body collision avoidance problem is considered. In this problem, agents must avoid collision while moving according to individual reward functions in a crowded environment. The main contribution of this work is the novel result that there is a quantifiable relationship between system dynamics and the requirement for agent coordination, and that this requirement can change the complexity class of the problem dramatically: from P to NEXP or even NEXP$^{\text{NP}}$. In addition, a constructive proof is provided that demonstrates the relationship and potential real-world applications of the result are discussed.**

I. Introduction

In industries as varied as mining, agriculture, health care, and autonomous driving, many practical applications in robotics involve navigating through dynamic environments in the presence of intelligent agents. A large and relatively mature body of literature has been developed that examines various types of these multi-agent systems and the theoretical complexity of planning within them. The focus in this work is specifically how system dynamics interact with problem complexity. For single agent systems, an early result due to Reif and Sharir [27] showed that adding velocity bounds to one type of motion planning problem can change its complexity from NP-hard to PSPACE-hard. This result clearly indicates that system dynamics can play a role in determining complexity class, however, relatively little attention has been paid to the role that system dynamics play in the complexity of multi-agent problems.

As will be shown, one of the key factors affecting complexity of multi-agent problems is agent coordination. In real world applications for multi-agent systems, path planning has historically been treated as the primary problem ahead of agent coordination. But, in many cases, this may not reflect where the true difficulty lies. For example, in the case of an automated vehicle moving along a road network, the environment explicitly bounds the set of paths that make progress toward a goal, which significantly reduces the difficulty of path planning without necessarily a commensurate reduction in difficulty of the coordination problem. The team behind the planner used in the Bertha Benz drive (Ziegler et al. [35]), as well as the winning teams of the 2007 DARPA Urban Challenge (Urmson et al. [29], Montemerlo et al. [23], and Bacha et al. [4]) all cite coordination ahead of path planning as an area of future work. Given the importance of the coordination problem and that practically any real-world system is dynamically constrained, understanding the relationship between system dynamics and coordination requirements is especially important and serves as one of the primary motivations for this work.

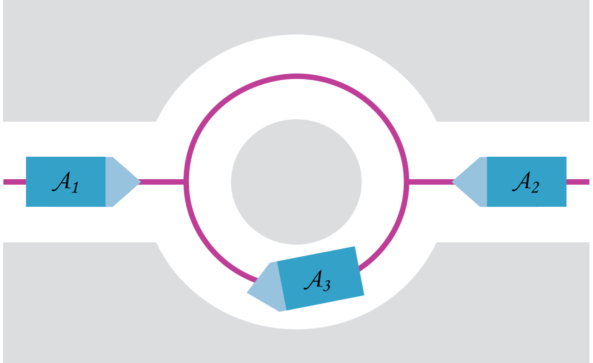


Fig. 1: Agents $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ attempt to navigate past each other along fixed paths. This paper examines how system dynamics affect the need for them to coordinate their actions.

Recent years have seen a significant push toward automated driving (AD), but one of main difficulties of AD is the problem of handling inter-vehicle interactions safely and efficiently. At first glance, explicit coordination among vehicles (V2V) would seem to provide a solution, and much work has been done to develop the technology and standards. However, it is not foreseeable that the availability of the required communication channels can be guaranteed to levels required of safety-critical applications [15]. But it's also unclear to what degree that kind of communication is actually necessary: human drivers navigate successfully with only very limited[1] forms of communication, which implies that the coordination they do is also very limited. This raises the question of whether, and to what extent, coordination is actually required for navigating multi-agent systems, and it implies that a better understanding of that

[1] Indicator lights are a common channel of communication, but they are notoriously unreliable. Horns also provide a form of communication, but are limited by context. Relative positions and speeds can convey intent, but, as channels of communication, these are very low bandwidth.

requirement will lead to the development of more practical and robust navigation algorithms.

This paper examines agent coordination in a variant of the reciprocal n-body collision avoidance problem described by van den Berg et al. [31]. The key insight is that system dynamics can introduce a requirement for coordination where there otherwise would be none, and a constructive proof is given that allows the existence of this requirement to be tested. The importance of the coordination requirement is that once it exists within a system, the space of appropriate models for the problem changes, which changes the complexity class of any solutions to the problem. This result demonstrates the existence of fundamental ties between system dynamics and problem complexity for multi-agent collision avoidance problems.

First a selection of relevant background literature is covered and then the main results of the paper are derived. Finally, future work and conclusions will be discussed.

II. Related Work

This section will detail a selection of the large body of relevant work on collision avoidance, planning, and complexity. First several general theoretical complexity results are described, then a brief description of notable solution techniques for multi-agent systems, and, finally, a brief survey of approaches taken in practical solutions is given.

In the absence of dynamic constraints and other moving agents, the problem of planning a collision-free path through an environment is typically referred to as the "mover's problem," which is the problem of moving an articulated polyhedral body through a Euclidean space populated with static polyhedral obstacles. Reif [26] showed the general problem to be complete for PSPACE and the classical problem, referred to as the "piano mover's problem," where the moving body is a rigid polyhedron moving in $\mathbb{R}^2$ or $\mathbb{R}^3$, to be in P under the condition that geometric constraints can be expressed algebraically. Work by Halperin and Sharir [14] further showed near quadratic bounds for the $\mathbb{R}^2$ case. The multi-body variant of the piano mover's problem, known as the "warehouseman's problem," was shown by Hopcroft et al. [17] to be PSPACE-hard. Reif and Sharir [27] additionally showed that introducing agents that follow fixed-trajectories into the piano mover's problem for $\mathbb{R}^3$ changes the complexity class of the problem to NP-hard, and that adding velocity bounds makes the problem PSPACE-hard.

For multiple agents following non-fixed trajectories, the problem is generally formulated in terms of sequential decision making in a discretized space rather than geometric motion in a continuous space. When planning for these agents can be done independently while still achieving a jointly optimal solution, the problem can be formulated as a type of Markov decision process (MDP), which Papadimitriou and Tsitsiklis [24] showed belongs to complexity class P. However, as noted by Boutilier [8], independent planning cannot be guaranteed in general to result in a globally optimal plan. For global optimality to be guaranteed, joint planning must occur. While typical centralized[2] joint planning problems can also be formulated as types of MDP's, and therefore belong to P, others, such as the unlabeled[3] variant, are PSPACE-hard. For decentralized problems Bernstein et al. [7] showed that for cooperative[4] agents this class of problems is at least complete for NEXP in both the jointly fully-observable (DEC-MDP) and jointly partially-observable (DEC-POMDP) cases. Goldsmith and Mundhenk [13] showed the non-cooperative version of this problem, the partially-observable stochastic game (POSG), to be complete for $\mathrm{NEXP}^{\mathrm{NP}}$.

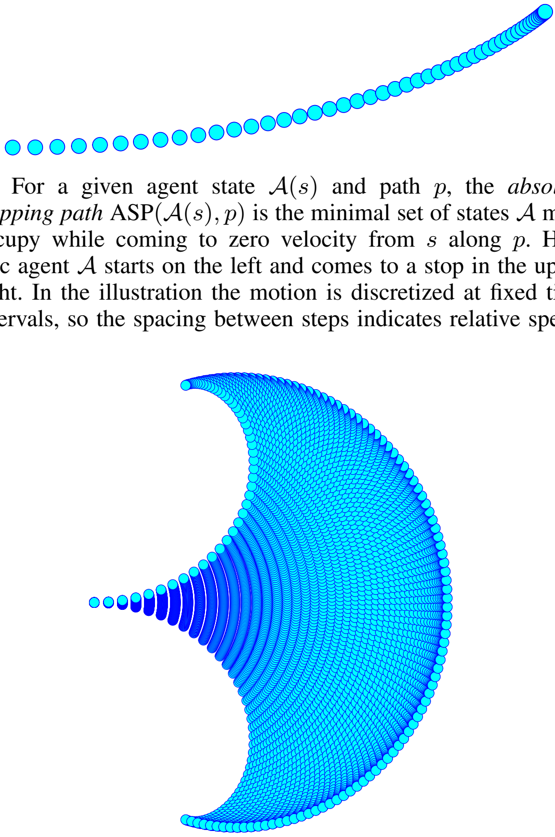
In the context of collision avoidance in multi-agent systems, Fiorini and Shiller [10] introduced the notion of "velocity obstacles" to address the pairwise collision avoidance problem. In this approach the set of velocities resulting in collision between a robot and another moving agent are computed explicitly, and this set is called the velocity obstacle (VO). Collision avoidance is then guaranteed by assigning velocities outside the VO to the agent. Fraichard and Asama [12] described the more general idea of an "inevitable collision state" (ICS) as a "state for which, no matter what the future trajectory followed by the system is, a collision with an obstacle eventually occurs." Similar state descriptors had been proposed by LaValle and Kuffner [21]. Owing to the inherent computational complexity of the ICS representation, Bekris [5] examined sampling-based approximation methods. The Optimal Reciprocal Collision Avoidance (ORCA) framework introduced by van den Berg et al. [31] expanded the ideas of VO and ICS to first-order multi-agent systems. Later, van den Berg et al. [32] extended their results to consider second-order constraints. Pairwise collision avoidance for holonomically constrained systems was demonstrated by Wilkie et al. [34] and extended to general multi-agent systems by Alonso-Mora et al. [2]. When coordination among agents is allowed, Bekris et al. [6] demonstrated that non-collision can be guaranteed for a broad class of de-centralized motion planning problems. Shoham and Tennenholtz [28] describe an alternate approach to these types of problems that imposes artificial rules, or "social laws," on agent coordination in order to remove the need for online coordination altogether.

The distinction between sequential decision making problems and continuous geometric motion planning problems is typically formulated mathematically as the problem of choosing among a finite number of homotopy channels in some state space (decision making), and generating actuation commands to navigate those channels (motion planning). In practice, most interesting problems have characteristics of both problem types, and therefore are hybrid problems with hybrid solutions. An early approach from Kambhampati et al. [20] solved these hybrid problems by interleaving graph planning

[2] "Centralized" planning problems are those in which a central planner coordinates the motions of all agents. In "decentralized" problems, each agent invokes its own planner.

[3] "Unlabeled" refers here to the idea that multiple agents must reach multiple goal positions without concern for which agent reaches which goal.

[4] "Cooperative" here means that agents have a *shared* reward function. In contrast, "non-cooperative" means that each agent only optimizes for positive outcomes for some subset of all agents.



(a) For a given agent state $\mathcal{A}(s)$ and path p, the *absolute stopping path* $\text{ASP}(\mathcal{A}(s), p)$ is the minimal set of states $\mathcal{A}$ must occupy while coming to zero velocity from s along p. Here disc agent $\mathcal{A}$ starts on the left and comes to a stop in the upper right. In the illustration the motion is discretized at fixed time intervals, so the spacing between steps indicates relative speed.

(b) For a given agent state $\mathcal{A}(s)$ and complete set of followable paths P, the *absolute stopping region* $\text{ASR}(\mathcal{A}(s), P)$ is the union of all ASPs over P. This illustration shows the ASR for disc agent $\mathcal{A}$ from (a). The region is plotted by sampling agent trajectories generated by sweeping steering commands from hard right to hard left.

Fig. 2: Illustrations of the shapes of an ASP (Definition 6) and an ASR (Definition 7) for a hypothetical disc agent following constant control trajectories with unicycle dynamics traveling along a 2D plane. System was initialized with non-zero velocity, bounded deceleration, and bounded yaw rate.

with motion planning in pre-defined discrete spaces. Later, Kaelbling and Lozano-Pérez [19] dealt with uncertainty and introduced sophisticated task description languages. When the topography of the state space is not known beforehand, Alterovitz et al. [3] introduced a sampling-based approach that can be used to construct roadmaps in the state space.

III. Problem Description

As stated earlier, this paper examines agent coordination under a variant of the reciprocal n-body collision avoidance problem described by van den Berg et al. [31]. In the formulation used here, the problem is generalized slightly to allow general system dynamics, and to make each agent's task to choose an appropriate control command rather than velocity:

Problem 1. Let $\mathbb{A}$ be a set of agents navigating a shared space with a shared reference frame and assume that collision is never inevitable in the initial system state. Assume each agent can fully observe the instantaneous dynamic state of the environment. Assume global constraints on dynamics, and that each agent moves according to a unique decision process. Each agent may assume with certainty that other agents will prefer both to avoid collision and to avoid causing collision, but that otherwise the future actions of other agents are not generally observable. Agents may coordinate, or negotiate, their future actions via communication under the following restrictions:

1) Communications are strictly pairwise
2) Agents may only communicate with regard to their own actions (that is, they may not relay information)
3) There is always some non-zero cost associated with communication

The question is: when $|\mathbb{A}| > 2$, how can a given agent choose a control with the guarantee that it will be possible for *all* agents to remain collision free for some time horizon?

The focus of this paper is on how system dynamics affect the model space and complexity of Problem 1.

A. Notations and Definitions

Assume all agents operate in a shared workspace $\mathcal{W}$, and let $\mathcal{S}$ denote the time-augmented state space [11] for all agents. Let Φ denote the set of all control trajectories[5] where for each $\phi \in \Phi$ the state of an agent $\mathcal{A}$ at time t from start state $s_0 \in \mathcal{S}$ under control trajectory ϕ is $s_t = \phi(s_0, t)$. Let $\mathcal{A}(s)$ denote the region of state space occupied by agent $\mathcal{A}$ at state s.

Definition 1. The actions of two agents are said to require *coordination* when the feasibility of either agent's actions cannot be guaranteed independently of the actions of the other.

Definition 2. For a given agent $\mathcal{A}$ an *obstacle* ($\mathcal{O}$) is a member of the set of obstacles $\mathbb{O}$, which is defined as[6]:

$$\mathbb{O} = \mathbb{A} \setminus \mathcal{A}$$

Definition 3. A *state space obstacle* ($\mathcal{B}_i$) is the set of states swept out by an obstacle $\mathcal{O}_i$ as it is controlled from an initial state s_i to an infinite time horizon T under ϕ_i:

$$\mathcal{B}_i = \bigcup_T \mathcal{O}_i(\phi_i(s_i, t))$$

Definition 4. An *inevitable collision state* (ICS)[7] for an agent $\mathcal{A}$ is a state from which all feasible future trajectories of $\mathcal{A}$ result in collision:

$$s \text{ is ICS} \leftrightarrow \forall \phi, \exists \mathcal{B}_i, \exists t :: \mathcal{A}(\phi(s,t)) \cap \mathcal{B}_i \neq \emptyset$$

It is important to note that by Definition 3, the computation of ICS space requires knowledge of future control trajectories of all obstacles.

[5] The set of controls is assumed to be shared, but this is not necessary.

[6] Non-interacting obstacles (e.g. walls or rolling rocks) follow fixed policies, so coordination with them is never required (or possible). For simplicity, they are excluded from treatment in this paper.

[7] Notations and definitions relating to ICSs are adapted from [12].

Definition 5. A *contingency plan* is a control sequence that an agent can execute that is guaranteed to avoid ICS space.

Definitions 6 & 7 below introduce concepts that will aid in the analysis of Problem 1. While sufficient to derive the results of this paper, these definitions will be refined in §IV-E to slightly generalize how dynamic systems are described.

Definition 6. For a given agent state $\mathcal{A}(s)$ and path p, the *absolute stopping path* $\text{ASP}(\mathcal{A}(s), p)$ is the minimal set of states $\mathcal{A}$ must occupy while coming to zero velocity from s along p (Figure 2a).

Definition 7. For a given agent state $\mathcal{A}(s)$ and complete set of followable paths P, the *absolute stopping region* $\text{ASR}(\mathcal{A}(s), P)$ is the union of all ASPs over P (Figure 2b):

$$\text{ASR}(\mathcal{A}(s), P) = \bigcup_{P} \text{ASP}(\mathcal{A}(s), p)$$

IV. Theory

This section will derive the main results of the paper. First the necessary notations and definitions are defined, and then the specific conditions are derived for which solutions to Problem 1 can make non-collision guarantees with and without agent coordination (Definition 1). An explicit problem formulation for Problem 1 is given and it is shown that the complexity of the problem is directly influenced by system dynamics via the coordination requirement. The section is closed with a discussion of results.

A. Relating Dynamics to Coordination

This section derives the relationship between system dynamics and the requirement for agent coordination under Problem 1.

Lemma 1. *In order to guarantee that any system can remain collision free, at least some reachable subset of ICS space must be computable from any state.*

Proof: This follows from Definition 4. In order to guarantee that a system can remain collision free, it must only move into states that are *not* in ICS space. To do that, some subset of the *complement* of ICS space must be computable. This equivalently means that some subset of ICS space must be computable. ■

Lemma 2. *Under Problem 1, ICS space is not computable without coordination among agents.*

Proof: This follows from Definitions 3 & 4: in order to compute ICS space the future control trajectories of *all* agents must be fully observable. However, under Problem 1, the future control trajectories are not fully observable without coordination among the agents. ■

Theorem 1. *Under Problem 1, coordination is required in general for each agent to maintain the ability to remain collision free.*

Proof: It follows directly from Definition 5 that all agents must have a contingency plan in order to guarantee the system can remain collision free. By Lemma 1 and Definition 5, computing a contingency plan requires computing some subset of ICS space. By Lemma 2, computing ICS space requires coordination among agents. ■

Using the ASR concept from Definition 7, the requirement for coordination will now be framed in terms of system dynamics.

Lemma 3. *Consider a system with two agent states $\mathcal{A}_1(s_1)$ and $\mathcal{A}_2(s_2)$. Let $ASR(\mathcal{A}_1(s_1), P_1) \cap ASR(\mathcal{A}_2(s_2), P_2) = \mathcal{Q}$. Assume there exist $ASP(\mathcal{A}_1(s_1), p_1)$ and $ASP(\mathcal{A}_2(s_2), p_2)$ such that neither intersect $\mathcal{Q}$. In other words, both $\mathcal{A}_1$ and $\mathcal{A}_2$ have disjoint stopping paths available. Then neither s_1 nor s_2 is an ICS.*

Proof: Since the stopping paths are disjoint, by Definition 6 both agent $\mathcal{A}_1$ and agent $\mathcal{A}_2$ can come to a stop without intersecting. Therefore, collision is not inevitable. ■

Lemma 4. *For a given agent state $\mathcal{A}_i(s_i)$, let $ASR(\mathbb{O}, \mathbb{P}) = \cup_i ASR(\mathcal{O}_i(s_i), P_i)$ be the union of all obstacle ASRs. A contingency plan for agent $\mathcal{A}_i$ is guaranteed to exist if there exists $ASP(\mathcal{A}_i(s_i), p)$ such that $ASP(\mathcal{A}_i(s_i), p) \cap ASR(\mathbb{O}, \mathbb{P}) = \emptyset$.*

Proof: If $\text{ASP}(\mathcal{A}_i(s_i), p) \cap \text{ASR}(\mathbb{O}, \mathbb{P}) = \emptyset$, then Lemma 3 says that s_i is not in an ICS. Therefore there must exist a control sequence ϕ under which $\mathcal{A}$ remains collision free. By Definition 5, ϕ is a contingency plan. ■

Theorem 2 establishes a condition under which any general dynamic system is guaranteed to be able to remain collision free without coordination:

Theorem 2. *A multi-agent system is guaranteed to be able to remain collision free without coordination if and only if for all $\mathcal{A}$ there exists an $ASP(\mathcal{A}_i(s_i), p)$ such that $ASP(\mathcal{A}_i(s_i), p) \cap ASR(\mathbb{O}, \mathbb{P}) = \emptyset$.*

Proof: It is equivalent to show that contingency plans can be guaranteed without coordination if and only if the above condition is true. By Lemma 4, *if* all agents have disjoint ASPs, then there must exist for all agents a contingency plan. To show that a contingency plan is guaranteed without coordination for all agents *only if* they all have disjoint ASPs, assume the contrary. Then no agent can come to a stop without potentially crossing the stopping path of one or more other agents. Without disjoint stopping paths, agents must know the future actions of other agents in order to have a valid contingency plan. But by Theorem 1 this requires coordination, which violates the premise. Thus, all agents are guaranteed to have a contingency plan without coordination *if* there is ASP disjointness, and *if* ASPs are *not* disjoint, they are *not* guaranteed a contingency plan without coordination. ■

The intuition behind Theorem 2 is closely tied to the notion of ICS space, and it would be equivalent to state that under certain conditions, it is impossible to compute *any subset* of ICS space without knowing the future actions of other agents. This idea is related to the sufficient safety condition for partial motion plans derived by Petti and Fraichard [25], which states that if the final state of a collision-free trajectory is not an ICS, then no state along the trajectory is an ICS state.

The following definition is made for convenience:

Definition 8. The condition that satisfies Theorem 2, that all agents have at least one disjoint stopping path, will be called *ASP disjointness*.

Theorem 2 states that coordination is unnecessary under ASP disjointness. But how, in practice, could agents maintain that property without coordination? Trivially, if an agent modulates its dynamics such that it can always come to a stop without *possibly* intersecting the path of any other agent, the property is satisfied. Agents need no knowledge of the plans of other agents for this; they simply need knowledge of the dynamics of the system. This is the approach taken, for example, by Mazer et al. [22] in the Ariadne's Clew algorithm. For most second-order systems, however, this behavior would likely be too conservative to be very useful. Worse, it's possible to specify initial conditions in a second-order system such that it is not possible to satisfy the property required by Theorem 2 (see §IV-D). This is why, for example, the algorithm for multi-agent collision avoidance for second-order systems given by Bekris et al. [6] requires coordination in order to maintain its guarantees.

But if Bekris et al. [6] require coordination, why is it that van den Berg et al. [31] do not? The ORCA framework they present is an efficient collision avoidance algorithm based on the VO representation that guarantees non-collision for very complex scenes without the need for agent coordination. As it turns out, Theorem 2 allows the requirement for agent coordination for certain systems to always be dropped. Theorem 3 will establish this possibility:

Lemma 5. *For first-order systems, $ASP(\mathcal{A}(s), p) = \mathcal{A}(s)$, and by extension, $\bigcup_P ASP(\mathcal{A}(s), p) = \mathcal{A}(s) = ASR(\mathcal{A}(s), P)$.*

Proof: That $\mathcal{A}$ can instantaneously stop means that the minimal set of states $\mathcal{A}$ must occupy while coming to a stop along *any* path p is exactly $\mathcal{A}(s)$. This implies further that $\bigcup_{\mathcal{P}} \text{ASP}(\mathcal{A}, P) = \mathcal{A}(s)$ which is equal to $\text{ASR}(\mathcal{A})$ by Definition 7. ∎

Theorem 3. *A first-order multi-agent system that is not currently in collision is guaranteed to be able to remain collision free without coordination.*

Proof: Lemma 5 implies that for systems with only first-order constraints, ASP disjointness holds for all non-collision states. By Theorem 2, such a system is guaranteed to be able to remain collision free without coordination. ∎

Note that any system described by the VO formulation is necessarily a first-order system and therefore Theorem 3 applies to it. This means the fact that Bekris et al. [6] required coordination for their solution and van den Berg et al. [31] did not is directly a result of the system dynamics they employed: the former dealt with a second-order system, and the latter a first-order system.

At a deeper level, these results speak to the fundamental problems with representing higher-order systems with lower-order approximations. The VO representation, for instance, is a first-order approximation often used for mutual collision avoidance in multi-agent systems because of its simplicity and elegance. However, Theorem 3 implies that the guarantees it makes are invalid for higher-order systems. This is demonstrated empirically by Wilkerson et al. [33] who showed that using the VO representation in a second-order system can result in collisions, even though the algorithms guarantee non-collision.

B. Collision Avoidance as a Decision Problem

This section gives a formulation for Problem 1 as a sequential decision making problem. The problem formulation is constructed such that it might reasonably map to real-world problems, but, to be clear, its purpose in this paper is to aid in the derivation of the main result, not to provide a practical solution to any specific problem instance.

As formulated, Problem 1 is hybrid decision making/motion planning problem, so its solution will also take a form similar to the hybrid solutions mentioned in §II. Let $\mathcal{R}$ be some sufficiently dense roadmap approximation to $\mathcal{S}$ (for instance, a Stochastic Motion Roadmap [3]), where "sufficiently dense" means dense enough to allow solutions to be found. Assume each agent is initialized at some vertex of $\mathcal{R}$, and assume all agents plan at a uniform and aligned frequency. Assume all agents have full knowledge of system dynamics and of $\mathcal{R}$, and that some efficient method for computing ASRs exists[8]. Assume that agents are capable of coordinating their actions (subject to the restrictions outlined in Problem 1) in a way similar to that presented in Bekris et al. [6], in which agents negotiate joint-contingency plans.

Define the problem G in its most general as a POSG:

Problem 2. Let $G = (\mathbb{A}, O, \mathcal{C}, c_0, A, T, \Omega, R)$, where:

- $\mathbb{A}$ is a set of agents whose states include contingency plans
- O is a set of finite observations (mapping of observable agent states to vertices of $\mathcal{R}$)
- $\mathcal{C}$ is a set of configurations of the system (mapping of full agent states to vertices of $\mathcal{R}$)
- c_0 is a designated initial configuration
- A is a finite set of actions that enable transition between any two vertices on $\mathcal{R}$
- $T : \mathcal{C} \times A^k \times \mathcal{C} \to [0, 1]$ is the transition probability function, where $T(c, a_1, \ldots, a_k, c')$ is the probability that configuration c' is reached from configuration c when each agent i chooses an action a_i
- $\Omega : \mathcal{C} \times I \to O$ is the observation function, where $\Omega(c, i)$ is the observation made in configuration c by agent i. The observation of one other agent may include the result of a negotiation (a joint-contingency plan); for all others the observation includes a distribution over contingency plans

[8] ASR computation can be treated as reachable set computations performed with either fast online approximations as in Johnson and Hauser [18] or Allen et al. [1], or via offline computation as in Valtazanos and Ramamoorthy [30]

- $R : \mathcal{C} \times A^k \times I \rightarrow \mathfrak{R}$ is an reward function, where $R(c, a_1, \ldots, a_k, i)$ is the reward gained by agent i in configuration c when the agents take actions $a_1, \ldots, a_k$

C. Main Result

This section derives main result of the paper, namely, that Problem 2 can be reduced to an MDP provided the non-coordination guarantee of Theorem 2 holds, and that it remains a POSG when it does not.

First, the reduction of G to an MDP under ASP disjointness:

Lemma 6. *An agent $\mathcal{A}$ can assume arbitrary policies for all $\mathcal{O} \in \mathbb{O}$ and maintain the non-collision guarantee provided the assumed policies at least maintain ASP disjointess.*

Proof: This follows directly from Theorem 2. From the standpoint of guaranteeing the ability to remain collision free, the action an agent takes is irrelevant so long as ASP disjointness is maintained. ■

Note that the principle of Lemma 6 is closely related to that exploited by the ORCA framework.

Theorem 4. *Under ASP disjointness, G can be modeled as an MDP.*

Proof: Lemma 6 states that the ability to remain collision free can be assured whenever an agent $\mathcal{A}$ assigns arbitrary polices to other agents so long as ASP disjointness is maintained. This is key, because it means $\mathcal{A}$ does not need to *observe* anything about the other agents beyond what is already fully observable in order to plan and execute motions in the presence of those other agents. $\mathcal{A}$ is free to assume full observability in future actions and a shared reward function. The quality of the assumptions about future actions is irrelevant; they can be arbitrarily bad. It is only important that in the current state the agents can rely on each other not to violate ASP disjointness, and, as discussed previously, this can be done strictly with knowledge of system dynamics. Under the assumption of full observability, $\mathcal{A}$ can then incorporate the state of other agents into its own transition function, effectively centralizing the decision process. Thus, G is now equivalent to a fully-observable, centralized, single-agent system. In other words, G is an MDP. ■

Now, if ASP disjointness cannot be guaranteed, G must remain a POSG:

Theorem 5. *Without ASP disjointness, G must be modeled as a POSG.*

Proof: By Theorem 2 says that in the absence of ASP disjointness, coordination among agents is required to maintain any non-collision guarantee. In the worst case, the agent ASRs may intersect in a way that requires more than two agents to coordinate contingency plans. Due to communication limitations, however, this necessarily induces partial observability of the contingency plans, and therefore the state, of at least one of the agents. Further, reasoning about future actions involving other agents requires consideration of non-shared reward functions because they are what determine the distribution over future actions. Under these conditions, the decision process is decentralized, multi-agent, and partially-observable with non-shared reward functions. By definition this is a POSG. ■

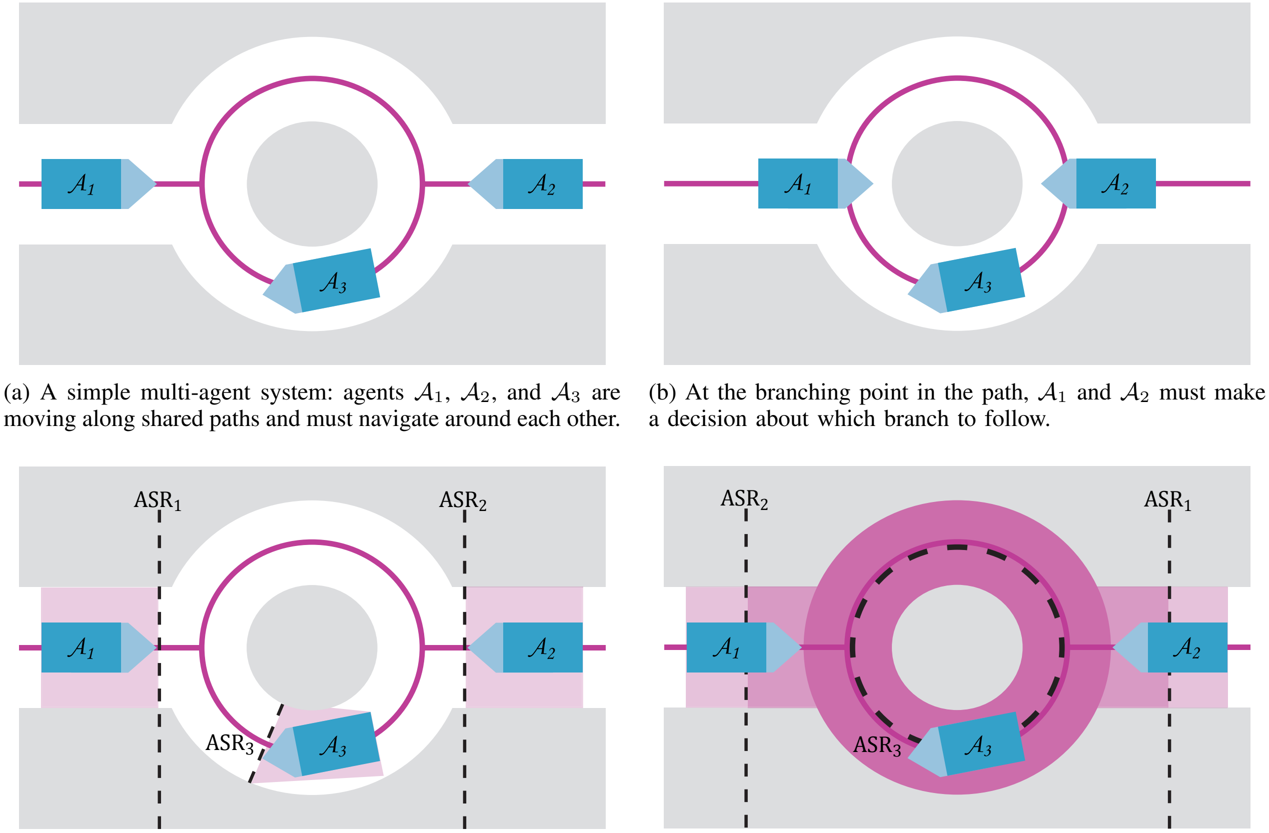
As covered in §II the complexity difference between MDP's and POSG's is staggering, with the former falling into complexity class P, and the latter into NEXP in the cooperative case or NEXP$^{\text{NP}}$ in the non-cooperative case. The fact that the complexity of the system can be manipulated to keep it within a tractable realm simply by controlling the dynamics is both surprising and powerful, and may provide insight into how humans are capable of efficiently and successfully navigating complex, multi-agent systems. In the case of roadways, for instance, the environment constrains the set of motions to such an extent that virtually any forward motion ensures progress toward the goal, so optimality of the plan is of little value. Instead, if the focus is on maintaining ASP disjointness in the system, a planning problem that is, in principle, wildly intractable becomes comfortably tractable.

To summarize, this section has shown that system dynamics alone can be responsible for moving a problem between two types of problem models, and so have shown that the dynamics of a system can fundamentally change both the complexity class and model space of the problem. This is the main result of the paper, and is, to the best knowledge of the author, novel within the field of multi-agent collision avoidance.

D. Exemplar Problem

This section presents a simple multi-agent system that can be manipulated in certain ways to clearly demonstrate the ideas of this paper. The problem is as follows: three agents $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ are traversing a fixed path that splits around a single median (Figure 3a). All agents can only move along the path or one of its branches. $\mathcal{A}_1$ and $\mathcal{A}_2$ desire to make it past the median, and $\mathcal{A}_3$ desires to stay around the median. As $\mathcal{A}_1$ and $\mathcal{A}_2$ traverse the path, they reach a point where they must make a decision about how to proceed (Figure 3b). Note that because only the collision avoidance problem is of concern, it is not a criterion for success that the agents can make it past each other successfully. Success only requires that they remain collision free, so, for instance, a deadlock situation satisfies the requirements, even if it is not the most desirable outcome.

Suppose the agents occupy a first-order system. In this case, their ASRs are disjoint unless or until they actually collide. Theorem 3 guarantees that, for any initial velocity, they can all proceed without coordination while maintaining the guarantee that collision is not inevitable (Figure 3c). On the other hand, assume the agents occupy a second-order system. For a sufficiently high initial velocity, none of the three agents have an ASP that can be guaranteed to be disjoint of all other ASRs (Figure 3d). It should be clear both by inspection and by Theorem 2 that maintaining any non-collision guarantee is *only* possible in this case if they somehow coordinate.



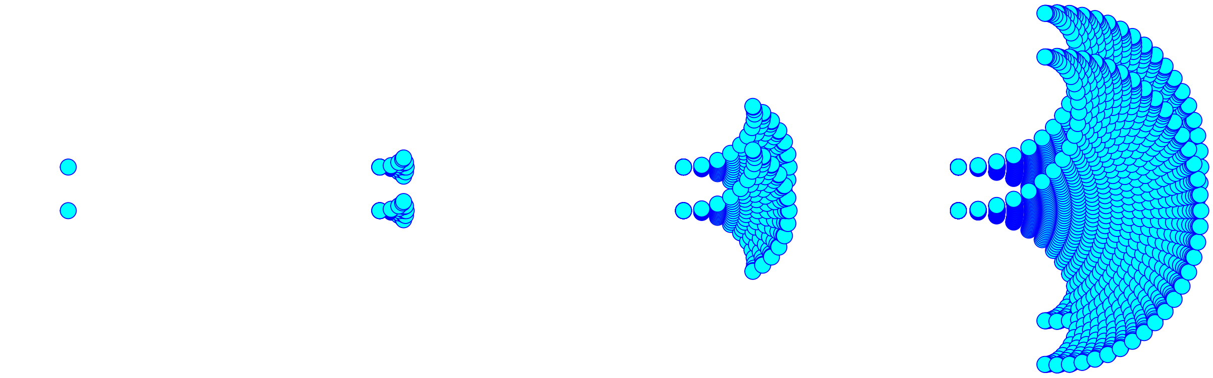
(a) A simple multi-agent system: agents $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ are moving along shared paths and must navigate around each other.

(b) At the branching point in the path, $\mathcal{A}_1$ and $\mathcal{A}_2$ must make a decision about which branch to follow.

(c) In a first-order system, the ASR's are invariant to the state of the system, and only extend longitudinally to the extents of the agents. The pink regions illustrate the extent of the ASR along the path being followed. In such a scenario, the ASP disjointness property always holds for any initial state.

(d) A second-order system with a high initial velocity results in an initial system state without a guarantee of ASP disjointness. ASR$_1$ and ASR$_2$ extend beyond the median, and ASR$_3$ circles the median. In such a scenario, there is significant overlap in the ASR's for each agent, indicated by the darker shaded regions.

Fig. 3: Exemplar problem. ASRs are indicated as exaggerated pink regions for each agent with dashed lines indicating extents.



(a) $v_{ref} = v$

(b) $v_{ref} = 0.66v$

(c) $v_{ref} = 0.33v$

(d) $v_{ref} = 0$

Fig. 4: For a given agent state $\mathcal{A}(s)$, reference velocity v_r, and set of followable paths P, the *relative stopping region* RSR$(\mathcal{A}(s), v_r, P)$ is the ASR computed with respect to the reference velocity v_r. This figure illustrates RSRs for various reference velocities v_{ref} in a two agent system. The two disc agents are traveling on a 2D plane with the same velocity v. In (a) the reference velocity is taken as v, so the stopping regions are the agents themselves. In (b)-(d) the reference velocity is taken as progressively smaller fractions of v. Here agents obey the same dynamics as in in Figure 2.

E. Generalizing from Absolute to Relative Stopping Regions

So far ASPs and ASRs have been constructed in terms of the agent coming to an absolute zero velocity state. However, this is not strictly necessary. A stopping region defined in terms of a *relative* zero velocity state is more general and can allow ASP disjointness to be exploited in a wider range of systems, in particular, those systems consisting of agents moving at high absolute velocities, but low relative velocities. This wider application is possible due to the fact that stopping regions defined in terms of relative velocities can be much smaller than their absolute counterparts. With only minor refinement in the existing definitions, these *relative* stopping regions will now be defined.

To derive relative stopping regions, the reference frame used in Problem 1 is extended to include a velocity:

Definition 9. A *velocity reference frame* is a spatial reference frame that is translationally offset from a parent frame over time according to a constant reference velocity. Unless otherwise noted, the phrase "velocity frame" will refer to velocity reference frame.

Rather than restricting agents to a fixed, global reference frame, they are allowed to choose among any velocity frame. Naturally, this induces a dependence of the coordination requirement on the reference velocity. Lemma 7 extends Theorem 2 to account for this:

Lemma 7. *A multi-agent system is guaranteed to be able to remain collision free without coordination if there exists a velocity frame such that Theorem 2 holds.*

Proof: Suppose there exists a velocity frame $\mathcal{F}$ such that Theorem 2 holds. Let $\mathcal{F}$ be the shared reference frame. Theorem 2 can now be applied. ■

Relative stopping regions can now be defined:

Definition 10. For a given agent state $\mathcal{A}(s)$, reference velocity v_r, and set of followable paths P, the *relative stopping region* $\text{RSR}(\mathcal{A}(s), v_r, P)$ is the ASR computed for the velocity frame with reference velocity v_r (Figure 4).

The relative stopping path (RSP) is defined similarly. Note that there's no incompatibility between the results derived in this paper and the notion of relative stopping versus absolute stopping; the reference frame chosen for any problem is arbitrary to begin with. Formally including the reference frame in the definition simply acknowledges that fact.

The next section will discuss future work and possible applications for these results.

V. Future Work & Applications

As stated before, RSPs and RSRs are especially useful for high-absolute/low-relative velocity systems, and one very important class of such systems is the highway (or freeway, or motorway) system. One formulation for the highway problem would be to plan motions that minimize RSR intersections by treating the minimization as a minimum constraint removal problem (Hauser [16], Erickson and LaValle [9]). Such a formulation would allow fine-grained control over the degree of coordination necessary and with whom it must be done.

In this paper, agents are assumed to have no knowledge of how others move aside from their dynamic capabilities, but this could be generalized to give all agents access to a set of predefined rules. For instance, right-of-way rules could be defined that allow the need for coordination to be removed from more complex interactions because agents following those rules are guaranteed to have contingency plans. This achieves the same result as the ASP disjointness condition while allowing more complex interactions. In fact, the ASP disjointness condition itself is essentially a pre-defined rule, but one that is derivable strictly from the physical properties of the system. Such an extension to incorporate arbitrary rules would tie together the results of this paper with the "social laws" ideas presented by Shoham and Tennenholtz [28] and would be an interesting and valuable avenue of investigation.

It seems also plausible that the applicability of the results could be expanded even more by extending the stopping region concept to include *holding patterns*, which are control trajectories of agents that follow some fixed pattern. Coordination requirements could then be derived for systems consisting of airplane-like agents. Additionally, for systems with low absolute and low relative velocities (such as navigating among pedestrians), low-energy collisions may be permissible, and the "stopping" region concept could be amended to define terminal states of non-zero velocities.

Problem 1 also does not explicitly specify a discrete or continuous time system. In practice, most systems are discrete time. It should be straightforward to amend these results to deal with them explicitly, and the incorporation of system timing will be vital for any practical application. The work of Bekris et al. [6] dealt extensively with the general problem of discrete time steps in multi-agent systems and would serve as an excellent resource.

VI. Conclusions

This paper presented a novel result showing that system dynamics can have a direct impact on both the theoretical complexity and solution space of multi-agent collision avoidance problems. The result is based on the fact that a requirement for agent coordination in a multi-agent system can fundamentally alter the problem model, and it was shown that system dynamics alone can add or remove this requirement. The proof of this assertion is constructive in nature, which allows the coordination requirement to be quantified, which, to the best of the author's knowledge, is also a novel contribution. An exemplar problem was given to demonstrate the results and then future work and applications were discussed.

References

[1] Ross E. Allen, Ashley A. Clark, Joseph A. Starek, and Marco Pavone. A machine learning approach for real-time reachability analysis. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems,*

Chicago, IL, USA, September 14-18, 2014, pages 2202–2208. IEEE, 2014. doi: 10.1109/IROS.2014.6942859. URL http://dx.doi.org/10.1109/IROS.2014.6942859.

[2] Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Paul A. Beardsley, and Roland Siegwart. Optimal reciprocal collision avoidance for multiple nonholonomic robots. In Alcherio Martinoli, Francesco Mondada, Nikolaus Correll, Grégory Mermoud, Magnus Egerstedt, M. Ani Hsieh, Lynne E. Parker, and Kasper Støy, editors, *Distributed Autonomous Robotic Systems - The 10th International Symposium, DARS 2010, Lausanne, Switzerland, November 1-3, 2010*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 203–216. Springer, 2010. ISBN 978-3-642-32722-3. doi: 10.1007/978-3-642-32723-0_15. URL http://dx.doi.org/10.1007/978-3-642-32723-0_15.

[3] Ron Alterovitz, Thierry Siméon, and Kenneth Y. Goldberg. The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty. In Wolfram Burgard, Oliver Brock, and Cyrill Stachniss, editors, *Robotics: Science and Systems III, June 27-30, 2007, Georgia Institute of Technology, Atlanta, Georgia, USA*. The MIT Press, 2007. ISBN 978-0-262-52484-1. URL http://www.roboticsproceedings.org/rss03/p30.html.

[4] Andrew Bacha, Cheryl Bauman, Ruel Faruque, Michael Fleming, Chris Terwelp, Charles F. Reinholtz, Dennis Hong, Al Wicks, Thomas Alberi, David Anderson, Stephen Cacciola, Patrick Currier, Aaron Dalton, Jesse Farmer, Jesse Hurdus, Shawn Kimmel, Peter King, Andrew Taylor, David Van Covern, and Mike Webster. Odin: Team victortango's entry in the DARPA urban challenge. *J. Field Robotics*, 25(8):467–492, 2008. doi: 10.1002/rob.20248. URL http://dx.doi.org/10.1002/rob.20248.

[5] K. E. Bekris. Avoiding inevitable collision states: Safety and computational efficiency in replanning with sampling-based algorithms. May 2010 2010. URL http://www.cs.rutgers.edu/~kb572/pubs/ics_tradeoffs.pdf.

[6] Kostas E. Bekris, Devin K. Grady, Mark Moll, and Lydia E. Kavraki. Safe distributed motion coordination for second-order systems with different planning cycles. *I. J. Robotic Res.*, 31(2):129–150, 2012. doi: 10.1177/0278364911430420. URL http://dx.doi.org/10.1177/0278364911430420.

[7] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Math. Oper. Res.*, 27(4):819–840, 2002. doi: 10.1287/moor.27.4.819.297. URL http://dx.doi.org/10.1287/moor.27.4.819.297.

[8] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In Yoav Shoham, editor, *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge, De Zeeuwse Stromen, The Netherlands, March 17-20 1996*, pages 195–210. Morgan Kaufmann, 1996. ISBN 1-55860-417-0.

[9] Lawrence H. Erickson and Steven M. LaValle. A simple, but NP-hard, motion planning problem. In Marie desJardins and Michael L. Littman, editors, *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*. AAAI Press, 2013. ISBN 978-1-57735-615-8. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6280.

[10] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *I. J. Robotic Res.*, 17(7):760–772, 1998. doi: 10.1177/027836499801700706. URL http://dx.doi.org/10.1177/027836499801700706.

[11] Thierry Fraichard. Trajectory planning in a dynamic workspace: a 'state-time space' approach. *Advanced Robotics*, 13(1):75–94, 1998. doi: 10.1163/156855399X00928. URL http://dx.doi.org/10.1163/156855399X00928.

[12] Thierry Fraichard and Hajime Asama. Inevitable collision states - a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004. doi: 10.1163/1568553042674662. URL http://dx.doi.org/10.1163/1568553042674662.

[13] Judy Goldsmith and Martin Mundhenk. Competition adds complexity. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 561–568. Curran Associates, Inc., 2007. URL http://papers.nips.cc/paper/3163-competition-adds-complexity.

[14] Dan Halperin and Micha Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete & Computational Geometry*, 16 (2):121–134, 1996. doi: 10.1007/BF02716803. URL http://dx.doi.org/10.1007/BF02716803.

[15] John Harding, Gregory Powell, Rebecca Yoon, Joshua Fikentscher, Charlene Doyle, Dana Sade, Mike Lukuc, Jim Simons, and Jing Wang. Vehicle-to-vehicle communications: Readiness of v2v technology for application. Technical Report DOT HS 812 014, U.S. Department of Transportation, National Highway Transportation Safety Administration, National Highway Traffic Safety Administration, 1200 New Jersey Avenue SE. Washington, DC 20590, August 2014. URL http://www.nhtsa.gov/staticfiles/rulemaking/pdf/V2V/Readiness-of-V2V-Technology-for-Application-812014.pdf.

[16] Kris K. Hauser. The minimum constraint removal problem with three robotics applications. In Emilio Frazzoli, Tomás Lozano-Pérez, Nicholas Roy, and Daniela Rus, editors, *Algorithmic Foundations of Robotics X - Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics, WAFR 2012, MIT, Cam-*

bridge, Massachusetts, USA, June 13-15 2012, volume 86 of *Springer Tracts in Advanced Robotics*, pages 1–17. Springer, 2012. ISBN 978-3-642-36278-1. doi: 10.1007/978-3-642-36279-8_1. URL http://dx.doi.org/10.1007/978-3-642-36279-8_1.

[17] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. 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, 1984. URL http://ijr.sagepub.com/content/3/4/76.short.

[18] Jeff Johnson and Kris K. Hauser. Optimal acceleration-bounded trajectory planning in dynamic environments along a specified path. In *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pages 2035–2041. IEEE, 2012. ISBN 978-1-4673-1403-9. doi: 10.1109/ICRA.2012.6225233. URL http://dx.doi.org/10.1109/ICRA.2012.6225233.

[19] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *I. J. Robotic Res.*, 32(9-10):1194–1227, 2013. doi: 10.1177/0278364913484072. URL http://dx.doi.org/10.1177/0278364913484072.

[20] Subbarao Kambhampati, Mark R. Cutkosky, Marty Tenenbaum, and Soo Hong Lee. Combining specialized reasoners and general purpose planners: A case study. In Thomas L. Dean and Kathleen McKeown, editors, *Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14-19, 1991, Volume 1.*, pages 199–205. AAAI Press / The MIT Press, 1991. ISBN 0-262-51059-6. URL http://www.aaai.org/Library/AAAI/1991/aaai91-032.php.

[21] Steven M. LaValle and James J. Kuffner. Randomized kinodynamic planning. *I. J. Robotic Res.*, 20(5):378–400, 2001. doi: 10.1177/02783640122067453. URL http://dx.doi.org/10.1177/02783640122067453.

[22] Emmanuel Mazer, Juan Manuel Ahuactzin, and Pierre Bessière. The ariadne's clew algorithm. *J. Artif. Intell. Res. (JAIR)*, 9:295–316, 1998. doi: 10.1613/jair.468. URL http://dx.doi.org/10.1613/jair.468.

[23] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Hähnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, Doug Johnston, Stefan Klumpp, Dirk Langer, Anthony Levandowski, Jesse Levinson, Julien Marcil, David Orenstein, Johannes Paefgen, Isaac Penny, Anna Petrovskaya, Mike Pflueger, Ganymed Stanek, David Stavens, Antone Vogt, and Sebastian Thrun. Junior: The stanford entry in the urban challenge. *J. Field Robotics*, 25(9):569–597, 2008. doi: 10.1002/rob.20258. URL http://dx.doi.org/10.1002/rob.20258.

[24] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3689975.

[25] Stéphane Petti and Thierry Fraichard. Safe motion planning in dynamic environments. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2-6, 2005*, pages 2210–2215. IEEE, 2005. ISBN 0-7803-8912-3. doi: 10.1109/IROS.2005.1545549. URL http://dx.doi.org/10.1109/IROS.2005.1545549.

[26] John H. Reif. Complexity of the mover's problem and generalizations (extended abstract). In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 421–427. IEEE Computer Society, 1979. doi: 10.1109/SFCS.1979.10. URL http://dx.doi.org/10.1109/SFCS.1979.10.

[27] John H. Reif and Micha Sharir. Motion planning in the presence of moving obstacles. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 144–154. IEEE Computer Society, 1985. ISBN 0-8186-0644-4. doi: 10.1109/SFCS.1985.36. URL http://dx.doi.org/10.1109/SFCS.1985.36.

[28] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artif. Intell.*, 73(1-2):231–252, 1995. doi: 10.1016/0004-3702(94)00007-N. URL http://dx.doi.org/10.1016/0004-3702(94)00007-N.

[29] Christopher Urmson, Joshua Anhalt, J. Andrew (Drew) Bagnell, Christopher R. Baker, Robert E Bittner, John M Dolan, David Duggins, David Ferguson , Tugrul Galatali, Hartmut Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert , Thomas Howard, Alonzo Kelly , David Kohanbash, Maxim Likhachev , Nick Miller, Kevin Peterson, Ragunathan Rajkumar, Paul Rybski, Bryan Salesky, Sebastian Scherer, Young-Woo Seo, Reid Simmons , Sanjiv Singh, Jarrod M Snider, Anthony (Tony) Stentz , William (Red) L. Whittaker, and Jason Ziglar. Tartan racing: A multi-modal approach to the darpa urban challenge. Technical Report CMU-RI-TR-, Robotics Institute, Pittsburgh, PA, April 2007. URL http://www.ri.cmu.edu/publication_view.html?pub_id=6906.

[30] Aris Valtazanos and Subramanian Ramamoorthy. Online motion planning for multi-robot interaction using composable reachable sets. In Thomas Röfer, Norbert Michael Mayer, Jesus Savage, and Uluc Saranli, editors, *RoboCup 2011: Robot Soccer World Cup XV [papers from the 15th Annual RoboCup International Symposium, Istanbul, Turkey, July 2011]*, volume 7416 of *Lecture Notes in Computer Science*, pages 186–197. Springer, 2011. ISBN 978-3-642-32059-0. doi: 10.1007/978-3-642-32060-6_16. URL http://dx.doi.org/10.1007/978-3-642-32060-6_16.

[31] Jur van den Berg, Stephen J. Guy, Ming C. Lin, and Dinesh Manocha. Reciprocal *n*-body collision avoidance. In Cédric Pradalier, Roland Siegwart, and Gerhard Hirzinger, editors, *Robotics Research - The 14th International Symposium, ISRR 2009, August 31*

- *September 3, 2009, Lucerne, Switzerland*, volume 70 of *Springer Tracts in Advanced Robotics*, pages 3–19. Springer, 2009. ISBN 978-3-642-19456-6. doi: 10.1007/978-3-642-19457-3_1. URL http://dx.doi.org/10.1007/978-3-642-19457-3_1.

[32] Jur P. van den Berg, Jamie Snape, Stephen J. Guy, and Dinesh Manocha. Reciprocal collision avoidance with acceleration-velocity obstacles. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 3475–3482. IEEE, 2011. doi: 10.1109/ICRA.2011.5980408. URL http://dx.doi.org/10.1109/ICRA.2011.5980408.

[33] Josh L. Wilkerson, Jim Bobinchak, Michael Culp, Josh Clark, Tyler Halpin-Chan, Katia Estabridis, and Gary Hewer. Two-dimensional distributed velocity collision avoidance. Technical Report NAWCWD TP 8786, Physics Division, Research and Intelligence Department, Naval Air Warfare Center Weapons Division, China Lake, CA 93555-6100, November 2014. URL http://www.dtic.mil/dtic/tr/fulltext/u2/a598520.pdf.

[34] David Wilkie, Jur P. van den Berg, and Dinesh Manocha. Generalized velocity obstacles. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, MO, USA*, pages 5573–5578. IEEE, 2009. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354175. URL http://dx.doi.org/10.1109/IROS.2009.5354175.

[35] Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. Trajectory planning for bertha - A local, continuous method. In *2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, June 8-11, 2014*, pages 450–457. IEEE, 2014. doi: 10.1109/IVS.2014.6856581. URL http://dx.doi.org/10.1109/IVS.2014.6856581.