

Conservative Collision Prediction and Avoidance for Stochastic Trajectories in Continuous Time and Space

Jan-P. Calliess, Michael Osborne and Stephen J. Roberts^{*}
Dept. of Engineering Science
Oxford University
United Kingdom

ABSTRACT

Existing work in multi-agent collision prediction and avoidance typically assumes discrete-time trajectories with Gaussian uncertainty or that are completely deterministic. We propose an approach that allows detection of collisions even between continuous, stochastic trajectories with the only restriction that means and covariances can be computed. To this end, we employ probabilistic bounds to derive criterion functions whose negative sign provably is indicative of probable collisions. For criterion functions that are Lipschitz, an algorithm is provided to rapidly find negative values or prove their absence. We propose an iterative policy-search approach that avoids prior discretisations and yields collision-free trajectories with adjustably high certainty. We test our method with both fixed-priority and auction-based protocols for coordinating the iterative planning process. Results are provided in collision-avoidance simulations of feedback controlled plants.

1. INTRODUCTION

Due to their practical importance, multi-agent collision avoidance and control have been extensively studied across different communities including AI, robotics and control. Considering continuous stochastic trajectories, reflecting each agent's uncertainty about its neighbours' time-indexed locations in an environment space, we exploit a distribution-independent bound on collision probabilities to develop a conservative collision-prediction module. It avoids temporal discretisation by stating collision-prediction as a one-dimensional optimization problem. If mean and covariances are computable Lipschitz functions of time, one can derive Lipschitz constants that allow us to guarantee collision prediction success with low computational effort. This is often the case, for instance, when dynamic knowledge of the involved trajectories is available (e.g. maximum velocities or even the SDEs).

^{*}The authors gratefully acknowledge funds via EPSRC EP/I011587.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*
Copyright © 2014, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

To avoid collisions detected by the prediction module, we let an agent re-plan repeatedly until no more collisions occur with a definable probability. Here, re-planning refers to modifying a control signal (influencing the expected basin of attraction and equilibrium point of the agent's stochastic dynamics) so as to bound the collision probability while seeking low plan execution cost in expectation. To keep the exposition concrete, we focus our descriptions on an example scenario where the plans correspond to sequences of setpoints of a feedback controller regulating an agent's noisy state trajectory. However, one can apply our method in the context of more general policy search problems.

In order to ensure collision avoidance yields low social cost across the entire agent collective, we compare two different coordination mechanisms. Firstly, we consider a simple fixed-priority scheme [10], and secondly, we modify an auction-based coordination protocol [5] to work in our continuous setting. In contrast to pre-existing work in auction-style multi-agent planning (e.g. [5, 15]) and multi-agent collision avoidance (e.g. [1, 2, 14]), we avoid *a priori* discretizations of space and time. Instead, we recast the coordination problem as one of incremental open-loop policy search. That is, as a succession of continuous optimization or root-finding problems that can be efficiently and reliably solved by modern optimisation and root-finding techniques (e.g. [12, 21]).

While our current experiments were conducted with linear stochastic differential equation (SDE) models with state-independent noise (yielding Gaussian processes), our method is also applicable to any situation where mean and covariances can be evaluated. This encompasses non-linear, non-Gaussian cases that may have state-dependent uncertainties (cf. [11]).

1.1 Related Work

Multi-agent trajectory planning and task allocation methods have been related to auction mechanisms by identifying locations in state space with atomic goods to be auctioned in a sequence of repeated coordination rounds (e.g. [5, 15, 24]). Unfortunately, even in finite domains the coordination is known to be intractable – for instance the sequential allocation problem is known to be NP-hard in the number of goods and agents [13, 20]. Furthermore, collision avoidance corresponds to non-convex interactions.

This renders the coordination problem inapplicable to standard optimization techniques that rely on convexity of the joint state space. In recent years, several works have investigated the use of mixed-integer programming techniques for single- and multi-agent model-predictive control with col-

lision avoidance both in deterministic and stochastic settings [5, 18]. To connect the problem to pre-existing mixed-integer optimization tools these works had to limit the models to dynamics governed by linear, time-discrete difference equations with state-independent state noise. The resulting plans were finite sequences of control inputs that could be chosen freely from a convex set. The controls gained from optimization are open-loop – to obtain closed-loop policies the optimization problems have to be successively re-solved on-line in a receding horizon fashion. However, computational effort may prohibit such an approach in multi-agent systems with rapidly evolving states.

Furthermore, prior time-discretisation comes with a natural trade-off. On the one hand, one would desire a high temporal resolution in order to limit the chance of missing a collision predictably occurring between consecutive time steps. On the other hand, communication restrictions, as well as poor scalability of mixed-integer programming techniques in the dimensionality of the input vectors, impose severe restrictions on this resolution. To address this trade-off, [9] proposed to interpolate between the optimized time steps in order to detect collisions occurring between the discrete time-steps. Whenever a collision was detected they proposed to augment the temporal resolution by the time-step of the detected collision thereby growing the state-vectors incrementally as needed. A detected conflict, at time t , is then resolved by solving a new mixed-integer linear programme over an augmented state space, now including the state at t . This approach can result in a succession of solution attempts of optimization problems of increasing complexity, but can nonetheless prove relatively computationally efficient. Unfortunately, their method is limited to linear, deterministic state-dynamics. Another thread of works relies on dividing space into polytopes [1, 16], while still others [7, 8, 14, 19] adopt a potential field. In not accommodating uncertainty and stochasticity, these approaches are forced to be overly conservative in order to prevent collisions in real systems.

In contrast to all these works, we will consider a different scenario. We focus on the assumption that each agent is regulated by influencing its continuous stochastic dynamics. For instance, we might have a given feedback controller with which one can interact by providing a sequence of setpoints constituting the agent’s plan. While this restricts the choice of control action, it also simplifies computation as the feedback law is fixed. The controller can generate a continuous, state-dependent control signal based on a discrete number of control decisions, embodied by the setpoints. Moreover, it renders our method applicable in settings where the agents’ plants are controlled by standard off-the-shelf controllers (such as the omnipresent PID-controllers) rather than by more sophisticated customized ones. Instead of imposing discreteness, we make the often more realistic assumption that agents follow continuous time-state trajectories within a given continuous time interval. Unlike most work [1, 19, 23, 25] in this field, we allow for stochastic dynamics, where each agent cannot be certain about the location of its team-members. This is crucial for many real-world multi-agent systems. The uncertainties are modelled as state-noise which can reflect physical disturbances or merely model inaccuracies. While our exposition’s focus is on stochastic differential equations, our approach is generally applicable in all contexts where the first two moments of the predicted trajectories can be evaluated for all time-steps.

2. PREDICTIVE PROBABILISTIC COLLISION DETECTION WITH CRITERION FUNCTIONS

Task. Our aim is to design a collision-detection module that can decide whether a set of (predictive) stochastic trajectories is collision-free (in the sense defined below). The module we will derive is guaranteed to make this decision correctly, based on knowledge of the first and second order moments of the trajectories alone. In particular, no assumptions are made about the family of stochastic processes the trajectories belong to. As the required collision probabilities will generally have to be expressed as non-analytic integrals, we will content ourselves with a fast, *conservative* approach. That is, we are willing to tolerate a non-zero false-alarm-rate as long as decisions can be made rapidly and with zero false-negative rate. Of course, for certain distributions and plant shapes, one may derive closed-form solutions for the collision probability that may be less conservative and hence, lead to faster termination and shorter paths. In such cases, our derivations can serve as a template for the construction of criterion functions on the basis of the tighter probabilistic bounds.

Problem Formalization. Formally, a collision between two objects (or agents) \mathbf{a}, \mathbf{r} at time $t \in I := [t_0, t_f] \subset \mathbb{R}$ can be described by the event

$\mathcal{C}^{\mathbf{a}, \mathbf{r}}(t) = \{(x^{\mathbf{a}}(t), x^{\mathbf{r}}(t)) \mid \|x^{\mathbf{a}}(t) - x^{\mathbf{r}}(t)\|_2 \leq \frac{\Lambda^{\mathbf{a}} + \Lambda^{\mathbf{r}}}{2}\}$. Here, $\Lambda^{\mathbf{a}}, \Lambda^{\mathbf{r}}$ denote the objects’ diameters, and $x^{\mathbf{a}}, x^{\mathbf{r}} : I \rightarrow \mathbb{R}^D$ are two (possibly uncertain) trajectories in a common, D -dimensional interaction space.

In a stochastic setting, we desire to bound the collision probability below a threshold $\delta \in (0, 1)$ at any given time in I . We loosely say that the trajectories are *collision-free* if $\Pr[\mathcal{C}^{\mathbf{a}, \mathbf{r}}(t)] < \delta, \forall t \in I$.

Approach. For conservative collision detection between two agents’ stochastic trajectories $x^{\mathbf{a}}, x^{\mathbf{r}}$, we construct a *criterion function* $\gamma^{\mathbf{a}, \mathbf{r}} : I \rightarrow \mathbb{R}$ (eq. as per Eq. 1 below). A conservative criterion function has the property $\gamma^{\mathbf{a}, \mathbf{r}}(t) > 0 \Rightarrow \Pr[\mathcal{C}^{\mathbf{a}, \mathbf{r}}(t)] < \delta^{\mathbf{a}}$. That is, a collision between the trajectories with probability above δ can be ruled-out if $\gamma^{\mathbf{a}, \mathbf{r}}$ attains only positive values. This result is the essence of Thm. 2.3. Being conservative, we assume a collision occurs unless $\min_{t \in I} \gamma^{\mathbf{a}, \mathbf{r}}(t) > 0, \forall \mathbf{r} \neq \mathbf{a}$. If the trajectories’ means and covariances are Lipschitz functions of time then one can show that $\gamma^{\mathbf{a}, \mathbf{r}}$ is Lipschitz as well. In such cases negative values of $\gamma^{\mathbf{a}, \mathbf{r}}$ can be found or ruled out rapidly, as will be discussed in Sec. 2.1. In situations where a Lipschitz constant is unavailable or hard to determine, we can base our detection on the output of a global minimization method such as DIRECT [12].

2.1 Finding negative function values of Lipschitz functions

Let $a, b \in \mathbb{R}, a \leq b, I := [a, b] \subset \mathbb{R}$. Assume we are given a Lipschitz continuous *target function* $f : I \rightarrow \mathbb{R}$ with Lipschitz constant $L \geq 0$. That is, $\forall S \subset I \exists L_S \leq L \forall x, x' \in S : |f(x) - f(x')| \leq L_S |x - x'|$. Let $a = t_1 < t_2 < \dots < t_N = b$ and define $G_N = \{t_1, \dots, t_N\}$ to be the *sample grid* of size $N \geq 2$ consisting of the inputs at which we choose to evaluate the target f .

Our goal is to prove or disprove the existence of a negative function value of target f .

2.1.1 An adaptive algorithm

We can define two functions, *ceiling* u_N and *floor* l_N , such that (i) they bound the target $\forall t \in I : l_N(t) \leq f(t) \leq u_N(t)$, and (ii) the bounds get tighter for denser grids. In particular, one can show that $l_N, u_N \xrightarrow{N \rightarrow \infty} f$ uniformly if G_N converges to a dense subset of $[a, b]$. It has been shown that $\xi^l := \arg \min_{i=1}^{N-1} \min_{x \in [t_i, t_{i+1}]} l_N(x) = \min_{i=1}^{N-1} \frac{t_{i+1} + t_i - f(t_{i+1}) - f(t_i)}{2L}$, $l_N(\xi^l) = \min_i \frac{f(t_{i+1}) + f(t_i)}{2} - L \frac{t_{i+1} - t_i}{2}$ (see [12, 21]). It is trivial to refine this, to take localised Lipschitz constants into account: $\xi^l = \min_i \frac{f(t_{i+1}) + f(t_i)}{2} - L_{J_i} \frac{t_{i+1} - t_i}{2}$ where L_{J_i} is a Lipschitz number valid on interval $J_i = (t_i, t_{i+1})$.

This suggests the following algorithm: *We refine the grid by including a new sample $\xi^l, f(\xi^l)$ as defined above, until either we find a negative function value of f , or until $l_N(\xi^l) \geq 0$ (in which case we are guaranteed that no negative function values can exist).*

An example run is depicted in Fig. 1. Note, without our stopping criteria, our algorithm degenerates to Shubert's minimization method [21]. The stopping criteria are important to save computation, especially in the absence of negative function values.

2.2 Deriving collision criterion functions

This subsection is dedicated to the derivation of a (Lipschitz) criterion function. For ease of notation, we omit the time index t . For instance, in this subsection, x^a now denotes random variable $x^a(t)$ rather than the stochastic trajectory.

The next thing we will do is to derive sufficient conditions for absence of collisions, i.e. for $\Pr[\mathfrak{C}^{a,\tau}] < \delta$.

To this end, we make an intermediate step: For each agent $q \in \{a, \tau\}$ we define an open hyper-cuboid H^q centred around mean $\mu^q = \langle x^q(t) \rangle$. As a D -dimensional hyper-cuboid, H^q is completely determined by its centre point μ^q and its edge lengths l_1^q, \dots, l_D^q . Let O^q denote the event that $x^q \notin H^q$ and $P^q := \Pr[O^q]$. We derive a simple disjunctive constraint on the component distances of the means under which we can guarantee that the collision probability is not greater than the probability of at least one object being outside its hyper-cuboid. This is the case if the hypercuboids do not overlap. That is, their max-norm distance is at least $\Lambda^{a,\tau} := \frac{\Lambda^a + \Lambda^\tau}{2}$.

Theorem 2.1. *Let μ_j^q denote the j th component of object q 's mean and $r_j^q = \frac{1}{2}l_j^q$. Assume, x^a, x^τ are random variables with means $\mu^a = \langle x^a \rangle, \mu^\tau = \langle x^\tau \rangle$, respectively. The max-norm distance between hypercuboids H^a, H^τ is at least $\Lambda^{a,\tau} > 0$ (i.e. the hypercuboids do not overlap), which is expressed by the following disjunctive constraint:*

$$\exists j \in \{1, \dots, D\} : |\mu_j^a - \mu_j^\tau| > \Lambda^{a,\tau} + r_j^a + r_j^\tau.$$

Then, we have :

$$\Pr[\mathfrak{C}^{a,\tau}] \leq P^a + P^\tau - P^\tau P^a \leq P^a + P^\tau$$

where $P^q = \Pr[x^q \notin H^q]$, ($q \in \{a, \tau\}$).

Proof. See [6]. \square

One way to define a criterion function is as follows:

$$\gamma^{a,\tau}(t) := \max_{i=1, \dots, D} \{|\mu_i^a - \mu_i^\tau| - \Lambda^{a,\tau} - r_i^a - r_i^\tau\}. \quad (1)$$

For more than two agents, agent a 's overall criterion function is $\Gamma^a(t) := \min_{\tau \in \mathfrak{A} \setminus \{a\}} \gamma^{a,\tau}(t)$.

Thm. 2.1 tells us that the collision probability is bounded below the desired threshold δ if $\gamma^{a,\tau}(t) > 0$, provided we chose the radii r_j^a, r_j^τ ($j = 1, \dots, D$) such that $P^a, P^\tau \leq \frac{\delta}{2}$.

Let $q \in \{a, \tau\}$. Probability theory provides several distribution-independent bounds relating the radii of a hypercuboid to the probability of not falling into it. That is, these are bounds of the form

$$P^q \leq \beta(r_1^q, \dots, r_D^q; \Theta)$$

where β decreases monotonically with the radii and Θ represents additional information. In the case of Chebyshev-type bounds information about the first two moments are folded in, i.e. $\Theta = (\mu^q, C^q)$ where $C^q(t) \in \mathbb{R}^{D \times D}$ is the covariance matrix.

Setting $\frac{\delta}{2} \geq \beta(r_1^q, \dots, r_D^q; \Theta)$, we solve for the largest radii that fulfil the inequality (we desire to find large radii as this decreases the criterion function and hence, conservativeness of our criterion function (see Eq. 1). Due to monotonicity of β these will be the radii r_1^q, \dots, r_D^q such that $\frac{\delta}{2} = \beta(r_1^q, \dots, r_D^q; \Theta)$. As the final construction step of the criterion function, we insert these radii into Eq. 1.

Consider the following concrete example. Combining union bound and the standard (one-dim.) Chebyshev bound yields

$P^q = \Pr[x^q \notin H^q] \leq \sum_{j=1}^D \frac{C_{jj}^q}{r_j^q r_j^q} =: \beta(r_1^q, \dots, r_D^q; C^q)$. Setting every radius, except r_i^q , infinitely large values and β equal to $\frac{\delta}{2}$ yields $\frac{\delta}{2} = \frac{C_{ii}^q}{r_i^q r_i^q}$, i.e. $r_i^q = \frac{2C_{ii}^q}{\delta}$. Finally, inserting these radii (for $q = a, \tau$) into Eq. 1 yields our first collision criterion function: $\gamma^{a,\tau}(t) := |\mu_i^a - \mu_i^\tau| - \Lambda^{a,\tau} - \frac{2C_{ii}^a}{\delta} - \frac{2C_{ii}^\tau}{\delta}$.

Of course, this argument can be made for any choice of dimension i . Hence, a less conservative, yet valid, choice is

$$\gamma^{a,\tau}(t) := \max_{i=1, \dots, D} \{|\mu_i^a - \mu_i^\tau| - \Lambda^{a,\tau} - \frac{2C_{ii}^a}{\delta} - \frac{2C_{ii}^\tau}{\delta}\}. \quad (2)$$

Notice, this function has the desirable property of being Lipschitz continuous, provided the mean $\mu_i^q : I \rightarrow \mathbb{R}$ and variance functions $C_{ii}^q : I \rightarrow \mathbb{R}_+$ are. In particular, it is easy to show $L(\gamma^{a,\tau}) \leq \max_{i=1, \dots, D} L(\mu_i^a) + L(\mu_i^\tau) + \frac{2}{\delta} (L(C_{ii}^a) + L(C_{ii}^\tau))$ where, as before, $L(f)$ is the best Lipschitz constant of function f .

For the special case of two dimensions, we can derive a less conservative alternative criterion function based on a tighter two-dimensional Chebyshev-type bound [26]:

Theorem 2.2 (Alternative collision criterion function). *Let spatial dimensionality be $D = 2$. Choosing*

$$r_i^q(t) := \sqrt{\frac{1}{2\delta^a}} \sqrt{C_{ii}^q(t) + \frac{\sqrt{C_{ii}^q(t)C_{jj}^q(t)(C_{ii}^q(t)C_{jj}^q(t) - (C_{ij}^q(t))^2)}}{C_{jj}^q(t)}}$$

($q \in \{a, \tau\}, i \in \{1, 2\}, j \in \{1, 2\} - \{i\}$) in Eq. 1 yields a valid distribution-independent criterion function. That is, $\gamma^{a,\tau}(t) > 0 \Rightarrow \Pr[\mathfrak{C}^{a,\tau}(t)] < \delta^a$.

Proof. Refer to [6]. \square

The proof, as well as a Lipschitz constant (for non-zero uncertainty), is provided in the report version of this paper [6]. Note, the Lipschitz constant we have derived therein becomes infinite in the limit of vanishing variance. In that case, the presence of negative criterion values can be tested

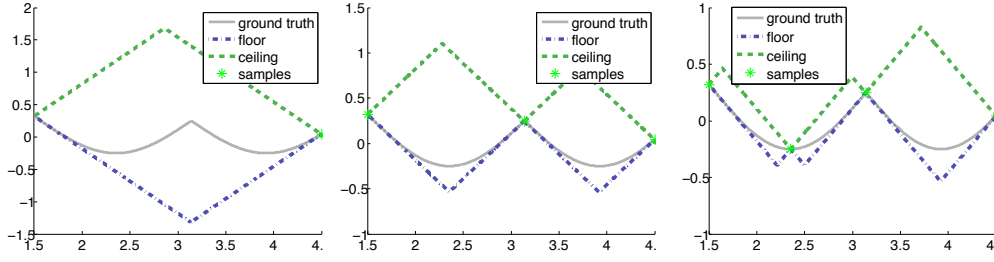


Figure 1: Proving the existence of a negative value of function $x \mapsto |\sin(x)|\cos(x) + \frac{1}{4}$. Left: Initial condition. Centre: First refinement. Right: The second refinement has revealed the existence of a negative value.

based on the sign of the minimum of the criterion function. This can be found employing a global optimiser.

2.2.1 Multi-agent case.

Let $\mathbf{a} \in \mathfrak{A}$, $\mathfrak{A}' \subset \mathfrak{A}$ such that $\mathbf{a} \notin \mathfrak{A}'$ a subset of agents. We define the event that \mathbf{a} collides with at least one of the agents in \mathfrak{A}' at time t as $\mathfrak{C}^{\mathbf{a}, \mathfrak{A}'}(t) := \{(x^{\mathbf{a}}(t), x^{\mathbf{r}}(t)) | \exists \mathbf{r} \in \mathfrak{A}' : \|x^{\mathbf{a}}(t) - x^{\mathbf{r}}(t)\|_2 \leq \Lambda\} = \bigcup_{\mathbf{r} \in \mathfrak{A}'} \mathfrak{C}^{\mathbf{a}, \mathbf{r}}$. By union bound, $\Pr[\mathfrak{C}^{\mathbf{a}, \mathfrak{A}'}(t)] \leq \sum_{\mathbf{r} \in \mathfrak{A}'} \Pr[\mathfrak{C}^{\mathbf{a}, \mathbf{r}}(t)]$.

Theorem 2.3 (Multi-Agent Criterion). *Let $\gamma^{\mathbf{a}, \mathbf{r}}$ be valid criterion functions defined w.r.t. collision bound $\delta^{\mathbf{a}}$. We define multi-agent collision criterion function $\Gamma^{\mathbf{a}, \mathfrak{A}'}(t) := \min_{\mathbf{r} \in \mathfrak{A}'} \gamma^{\mathbf{a}, \mathbf{r}}(t)$. If $\Gamma^{\mathbf{a}, \mathfrak{A}'}(t) > 0$ then the collision probability with \mathfrak{A}' is bounded below $\delta^{\mathbf{a}}|\mathfrak{A}'|$. That is, $\Pr[\mathfrak{C}^{\mathbf{a}, \mathfrak{A}'}(t)] < \delta^{\mathbf{a}}|\mathfrak{A}'|$.*

Moreover, $\Gamma^{\mathbf{a}, \mathfrak{A}'}$ is Lipschitz if the constituent functions $\gamma^{\mathbf{a}, \mathbf{r}}$ are [6].

Our distribution-independent collision criterion functions have the virtue that they work for all distributions – not only the omnipresent Gaussian. Unfortunately, distribution-independence is gained at the price of conservativeness (ref. to Fig. 2). In our experiments in Sec. 4, the collision criterion function as per Thm. 2.2 is utilized as an integral component of our collision avoidance mechanisms. The results suggest that the conservativeness of our detection module does not entail prohibitively high-false-alarm rates for the distribution-independent approach to be considered impractical. That said, whenever distributional knowledge can be converted into a criterion function. One could then use our derivations as a template to generate refined criterion functions using Eq. 1 with adjusted radii r_i, r_j , reflecting the distribution at hand.

3. COLLISION AVOIDANCE

In this section we outline the core ideas of our proposed approach to multi-agent collision avoidance. After specifying the agent’s dynamics and formalizing the notion of a single-agent plan, we define the multi-agent planning task. Then we describe how conflicts, picked-up by our collision prediction method, can be resolved. In Sec. 3.1 we describe the two coordination approaches we consider utilizing to generate conflict-free plans.

I) Model (example). We assume the system contains a set \mathfrak{A} of agents indexed by $\mathbf{a} \in \{1, \dots, |\mathfrak{A}|\}$. Each agent \mathbf{a} ’s associated plant has a probabilistic state trajectory following stochastic controlled D -dimensional state dynamics

(we consider the case $D = 2$) in the continuous interval of (future) time $I = (t_0, t_f]$. We desire to ask agents to adjust their policies to avoid collisions. Each policy gives rise to a stochastic belief over the trajectory resulting from executing the policy. For our method to work, all we require is that the trajectory’s mean function $m : I \rightarrow \mathbb{R}^D$ and covariance matrix function $\Sigma : I \rightarrow \mathbb{R}^{D \times D}$ are evaluable for all times $t \in I$.

A prominent class for which closed-form moments can be easily derived are linear stochastic differential equations (SDEs). For instance, we consider the SDE

$$dx^{\mathbf{a}}(t) = K(\xi^{\mathbf{a}}(t) - x^{\mathbf{a}}(t))dt + B dW \quad (3)$$

where $K, B \in \mathbb{R}^{D \times D}$ are matrices $x^{\mathbf{a}} : I \rightarrow \mathbb{R}^D$ is the state trajectory and W is a vector-valued Wiener process. Here, $u(x^{\mathbf{a}}; \xi^{\mathbf{a}}) := K(\xi^{\mathbf{a}} - x^{\mathbf{a}})$ could be interpreted as the control policy of a linear feedback-controller parametrised by $\xi^{\mathbf{a}}$. It regulates the state to track a desired trajectory $\xi^{\mathbf{a}}(t) = \zeta_0^{\mathbf{a}}\chi_{\{0\}}(t) + \sum_{i=1}^{H^{\mathbf{a}}} \zeta_i^{\mathbf{a}}\chi_{\tau_i^{\mathbf{a}}}(t)$ where $\chi_{\tau_i^{\mathbf{a}}} : \mathbb{R} \rightarrow \{0, 1\}$ denotes the indicator function of the half-open interval $\tau_i^{\mathbf{a}} = (t_{i-1}^{\mathbf{a}}, t_i^{\mathbf{a}}] \subset [0, T^{\mathbf{a}}]$ and each $\zeta_i^{\mathbf{a}} \in \mathbb{R}^D$ is a *setpoint*. If K is positive definite the agent’s state trajectory is determined by setpoint sequence $p^{\mathbf{a}} = (t_i^{\mathbf{a}}, \zeta_i^{\mathbf{a}})_{i=0}^{H^{\mathbf{a}}}$ (aside from the random disturbances) which we will refer to as the agent’s *plan*. For example, plan $p^{\mathbf{a}} := ((t_0, x_0^{\mathbf{a}}), (t_f, x_f^{\mathbf{a}}))$ could be used to regulate agent \mathbf{a} ’s *start state* $x_0^{\mathbf{a}}$ to a given *goal state* $x_f^{\mathbf{a}}$ between times t_0 and t_f . For simplicity, we assume the agents are always initialized with plans of this form before coordination commences.

One may interpret a setpoint as some way to alter the stochastic trajectory. Below, we will determine setpoints that modify a stochastic trajectory to reduce collision probability while maintaining low expected cost. From the vantage point of policy search, $\xi^{\mathbf{a}}$ is agent \mathbf{a} ’s policy parameter that has to be adjusted to avoid collisions.

II) Task. Each agent \mathbf{a} desires to find a sequence of setpoints ($p^{\mathbf{a}}$) such that (i) it moves from its start state $x_0^{\mathbf{a}}$ to its goal state $x_f^{\mathbf{a}}$ along a low-cost trajectory and (ii) such that along the trajectory its plant (with diameter Δ) does not collide with any other agents’ plant in state space with at least a given probability $1 - \delta \in (0, 1)$.

III) Collision resolution. An agent seeks to avoid collisions by adding new setpoints to its plan until the collision probability of the resulting state trajectory drops below threshold δ . For choosing these new setpoints we consider two methods **WAIT** and **FREE**. In the first method the agents insert a time-setpoint pair $(t, x_0^{\mathbf{a}})$ into the previous plan $p^{\mathbf{a}}$. Since this aims to cause the agent to wait

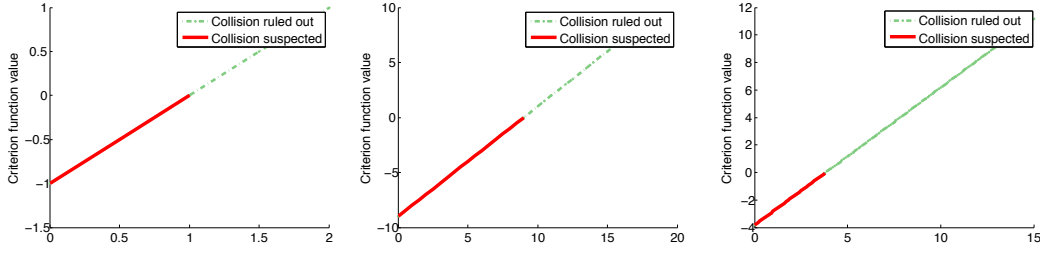


Figure 2: Criterion function values (as per Eq. 2) as a function of $\|\langle x^r \rangle - \langle x^a \rangle\|_\infty$ and with $\delta = 0.05$, $\Lambda^{a,r} = 1$. Left: variances $C^a = C^r = \text{diag}(.00001, .00001)$. Centre: variances $C^a = C^r = \text{diag}(.1, .1)$. Right: variances $C^a = C^r = \text{diag}(.1, .1)$ and with improved criterion function (as per Thm. 2.2).

at its start location x_0^a we will call the method WAIT. It is possible that multiple such insertions are necessary until collisions are avoided. Of course, if a higher-priority agent decides to traverse through x_0^a , this method is too rigid to resolve a conflict. In the second method the agent optimizes for the time and location of the new setpoint. Let $p_{\uparrow(t,s)}^a$ be the plan updated by insertion of time-setpoint pair $(t, s) \in I \times \mathbb{R}^D$. We propose to choose the candidate setpoint (t, s) that minimizes a function being a weighted sum of the expected cost entailed by executing updated plan $p_{\uparrow(t,s)}^a$ and a hinge-loss collision penalty $c_{\text{coll}}^a(p_{\uparrow(t,s)}^a) := \lambda \max\{0, -\min_t \Gamma^a(t)\}$. Here, Γ^a is computed based on the assumption we were to execute $p_{\uparrow(t,s)}^a$ and $\lambda \gg 0$ determines the extent to which collisions are penalized. Since the new setpoint can be chosen freely in time and state-space we refer to the method as FREE.

3.1 Coordination

We will now consider how to integrate our collision detection and avoidance methods into a coordination framework that determines who needs to avoid whom and at what stage of the coordination process. Such decisions are known to significantly impact the *social cost* (i.e. the sum of all agents' individual costs) of the agent collective.

Fixed-priorities (FP). As a baseline method for coordination we consider a basic fixed-priority method (e.g. [3,10]). Here, each agent has a unique ranking (or priority) according to its index \mathbf{a} (i.e. agent 1 has highest priority, agent $|\mathcal{A}|$ lowest). When all higher-ranking agents are done planning, agent \mathbf{a} is informed of their planned trajectories which it has to avoid with a probability greater than $1 - \delta$. This can be done by repeatedly invoking for collision detection and resolution methods described above until no further collision with higher-ranking agents are found.

Lazy Auction Protocol (AUC). While the FP method is simple and fast the rigidity of the fixed ranking can lead to sub-optimal social cost and coordination success. Furthermore, its sequential nature does not take advantage of possible parallelization a distributed method could. To alleviate this we propose to revert the ranking flexibly on a case-by-case basis. In particular, the agents are allowed to compete for the right to gain passage (e.g. across a region where a collision was detected) by submitting bids in the course of an auction. The structure of the approach is outlined in Alg. 1.

Assume an agent \mathbf{a} detects a collision at a particular time step t_{coll} and invites the set of agents $C^a = \{\mathbf{r} | \gamma^{a,r}(t_{\text{coll}}) \leq 0\}$

input : Agents $\mathbf{a} \in \mathcal{A}$, cost functions c^a , dynamics, initial start and goal states, initial plans $p^1, \dots, p^{|\mathcal{A}|}$.
output: collision-free plans $p^1, \dots, p^{|\mathcal{A}|}$.

```

repeat
  for  $\mathbf{a} \in \mathcal{A}$  do
    [ $\text{flag}^a, C^a, t_{\text{coll}}$ ]  $\leftarrow$  CollDetecta( $\mathbf{a}, \mathcal{A} - \{\mathbf{a}\}$ )
    if  $\text{flag}^a = 1$  then
      winner  $\leftarrow$  Auction( $C^a \cup \{\mathbf{a}\}, t_{\text{coll}}$ )
      foreach  $\mathbf{r} \in (C^a \cup \{\mathbf{a}\}) - \{\text{winner}\}$  do
         $p^r \leftarrow$  Avoidr( $(C^a \cup \{\mathbf{a}\}) - \{\mathbf{r}\}, t_{\text{coll}}$ )
        Broadcastr( $p^r$ )
      end
    end
  end
end

```

until $\forall \mathbf{a} \in \mathcal{A} : \text{flag}^a = 0$;

Algorithm 1: Lazy auction coordination method (AUC) (written in a sequentialized form). Collisions are resolved by choosing new setpoints to enforce collision avoidance. C^a : set of agents detected to be in conflict with agent \mathbf{a} . flag^a : collision detection flag ($=0$, iff no collision detected). t_{coll} : earliest time where a collision was detected. Avoid: collision resolution method updating the plan by a single new setpoint according to WAIT or FREE.

to join an auction to decide who needs to avoid whom. The auction determines a winner who is not required to alter his plan. The losing agents need to insert a new setpoint into their respective plans designed to avoid all other agents in C^a while keeping the plan cost function low. The idea is to design the auction rules as a heuristic method to minimize the social cost of the ensuing solution. To this end, we define the bids such that their magnitude is proportional to a heuristic magnitude of the expected regret for losing and not gaining passage. That is agent \mathbf{a} submits a bid $b^a = l^a - s^a$. Magnitude l^a is defined as \mathbf{a} 's anticipated cost $c_{\text{plan}}^a(p_{\uparrow(t,s)}^a)$ for the event that the agent will not secure "the right of passage" and has to create a new setpoint (t, s) (according to (III)) tailored to avoid all other agents engaged in the current auction. On the other hand, $s^a := c_{\text{plan}}^a(p^a)$ is the cost of the unchanged plan p^a . If there is a tie among multiple agents the agent with the lowest index among the highest bidders wins.

Acknowledging that $s^{\text{winner}} + \sum_{\mathbf{a} \neq \text{winner}} l^a$ is an estimated social cost (based on current beliefs of trajectories) after the auction, we see that the winner determination rule greedily attempts to minimize social cost: $b^{\text{winner}} \geq b^r \Leftrightarrow \forall \mathbf{r} : s^r + \sum_{\mathbf{a} \neq \mathbf{r}} l^a \geq s^{\text{winner}} + \sum_{\mathbf{a} \neq \text{winner}} l^a$.

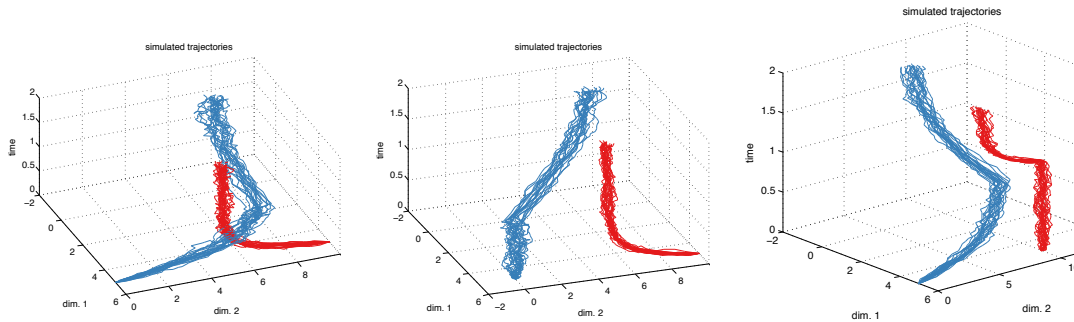


Figure 3: EXP1. Draws from uncoordinated agents’ plans (left), after coordination and collision resolution with methods FP-WAIT (centre) and AUC-WAIT (right).

Recorded quantity	Experiment 1			Experiment 2		
	NONE	AUC-WAIT	FP-WAIT	NONE	AUC-FREE	FP-FREE
est. Pr[collision] [%]	78	0	0	51	0	0
averaged path length away from goal	13.15	13.57	12.57	14.94	16.22	18.13
averaged sqr. dist. of final state to goal	0.05	0.04	25.8	0.05	0.05	0.05
no. of resolution rounds	0	6	3	0	4	4

Table 1: Quantities estimated based on 100 draws from SDEs simulating executions of the different plans in EXP1 and EXP2. Notice, our collision avoidance methods succeed in preventing collisions. In EXP1 the FP-WAIT method failed to reach its first goal in time which is reflected in the *sqr. distance to goal* measure. Note the discrepancies in avg. path length are relatively low due to convexity effects and the contribution of state noise to the path lengths.

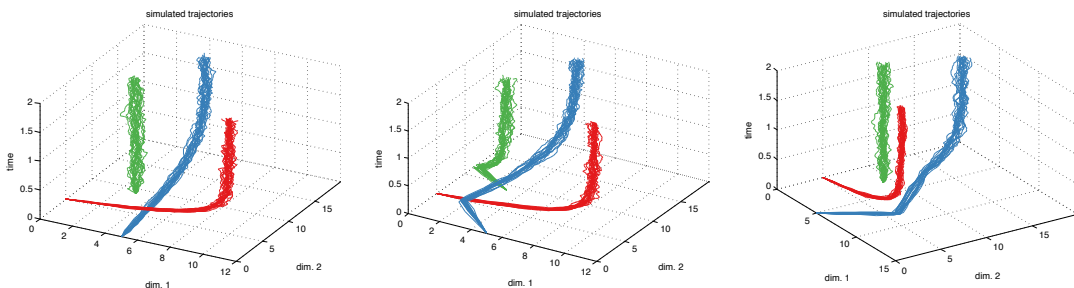


Figure 4: EXP2. Draws from uncoordinated agents’ plans (left), after coordination and collision resolution with methods FP-FREE (centre) and AUC-FREE (right).

4. SIMULATIONS

As a first test, we simulated three simple multi-agent scenarios, *EXP1*, *EXP2* and *EXP3*. Each agent’s dynamics were an instantiation of an SDE of the form of Eq. 3. We set δ to achieve collision avoidance with certainty greater than 95%. Collision prediction was based on the improved criterion function as per Thm. 2.2. During collision resolution with the FREE method each agent \mathbf{a} assessed a candidate plan $p^{\mathbf{a}}$ according to cost function $c_{plan}^{\mathbf{a}}(p^{\mathbf{a}}) = w_1 c_{traj}^{\mathbf{a}}(p^{\mathbf{a}}) + w_2 c_{miss}^{\mathbf{a}}(p^{\mathbf{a}}) + w_3 c_{coll}^{\mathbf{a}}(p^{\mathbf{a}})$. Here $c_{traj}^{\mathbf{a}}$ is a heuristic to penalize expected control energy or path length; in the second summand, $c_{miss}^{\mathbf{a}}(p^{\mathbf{a}}) = \|x^{\mathbf{a}}(t_f) - x_f^{\mathbf{a}}\|^2$ penalizes expected deviation from the goal state; the third term $c_{coll}^{\mathbf{a}}(p^{\mathbf{a}})$ penalizes collisions (cf. III). The weights are design parameters which we set to $w_1 = 10$, $w_2 = 10^3$ and $w_3 = 10^6$, emphasizing avoidance of mission failure and collisions.

Note, if our method was to be deployed in a receding horizon fashion, the parameters could also be adapted online, for instance, using standard learning techniques (e.g. [17, 22]).

EXP1. Collision resolution was done with the WAIT method to update plans. Draws from the SDEs with the initial plans of the agents are depicted in Fig. 3 (left). The curves represent 20 noisy trajectories of agents 1 (red) and 2 (blue). Each curve is a draw from the stochastic differential dynamics obtained by simulating the execution of the given initial plan. The trajectories were simulated with the Euler-Maruyama method for a time interval of $I = [0s, 2s]$. The spread of the families of curves is due to the random disturbances each agent’s controller had to compensate for during runtime.

Agent 1 desired to control the state from start state $x_0^1 = (5, 10)$ to goal $x_f^1 = (5, 5)$. Agent 2 desired to move from start state $x_0^2 = (5, 0)$ via intermediate goal $x_{f_1}^2 = (5, 7)$ (at 1s) to final goal state $x_{f_2}^2 = (0, 7)$. While the agents meet their goals under the initial plans, their execution would imply a high probability of colliding around state (5, 6) (cf. Fig. 3 (left), Tab. 1). Coordination with fixed priorities (1 (red) > 2 (blue)) yields conflict-free plans (Fig. 3 (cen-

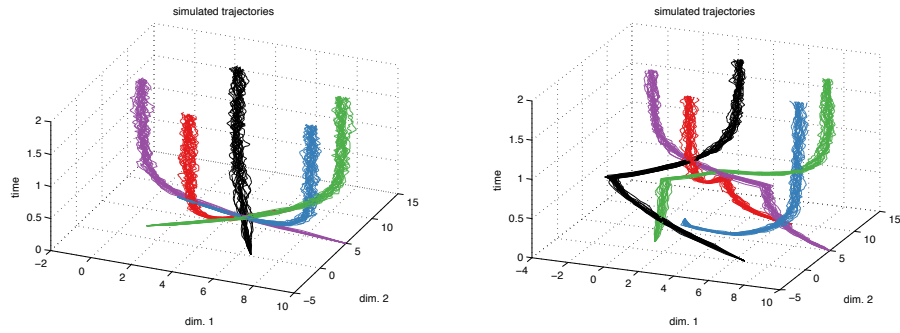


Figure 5: Ex. of EXP3 with 5 agents. Draws from uncoordinated agents’ plans (left), after coordination and collision resolution with methods AUC-FREE (right).

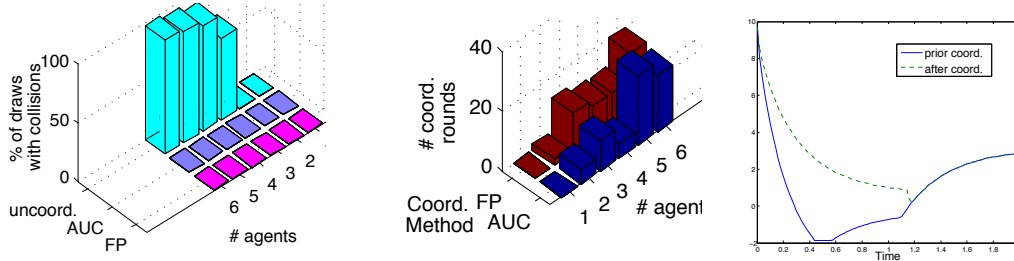


Figure 6: Left, Centre: Recorded results for EXP3 with 1 to 6 agents. Note, all collisions were successfully avoided. Right: EXP1. Criterion functions for collision detection of agent 2 before and after coordination. The graphs report the presence/absence of a collision as indicated by their sign.

tre)). However, agent 2 is forced to wait too long at its start location to be able to reach intermediate waypoint $x_{f,1}^2$ in time and therefore, decides to move directly to its second goal. This could spawn high social cost due to missing one of the designated goals (Tab. 1). By contrast, the auction method is flexible enough to reverse the ranking at the detected collision point causing agent 1 to wait instead of 2 (Fig. 3 (right)). Thereby, agent 2 is able to reach both of its goal states in time. This success is reflected by low social cost (see Tab. 1).

EXP2. The setup was analogous to EXP1 but with three agents and different start and goal states as depicted in Fig. 4. Furthermore, collisions were avoided with the FREE method with 10 random initializations of the local optimizer. Coordination of plans with fixed priorities (1 (red) > 2 (blue) > 3 (green)) caused 2 to avoid agent 1 by moving to the left. Consequently, 3 now had to temporarily leave its start and goal state to get out of the way (see Fig. 4 (centre)). With two agents moving to avoid collisions social cost was relatively high (see Tab. 1). During coordination with the auction-based method agent 2 first chose to avoid agent 1 (as in the FP method). However, losing the auction to agent 3 at a later stage of coordination, agent 2 decided to finally circumvent 1 by arcing to the right instead of to the left. This allowed 3 to stay in place (see Tab. 1).

EXP3. Next, we conducted a sequence of experiments for varying numbers of agents ranging from $|\mathcal{A}| = 1, \dots, 7$. In each experiment all agents’ start locations were placed on a circle. Their respective goals were placed on the opposite ends of the circle. The eigenvalues of the feedback gain matrices of each agent were drawn at random from a uniform distribution on the range [2,7]. An example situation for an experiment with 5 agents is depicted in Fig. 5. Collision avoidance was achieved.

Note, that despite this setting being close to worst case (i.e. almost all agents try to traverse a common, narrow corridor) the coordination overhead is moderate (see Fig. 6, centre). Also, all collisions were successfully avoided (see Fig. 6, left).

5. CONCLUSIONS

This work considered multi-agent planning under stochastic uncertainty and non-convex chance-constraints for collision avoidance. In contrast to pre-existing work, we did not need to rely on prior space or time-discretisation. This was achieved by deriving criterion functions with the property that the collision probability is guaranteed to be below a freely definable threshold $\delta \in (0, 1)$ if the criterion function attains no negative values. Thereby, stochastic collision detection is reduced to deciding whether such negative values exist. For Lipschitz criterion functions, we provided an algorithm for making this decision rapidly. We described a general procedure for deriving criterion functions and presented two such functions based on Chebyshev-type bounds. The advantage of using Chebyshev inequalities is their independence of the underlying distribution. Therefore, our approach is applicable to any stochastic state noise model for which the first two moments can be computed at arbitrary time steps. In particular, this would apply to models with state-dependent uncertainty and non-convex chance constraints which, to the best of our knowledge, have not been successfully approached in the multi-agent control literature. Nonetheless, future work could build on our results and derive less conservative criterion functions by using more problem-specific probabilistic inequalities. For

instance, in simple cases such as additive Gaussian noise, tighter bounds can be given [4] and used in Eq. 1.

To enforce collision avoidance, our method modified the agent's plans until no collisions could be detected. To coordinate the detection and avoidance efforts of the agents, we employed an auction-based as well as a fixed-priority method.

Our experiments are a first indication that our approach can succeed in finding collision-free plans with high-certainty with the number of required coordination rounds scaling mildly in the number of agents. While in its present form, the coordination mechanism does not come with a termination guarantee, in none of our simulations have we encountered an infinite loop. For graph routing, [5] provides a termination guarantee of the lazy auction approach under mild assumptions. Current work considers if their analysis can be extended to our continuous setting. Moreover, if required, our approach can be combined with a simple stopping criterion that terminates the coordination attempt when a computational budget is expended or an infinite loop is detected.

The computation time within each coordination round depends heavily on the time required for finding a new set-point and for collision detection. This involves minimizing $(t, s) \mapsto c_{plan}^a(p_{\uparrow}^a(t, s))$ and c_{coll}^a , respectively. The worst-case complexity depends on the choice of cost functions, their domains and the chosen optimizer. Fortunately, we can draw on a plethora of highly advanced global optimisation methods (eg [12, 21]) guaranteeing rapid optimization success. In terms of execution time, we can expect considerable alleviations from implementation in a compiled language. Furthermore, the collision detection and avoidance methods are based on global optimization and thus, would be highly amenable to parallel processing – this could especially benefit the auction approach.

While our exposition was focussed on the task of defining setpoints of feedback-controlled agents, the developed methods can be readily applied to other policy search settings, where the first two moments of the probabilistic beliefs over the trajectories (that would result from applying the found policies) can be computed.

6. REFERENCES

- [1] N. Ayanian and V. Kumar. Decentralized feedback controllers for multiagent teams in environments with obstacles. *IEEE Trans. on Robotics*, 26(5), 2010.
- [2] D. Bareiss and J. van den Berg. Reciprocal collision avoidance for quadrotor helicopters using lqr-obstacles. In *AAAI-12 Workshop on Multiagent Pathfinding*, 2012.
- [3] M. Bennewitz, W. Burgard, and S. Thrun. Exploiting constraints during prioritized path planning for teams of mobile robots. In *IROS*, 2001.
- [4] L. Blackmore, H. Li, and B. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *American Control Conference*, 2006.
- [5] J. Calliess, D. Lyons, and U. Hanebeck. Lazy auctions for multi-robot collision avoidance and motion control under uncertainty. Technical Report PARG-01-11, Dept. of Engineering Science, University of Oxford, 2011.
- [6] J. Calliess, M. Osborne, and S. Roberts. Conservative collision prediction and avoidance for stochastic trajectories in continuous time and space. ArXiv, also available at: http://www.robots.ox.ac.uk/~parg/pubs/AAMASjms_arxivreport.pdf, 2014.
- [7] D.E. Chang, S.C. Shadden, J.E. Marsden, and R. Olfati-Saber. Collision avoidance for multiple agent systems. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, pages 539–543. IEEE, 2003.
- [8] D.V. Dimarogonas, S.G. Loizou, K.J. Kyriakopoulos, and M.M. Zavlanos. A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica*, 42(2):229–243, 2006.
- [9] M. G. Earl and R. D'Andrea. Iterative MILP Methods for Vehicle-Control Problems. *IEEE Trans. on Robotics*, 2005.
- [10] M. A. Erdmann and T. Lozano-Perez. On multiple moving objects. *Algorithmica*, 1987.
- [11] C. Gardiner. *Stochastic Methods-A Handbook for the Natural and Social Sciences*. Springer, 4th edition, 2009.
- [12] D.R. Jones, C.D. Peritunen, and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *JOTA*, 79(1), 1991.
- [13] Sven Koenig, Craig Tovey, X. Zheng, and I. Sungur. Sequential bundle-bid single-sale auction algorithms for decentralized control. In *IJCAI*, 2007.
- [14] D. Kostic, S. Adinandra, J. Caarls, and H. Nijmeijer. Collision-free motion coordination of unicycle multi-agent systems. In *American Control Conference (ACC), 2010*, pages 3186–3191. IEEE, 2010.
- [15] M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, S. Koenig, A. Kleywegt, C. Tovey, A. Meyerson, and S. Jain. Auction-based multi-robot routing. In *Int. Conf. on Robotics: Science and Systems*, 2005.
- [16] Y. Li and K. Gupta. Motion planning of multiple agents in virtual environments on parallel architectures. In *ICRA*, 2007.
- [17] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. In *IEEE Symposium on Foundations of Computer Science*, pages 256–261, 1989.
- [18] D. Lyons, J. Calliess, and U. Hanebeck. Chance constrained model predictive control for multi-agent systems with coupling constraints. In *American Control Conference (ACC)*, 2012.
- [19] S. Mastellone, D.M. Stipanović, C.R. Graunke, K.A. Intlekofer, and M.W. Spong. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *The International Journal of Robotics Research*, 27(1):107–126, 2008.
- [20] T. Sandholm. Algorithm for optimal winner determination on combinatorial auctions. *Artif. Int.*, 2002.
- [21] B. Shubert. A sequential method seeking the global maximum of a function. *SIAM J. on Numerical Analysis*, 9, 1972.
- [22] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian Process Optimization in the Bandit Setting : No Regret and Experimental Design. In *ICML*, 2010.
- [23] D.M. Stipanović, P.F. Hokayem, M.W. Spong, D.D. Šiljak, et al. Cooperative avoidance control for multiagent systems. *Journal of Dynamic Systems, Measurement, and Control*, 129:699, 2007.
- [24] C. Tovey, M. Lagoudakis, S. Jain, and S. Koenig. The generation of bidding rules for auction-based robot coordination. In *Multi-Robot Systems: From Swarms to Intelligent Automata*, volume 3, pages 3–14, 2005.
- [25] J. Van den Berg, M. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *ICRA*, 2008.
- [26] P. Whittle. A multivariate generalization of Tchebichev's inequality. *Quarterly Journal of Mathem.*, 1958.