Deadlock Analysis and Resolution in Multi-Robot Systems: The Two Robot Case

Jaskaran Grover¹, Changliu Liu¹, Katia Sycara ¹

Abstract—Collision avoidance for multirobot systems is a well studied problem. Recently, control barrier functions (CBFs) have been proposed for synthesizing decentralized controllers that guarantee collision avoidance (safety) and goal stabilization (performance) for multiple robots. However, it has been noted in several works that reactive control synthesis methods (such as CBFs) are prone to deadlock, an equilibrium of system dynamics that causes the robots to come to a standstill before they reach their goals. In this paper, we analyze the incidence of deadlocks in a multirobot system that uses CBFs for goal stabilization and collision avoidance. Our analysis is formal, in that we demonstrate that system deadlock is indeed the result of a force-equilibrium on robots. We show how to interpret deadlock as a subset of the state space and prove that this set is non-empty, bounded, of measure zero and located on the boundary of the safe set. Based on this analysis, we develop a decentralized three-phase algorithm that uses feedback linearization to ensure that the robots provably exit the deadlock set and converge to their goals while avoiding collisions. We show simulation results and experimentally validate the deadlock resolution algorithm on Khepera-IV robots.

Index Terms—Multi-Robot Systems, Robot Safety, Motion Control

I. INTRODUCTION

Multirobot systems have been studied thoroughly for solving a variety of complex tasks such as search and rescue [1], sensor coverage [2] and environmental exploration [3]. Global coordinated behaviors result from executing local control laws on individual robots interacting with their neighbors [4], [5]. Typically, the local controllers running on these robots are a combination of a task-based controller responsible for completion of a primary objective and a reactive collision avoidance controller. However, including a hand-engineered safety control no longer guarantees that the original task will be satisfied [6]. This problem becomes all the more pronounced when the number of robots increases. Motivated by this bottleneck, this paper focuses on algorithmic analysis of the performance-safety trade-offs that result from augmenting a task-based controller with a collision avoidance constraints as done using CBF based quadratic programs (QPs) [7]. Although CBF based QPs are not hand-engineered, rather they mediate between safety and performance in a rigorous way, yet ultimately they are reactive controllers. They locally (in a decentralized way) explore the acceleration/velocity space from which one admissible control is selected at every time instant. These local approaches exhibit a lack of look-ahead, which causes the robots to be trapped in deadlocks as noted in [8], [9], [10]. Deadlock occurs because robots reach a state where conflict becomes inevitable, i.e. a control favoring goal stabilization will violate safety (see solid red dot in fig. 1). Hence, the only feasible strategy is to remain static. Although once in deadlock, small perturbations can steer the system away from deadlock, there is no theoretical guarantee that the robots will not fall back in deadlock. To circumvent these issues, this work addresses the following technical questions: 1) What are the characteristics of a system in deadlock? 2) How can we leverage these characteristics to synthesize a controller that provably makes the system exit deadlock and reach its goal?

As a first step towards addressing these questions, in Section III, we review technical definitions for CBF based QPs [10] to synthesize controllers for collision avoidance and goal stabilization. In Section IV, we recall the definition of deadlock and analyze the KKT conditions for the QP when a robot is in deadlock. The mechanics interpretation of KKT conditions indeed reveals the intrinsic force-balance on a robot in deadlock (section V-A). We motivate a set theoretic interpretation of deadlock for the case of 𝑁 robots, treating it as a subset of the state space, with an eye towards designing a strategy that evades/exists this set. In Section V, we focus on the easier to analyze case for two robots and analyze the mathematical properties of the deadlock set. We prove that this set is on the boundary of the safe set and that it is non-empty, bounded and of measure zero. We believe that these properties offer insight for the characteristics of deadlock in the general 𝑁 robot case. Finally, in Section VI,

Fig. 1: Two robots moving towards each other fall in deadlock. System state converges to the boundary of safe set.
we use these properties to derive a provably-correct three-phase control strategy to make the robots exit deadlock and make them reach their goals. We demonstrate this strategy on different robot models in simulation, and experimentally on Khepera-IV robots. Finally, we conclude with directions for future work.

II. PRIOR WORK

Several existing approaches provide inspiration for the results presented here. Of these, two are especially relevant to our work. In the first category, we describe existing methods for collision avoidance and in the second we focus on prior work on deadlock resolution.

A. Prior Work on Avoidance Control

Avoidance control is a well-studied problem with immediate applications for planning collision-free motions for multirobot systems. Classical avoidance control assumes a worst case scenario with no cooperation between robots [11], [12]. Cooperative collision avoidance is explored in [13] where avoidance control laws are computed using value functions. A robust control formulation of this work is presented in [14] for robot models described by Lagrangian dynamics and bounded noise in actuator torques. Velocity obstacles have been proposed in [15] for motion planning in dynamic environments. They select avoidance maneuvers outside of robot's velocity obstacles to avoid static and moving obstacles by means of a tree-search. While this method is prone to undesirable oscillations, the authors in [16], [17], [18] propose reciprocal velocity obstacles that are immune to such oscillations. More recently, control barrier function based controllers have been used in [6], [10] to mediate between safety and performance using QPs. The QP computes a controller closest to a task-based control that guarantees collision avoidance.

B. Prior Work on Deadlock Resolution

The importance of coordinating motions of multiple robots while simultaneously ensuring safety, performance and deadlock prevention has been acknowledged in works as early as in [9]. Here, authors proposed scheduling algorithms to asynchronously coordinate motions of two manipulators to ensure that their trajectories remain collision and deadlock free in addition to reaching their goals. However, their approach relies on prior knowledge of the explicit geometric path in the workspace each robot should take (without assuming knowledge of the other robot's path). In the context of mobile robots, [19] identified the presence of deadlocks in a cooperative scenario using mobile robot troops. To the best of our knowledge, [20] were the first to propose algorithms for deadlock resolution specifically for multiple mobile robots. Their strategy for collision avoidance modifies planned paths by inserting idle times and resolves deadlocks by asking the trajectory planners of each robot to plan an alternative trajectory until deadlock is resolved. Authors in [21] proposed coordination graphs to resolve deadlocks in robots navigating through narrow corridors. Authors in [22] added perturbation terms to their optimal controllers for avoiding deadlock. Similarly, authors in [10] use consistent perturbations and modify the QP to synthesize controllers which evade deadlock.

Differently from these works, we characterize the analytical properties of states in deadlock. We explicitly analyze controls from CBF based QPs and demonstrate that intuitive explanations for system in deadlock are indeed obtained from analysis using duality. We do not consider additive perturbations for resolving deadlocks since there are no formal guarantees for such methods. Instead, we use feedback linearization to synthesize a provably correct controller that ensures safety, performance and deadlock resolution.

III. AVOIDANCE CONTROL WITH CBFs: REVIEW

In this section, we review CBF based QPs used for synthesizing controllers that mediate between safety (collision avoidance) and performance (goal-stabilization) for multirobot systems. We refer the reader to [10] for a comprehensive treatment on this subject, since our work builds on top of their approach. Assume that we have $N$ mobile robots, each of which follows double-integrator dynamics:

$$\begin{bmatrix} \dot{p}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} v_i \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i$$

(1)

where $p_i = (x_i, y_i) \in \mathbb{R}^2$ represents the position of robot $i$, $v_i \in \mathbb{R}^2$ represents its velocity and $u_i \in \mathbb{R}^2$ represents the acceleration (i.e. control). The collective state of robot $i$ is denoted by $z_i = (p_i, v_i)$ and the collective state of the multirobot system is denoted as $Z = (z_1, z_2, \ldots, z_N)$. Assume that each robot has maximum allowable acceleration limits $|u_i| \leq \alpha_i$ that represent actuator constraints. The problem of goal stabilization with avoidance control requires that each robot $i$ must reach a goal $p_{d_i}$ while avoiding collisions with every other robot $j \neq i$.

For reaching a goal, assume that there is a prescribed PD controller $\dot{u}_i(z_i) = -k_p(p_i - p_{d_i}) - k_v v_i$ with $k_p, k_v > 0$. Although this controller by itself ensures exponential stabilization of each robot to its goal, there is no guarantee that the resulting trajectories will be collision free.

Based on [10], a pairwise safety constraint is formulated for every pair of robots to ensure mutual collision free motions. This constraint is mathematically posed by defining a safety function that maps the joint state space of a pair of robots $i$ and $j$ to a real-valued safety index $i.e. h : \mathbb{R}^4 \times \mathbb{R}^4 \rightarrow \mathbb{R}$. This index is defined as [10],

$$h_{ij} = \sqrt{2(\alpha_i + \alpha_j)(\|\Delta p_{ij}\| - D_s) + \Delta p_{ij}^T \Delta v_{ij} / \|\Delta p_{ij}\|}.$$  

(2)

Robots $i$ and $j$ are considered to be collision-free if their states $(z_i, z_j)$ are such that $h_{ij}(z_i, z_j) \geq 0$. We define the “safe set” as the zero level super set of $h_{ij} i.e. C_{ij} := \{(z_i, z_j) \in \mathbb{R}^8 \mid h_{ij}(z_i, z_j) \geq 0\}$. The boundary of the safe set is defined as:

$$\partial C_{ij} = \{(z_i, z_j) \in \mathbb{R}^8 \mid h(z_i, z_j) = 0\}.$$  

(3)

Assuming that the initial positions of robots are in the safe set $C_{ij}$, we would like to synthesize controls $u_i$ and $u_j$ that
ensure that future states of the robots $i$ and $j$ also stay in $C_{ij}$. This can be achieved by ensuring that
\[ \frac{dh_{ij}}{dt} \geq -\kappa(h_{ij}), \quad (4) \]
where we choose, $\kappa(h) := h^3$. For the given choice of $h$, constraint eq. (4) can be rewritten as compactly
\[
-\Delta p_{ij}^T \Delta u_{ij} \leq b_{ij}, \quad (5)
\]
and its Lagrange dual function is
\[
dh_{ij} \geq \kappa(h_{ij}),
\]
where (5) is said to be in deadlock if $u_i^* = 0$, $v_i = 0$, $\dot{u}_i \neq 0$ and $p_i \neq p_d$.

This definition states that for a robot to be in deadlock, the robot should be static i.e. its velocity should be zero, and the output from the QP based controller should also be zero, even though the prescribed PD controller reports non-zero acceleration since the robot is not at its intended goal. We now review the KKT conditions for the optimization problem in eq. (8) and then specifically analyze them in deadlock.

**Definition 1.** \([10]\) A robot $i$ is said to be in deadlock if $u_i^* = 0$, $v_i = 0$, $\dot{u}_i \neq 0$ and $p_i \neq p_d$.

From our numerical simulations, we observed that one scenario when deadlock occurs is when the initial states of a pair of robots and their goals are symmetric with respect to one another. That is to say that one scenario prone to deadlock in robots $i$ and $j$ is when the initial position of robot $i$ is the desired goal for robot $j$ and vice versa. To illustrate this point, Fig. 2 shows the states of two robots executing controllers based on eq. (8). Notice from fig. 2(a) that the positions of robots have converged, but not to their respective goals. Therefore, the outputs from the prescribed PD controller will still be non-zero after convergence. Additionally, the corresponding velocities have converged to zero (fig. 2(b)), and the control inputs from eq. (8) have converged to zero as well (fig. 2(c)). Following these observations, deadlock is defined as follows

**A. KKT Conditions**

As explained in Section III, we seek control inputs that guarantee collision avoidance and goal stabilization. Recall that each robot $i$ computes a control by solving a local QP as in eq. (8). For the following discussion, define $a_j := -\Delta p_{ij}$ and $b_j = \frac{\alpha_i}{\alpha_i + \alpha_j} b_{ij}$. To simplify notation, we will drop subscript $i$ and implicitly assume that the QP is being solved for the ego robot. Therefore, we rewrite eq. (8) as:

\[
\min_{\tilde{u}_i} \quad \|u_i - \tilde{u}_i(z_i)\|^2_2 \\
\text{subject to} \quad -\Delta p_{ij}^T u_i \leq \frac{\alpha_i}{\alpha_i + \alpha_j} b_{ij}, \quad j \in \{1, \ldots, M\} \quad (8)
\]

This QP has $(M + 4)$ linear constraints ($M$ from collision avoidance with $M$ neighbors and four from acceleration limits). Each robot $i$ executes a local version of this QP and computes its optimal $u_i^*$. The control generated by this QP represents a reactive collision avoidance mechanism. At every time step, $u_i^*$ depends on the state of the ego robot $z_i$ and its $M$ neighbors. As long as the QP remains feasible, the generated control $u_i^*$ ensures collision avoidance of robot $i$ with its neighbors. In the next section, we use this method to analyze the incidence of deadlocks due to this technique.

**IV. ANALYSIS FOR N ROBOT DEADLOCK**

We reviewed the problem formulation of multirobot collision avoidance and goal stabilization using the framework of CBF based QPs. In this section, we will show that this approach can result in deadlocks (depending on the initial conditions and goals of robots). Additionally, we want to analyze qualitative properties of a robot in deadlock. Towards that end, we will investigate the KKT conditions of the problem in eq. (8). Our goal is to use these properties to devise a strategy to make the system exit deadlock.

\[
L(u, \mu) = \|u - \hat{u}\|^2_2 + \sum_{k=1}^{M+4} \mu_k (\hat{a}_k^T u - \hat{b}_k) \quad (11)
\]

Let $(u^*, \mu^*)$ be the optimal primal-dual solution to problem eq. (9). The KKT conditions [23] for this problem are
Fig. 2: Positions, Velocities and accelerations of robots falling in deadlock. Video: https://tinyurl.com/y4ylzwh8

1) Stationarity: \( \nabla_u L(u, \mu) \big|_{(u^*, \mu^*)} = 0 \)
\[
\Rightarrow 2(u^* - \hat{u}) = -\sum_{k=1}^{M+4} \mu_k^* \tilde{a}_k^T
\]
\[
\Rightarrow u^* = \hat{u} - \frac{1}{2} \sum_{k=1}^{M+4} \mu_k^* \tilde{a}_k^T.
\] (12)

2) Primal Feasibility
\[
\tilde{a}_k^T u^* \leq \tilde{b}_k \quad \forall k \in \{1, 2, \ldots, M + 4\}
\] (13)

3) Dual Feasibility
\[
\mu_k^* \geq 0 \quad \forall k \in \{1, 2, \ldots, M + 4\}
\] (14)

4) Complementary Slackness
\[
\mu_k^* \cdot (\tilde{a}_k^T u^* - \tilde{b}_k) = 0 \quad \forall k \in \{1, 2, \ldots, M + 4\}
\] (15)

B. KKT Conditions for the deadlock case

Since in deadlock, the acceleration input to the robot i.e. \( u^* \equiv 0 \) is also obtained from QP, KKT conditions can still be used to analyze the characteristics of the system in deadlock. From Def.1, we know that in deadlock, \( u^* = 0 \), \( \hat{u} \neq 0 \) and \( v = 0 \). We conclude that:

1) \( u^* \neq \hat{u} \) i.e. the solution to the QP is not equal to the prescribed PD controller (which means that \( \hat{u} \) is infeasible in deadlock i.e. \( \alpha^T \hat{u} \notin \tilde{b} \)).

2) \( u^* = 0 \) \( \Rightarrow u^* \neq \pm \alpha \). This implies that at least the last four constraints in \( A\hat{u} \leq \tilde{b} \) are inactive i.e.
\[
\{M + 1, M + 2, M + 3, M + 4\} \in \mathcal{I}A(u^*).
\]

Using these observations, we can rewrite the KKT conditions for the deadlock case as follows:

1) Stationarity: \( \nabla_u L(u, \mu) \big|_{(0, \mu^*)} = 0 \)
\[
\Rightarrow \hat{u} = \frac{1}{2} \sum_{k \in \mathcal{A}(u^*)} \mu_k^* \tilde{a}_k^T
\] (20)

2) Primal Feasibility
\[
\tilde{b}_k \geq 0 \quad \forall k \in \{1, 2, \ldots, M + 4\}
\] (21)

3) Dual Feasibility
\[
\mu_k^* \geq 0 \quad \forall k \in \{1, 2, \ldots, M + 4\}
\] (22)

4) Complementary Slackness
\[
\mu_k^* \cdot (\tilde{a}_k^T u^* - \tilde{b}_k) = 0
\]
\[
\Rightarrow \mu_k^* \cdot \tilde{b}_k = 0 \quad \forall j \in \{1, 2, \ldots, M + 4\}
\] (23)

Based on these conditions, we will now motivate a set-theoretic interpretation of deadlock. Assume that the state of ego robot is \( z = (p, v) \) and it has \( M \) neighboring robots denoted as \( Z_{\text{neighbors}} \). Define \( P \in \mathbb{R}^{2 \times 4} \) and \( V \in \mathbb{R}^{2 \times 4} \) appropriately to extract the position and velocity components from \( z \) i.e. \( p = Pz \) and \( v = Vz \). Finally, define \( D \) as:
\[
D(z \mid Z_{\text{neighbors}}) = \{z \in \mathbb{R}^4 \mid u^*(z) = 0, \hat{u}(z) \neq 0, Vz = 0, \mu_k^*(Z) \geq 0 \quad \forall k \in \mathcal{A}(u^*)\}. \] (24)

The set \( D \) is defined as the set of all states of the ego robot which satisfy the criteria of being in deadlock. We have combined the conditions of deadlock into a set theoretic definition. Additionally note that for each robot, its set of deadlock states depends on the states of its neighboring robots. This is because the Lagrange multipliers depend on the states of all robots. To highlight this dependence, we use the notation \( D(z \mid Z_{\text{neighbors}}) \).

The motivation behind stating this definition is to interpret deadlock as a bonafide set in the state space of the ego robot and derive a control strategy that makes the robot evade/exit this set. We now rewrite this definition in more easily interpretable conditions. From eq. (19) and eq. (20), note that
\[
\begin{align*}
  u^*(z) = 0 &\iff \hat{u}(z) = \frac{1}{2} \sum_{k \in \mathcal{A}(u^*(z))} \mu_k^* \tilde{a}_k \\
  u^*(z) = 0 &\iff \hat{u}(z) = -\frac{1}{2} \sum_{k \in \mathcal{A}(u^*(z))} \mu_k^* P(z - z_k)
\end{align*}
\] (25)

Since \( \tilde{a}_k = -\Delta p_{ik} = -P(z - z_k) \), we rewrite eq. (25) as:
\[
\begin{align*}
  u^*(z) = 0 &\iff \hat{u}(z) = -\frac{1}{2} \sum_{k \in \mathcal{A}(u^*(z))} \mu_k^* P(z - z_k)
\end{align*}
\] (26)
We will use this condition to replace the \( u^*(z) = 0 \) criterion in the def. of \( D \) in (24). Secondly, we know that prescribed controller \( \hat{u}(z) \) is a PD controller. Define the goal state as \( z_d = (p_d, 0) \). Noting that \( \hat{u}(z) \neq 0 \) and \( v = 0 \) in deadlock:

\[
\hat{u}(z) = -k_p(p - p_d) - k_vv \\
\iff -k_pP(z - z_d) \neq 0 \iff P(z - z_d) \neq 0 \tag{27}
\]

This criterion is merely restating that in deadlock the ego robot is not at its goal. Finally, we note that the third condition is that the velocity of the ego robot is zero i.e. \( v = 0 \iff Vz = 0 \). Combining these conditions, we rewrite the definition of the deadlock from eq. (24) as follows:

\[
D(z \mid Z_{neighbors}) = \{ z \in \mathbb{R}^4 \mid P(z - z_d) \neq 0, Vz = 0 \}
\]

\[
\hat{u}(z) = -\frac{1}{2} \sum_{k \in A} \mu_k^*P(z - z_k),
\]

\[
\mu_k^* \geq 0 \quad \forall k \in A(u^*) \tag{28}
\]

The Lagrange multipliers \( \mu_k^* \) in general, are a nonlinear function of the state. The exact values of \( \mu_k^* \) can only be determined on the basis of the set of active constraints. If there are \( M \) neighboring robots i.e. \( M \) collision avoidance constraints per ego robot, there will be \( 2^M \) possible cases of active/inactive constraints. Therefore, for further analysis, we will restrict the discussion to the special case of two robots.

V. ANALYSIS FOR TWO ROBOTS

In section IV, we proposed a set-theoretic definition of deadlock for a specific robot in an \( N \) robot system. In this section, we will refine the KKT conditions derived in Sec. IV-B for the case of two robots in the system. We restrict to two robots not only because this case is easier to analyze, but also because it reveals several important underlying characteristics of the system in deadlock that will extend to the \( N \) robot case. Specifically, we believe that deadlock analysis for \( N \) robots can be decomposed into subproblems involving deadlock analysis involving pairs of robots. We will explore this future work. One key feature of a two-robot system is that a single robot by itself cannot be in deadlock i.e. either both robots are in deadlock simultaneously or neither. This is because the sole avoidance collision constraint is symmetric in a system with just two robots. Keeping this in mind, we define system deadlock set to be the set of states for which both robots are in deadlock. To define this set, we modify eq. (25); specifically for this case there is no summation since the ego robot avoids collision only with the other robot i.e.

\[
u^*(z) = 0 \iff \hat{u}(z) = \frac{1}{2} \mu^*a \tag{29}
\]

Assuming \((z_1, z_2)\) are the states of the two robots, we modify eq. (28) and define \( D_{system} \) as:

\[
D_{system} = \{ (z_1, z_2) \in \mathbb{R}^8 \mid \hat{u}_1 = \frac{1}{2} \mu_1^*a_1, \hat{u}_2 = \frac{1}{2} \mu_2^*a_2, (P(z_1 - z_d_1), P(z_2 - z_d_2)) \neq (0, 0), \mu_1^* \geq 0, \mu_2^* \geq 0, (Vz_1, Vz_2) = (0, 0) \}. \tag{30}
\]

where \( a_1 = -a_2 = -(p_1 - p_2) \). To gain more insight into this system deadlock, we derive analytical expressions for the Lagrange multipliers \( \mu_1^*, \mu_2^* \). Depending on whether the collision avoidance constraint is active/inactive at the optimum, there are two cases:

Case 1: The constraint \( a^T u \leq \hat{b} \) is active at \( u = u^* \).

\[
\implies a^T u^* = \hat{b}
\]

\[
\implies a^T (\hat{u} - \frac{1}{2} \mu^* a) = \hat{b}
\]

\[
\implies \mu^* = 2 a^T \hat{u} - \hat{b} \|a\|^2/2 \tag{31}
\]

Case 2: The constraint \( a^T u \leq \hat{b} \) is inactive at \( u = u^* \). i.e. \( a^T u^* < \hat{b} \). From complementary slackness, it follows that \( i.e. \mu^* = 0 \) and hence \( u^* = \hat{u} \). However, this contradicts the def. of deadlock. Hence, we conclude that case 2 can never arise when the two robots are in deadlock.

A. Mechanics interpretation of KKT conditions

Recall eq. (29) which is one of the conditions for two robots in deadlock. The left hand side of this equation is \( \hat{u}(z) = -k_p(p_{ego} - p_d) \). The right hand side of this equation is \( \frac{1}{2} \mu^* a = -\frac{1}{2} \mu^* (p_{ego} - p_{neighbor}) \). Writing eq. (29) (another way) we get \(-k_p(p_{ego} - p_d) + \frac{1}{2} \mu^* (p_{ego} - p_{neighbor}) = 0 \). We interpret the first term as an attractive force that pulls the ego robot towards the goal \( p_{ego} \). Since \( \mu^* > 0 \), we interpret the second term as a repulsive force pushing the ego robot away from its neighbor. Thus, deadlock occurs when the net force due to attraction to goal and repulsion from the other robot vanishes. We illustrate this equilibrium in Fig. 3.

B. Characteristics of two-robot deadlock

We are now interested in analyzing qualitative properties of the system deadlock set with an eye towards synthesizing controllers that enable the robots to get out of deadlock. We will show that when deadlock occurs, (1) the two robots are separated by the safety distance, (2) deadlock set is non-empty and (3) deadlock set is bounded and of measure zero.

Theorem 1. In deadlock, the two robots are separated by the safety distance and on verge of violating safety (Fig. 1, 3)

Proof. In eq. (31), we proved that the collision avoidance constraint is active in deadlock. Since both robots are in deadlock, we know that both of their constraints are active i.e. \( a_1^TU_1 = \hat{b}_1, a_2^T \bar{u}_2 = \hat{b}_2 \) and \( u_1^* = 0 \) and \( u_2^* = 0 \).

This implies \( b_1 = b_2 = 0 \). Using eqs. (6) and (7) and that in deadlock, \( (v_1, v_2) = (0, 0) \) we get

\[
\hat{b}_1 = \frac{a_1}{\alpha_1 + \alpha_2} \|\Delta p_{12}\| h_{12} = 0 \implies h_{12} = 0 \tag{32}
\]

Recall \( h_{12} \) from eq. (2) and using that \( (v_1, v_2) = (0, 0) \), we get

\[
h_{12}(z_1, z_2) = \sqrt{2(\alpha_1 + \alpha_2)} (\|\Delta p_{12}\| - D_s) \tag{33}
\]

Therefore, \( h_{12} = 0 \iff \|\Delta p_{12}\| = D_s \). Since robots remain safe (assuming QP is feasible), we disregard \( \|\Delta p_{12}\| = \)
0. Therefore, \( \|\Delta p_{12}\| = D_s \). Additionally, recalling the definition from \( \partial C \) from eq. (3) we deduce that, in deadlock, \((z_1, z_2) \in \partial C \) i.e. \( D_{\text{system}} \subset \partial C \) ■

This result confirms our intuition, because if the robots are separated by more than the safety distance, then they will have space to move towards each other, albeit with very small non-zero velocity, which however would contradict the definition of deadlock.

We now propose a family of states that are always in the deadlock set \( D_{\text{system}} \) eq. (30).

**Theorem 2.** \( \forall k_p, k_v, D_s > 0, \exists a \) family of states \( z_1^* \) and \( z_2^* \) such that \((z_1^*, z_2^*) \in D_{\text{system}} \). Moreover, these states are such that the robots and their goals are all collinear.

**Proof.** To prove this theorem, we first propose a set of candidate states \((z_1^*, z_2^*)\) and show that they satisfy the definition of deadlock eq. (30). See Fig. 3 for an illustration of quantities referred to in this proof. Let \( p_1^* = \alpha p_d_1 + (1 - \alpha)p_d_2 \) and \( p_2^* = p_1^* - D_s \hat{e}_\beta \) where \( \beta = \tan^{-1}\left(\frac{y_{d_2} - y_{d_1}}{x_{d_2} - x_{d_1}}\right) \) and \( \alpha \in (0, 1) \). Note that \( p_1^*, p_2^*, p_{d_1}, p_{d_2} \) are collinear by construction. Let \( z_1^* = (p_1^*, 0) \) and \( z_2^* = (p_2^*, 0) \). Then we will show that \( z^* = (z_1^*, z_2^*) \in D_{\text{system}} \). Note that

\[
\begin{align*}
\alpha_1 &= -(p_1^* - p_2^*) = -D_s \hat{e}_\beta \\
\alpha_1 &= -k_p(p_1^* - p_{d_1})
\end{align*}
\]

From definition, \( \hat{e}_\beta = \frac{1}{D_G}(x_{d_2} - x_{d_1}, y_{d_2} - y_{d_1}) \) where \( D_G = \|p_{d_2} - p_{d_1}\| = \sqrt{(x_{d_2} - x_{d_1})^2 + (y_{d_2} - y_{d_1})^2} \) is the distance between the goals. Therefore, we have

\[
\begin{align*}
p_1^* - p_{d_1} &= -(1 - \alpha)p_{d_1} + (1 - \alpha)p_{d_2} \\
&= (1 - \alpha)D_G \hat{e}_\beta
\end{align*}
\]

Substituting eq. (35) in eq. (34) gives

\[
\hat{u}_1 = -k_p(1 - \alpha)D_G \hat{e}_\beta
\]

From eq. (34) and eq. (36), we deduce that Lagrange multiplier \( \mu_1 \)

\[
\mu_1 = 2 a_1^T \alpha_1 \left\| \alpha_1 \right\|^2 = 2k_p(1 - \alpha)D_G D_s \hat{e}_\beta > 0 \ \forall \alpha \in (0, 1)
\]

\[
\Rightarrow \frac{1}{2} \mu_1 a_1 = -\frac{1}{2} k_p(1 - \alpha)D_G D_s \hat{e}_\beta = \hat{u}_1
\]

Hence, in eq. (37), we have shown that \( \hat{u}_1 = \frac{1}{2} k_1 \hat{a}_1 \) which is one condition in the definition of deadlock set. Similarly, we can show that \( \hat{u}_2 = \frac{1}{2} k_2 \hat{a}_2 \). Also note that in eq. (37) we have shown that the Lagrange multiplier \( \mu_1 \) is positive, which is another condition in eq. (30). We can similarly show that \( \mu_2 > 0 \). Finally, note that for our choice of states, \( v_1^* = v_2^* = 0 \) and we have restricted \( \alpha \in (0, 1) \) so we ensure that \( p_1^* \neq p_{d_1} \). Hence, the proposed states are always in deadlock. ■

Although in this theorem, we proposed a candidate set of states and verified that they are in deadlock, one can check by explicitly solving the equations in eq. (30) that these are in fact, the only states that satisfy the criteria of deadlock.

**Theorem 3.** \( D_{\text{system}} \) is bounded and of measure zero.

**Proof.** Following the definition of \( p_1^* \) and \( p_2^* \) from theorem 1 and 2, we can show that when two robots are in deadlock, their positions satisfy the following constraint:

\[
\| (p_1 - p_{d_1}) \| + \| (p_2 - p_{d_2}) \| = D_s + D_G
\]

This can be verified by straightforward substitution. From this constraint it is evident, that the deadlock set is not “large” and of measure zero. That is why, random perturbations are one feasible way to resolve deadlock. However, whether or not deadlock gets resolved depends on the stability of this set. We explore resolution of deadlock in the next section. ■

**Fig. 4:** Three Phase Controller to Exit Deadline Set.

**VI. DEADLOCK RESOLUTION**

We are now interested in using the properties derived above to synthesize a corrective strategy that (a) gets the robots out of deadlock, (b) ensures safety and (c) makes the robots converge to their respective goals. One approach to accomplish these three objectives is to detect the incidence of deadlock while the QP based controller is running on the robots. Once detected, any small non-zero perturbation to control will instantaneously give a non-zero velocity to the robots. Thereafter, the QP based controller can take charge again and we can hope that using this controller the system state will come out of deadlock at-least for a short time. This approach has two limitations however, firstly, since it was the QP based controller that led the system to deadlock in the first place, there is no guarantee that the system will not fall in deadlock again. Secondly, perturbations can violate safety.

We propose a provably-correct controller which ensures that goal stabilization, avoidance control and deadlock resolution are satisfied. Refer to Fig. 4 for a schematic of our approach. This algorithm is described here:

1) Given initial position of the robots, the algorithm starts by executing controls derived from CBFs based QP in Phase 1 of deadlock resolution.
1. This ensures movement of robots to the goals and safety by construction. To detect the presence of deadlock, we continuously compare \( \| u^* \|, \| v \|, \| p - p_o \| \) against small thresholds. If satisfied, we switch control to phase 2, otherwise, phase 1 continues to operate.

2) In this phase, we use any controller to ensure that the (1) robots rotate around each other to swap positions while (2) maintaining the safe distance. Feedback linearization (FL) is one viable approach that can generate such a controller, and moreover, FL based control can be easily decentralized. By ensuring \( \| \Delta p_{12} \| = D_s \) and rotation \( (\Delta p_{12}^T \Delta v_{12} = 0) \), we ensure that safety is maintained \( (i.e \ h_{12} \geq 0) \). Once the robots swap their positions (which is guaranteed by design), their new positions will ensure that prescribed PD controllers will be feasible in the future.

3) After convergence of Phase 2 (which happens in finite time), control switches to Phase 3, which simply uses the prescribed PD controllers. This phase guarantees that the distance between robots is non-decreasing and safety is maintained as we prove in theorem 4.

Fig. 5 shows simulation and experimental results from running this strategy on a pair of double-integrator (Fig. 5(a)) and nonholonomic robots (Fig. 5(b),5(c)). Videos of simulations and experiments are available at https://tinyurl.com/y4ylzwh8. Note that for nonholonomic robots, we noticed from simulations and experiments that deadlock only occurs if the body frames of both robots are perfectly aligned with one another at \( t = 0 \). Since this alignment is difficult to establish in experiments, we simulated a virtual deadlock at \( t = 0 \) \( i.e. \) assumed that the initial position of robots are ones that are in deadlock. We next prove the correctness of our three-phase strategy:

**Theorem 4.** Assuming that PD controllers are overdamped and \( D_G > D_s \), this strategy ensures that (1) the robots will never fall in deadlock and (2) converge to their goals.

**Proof.** We would like to show that once phase three control begins, the robots will never fall back in deadlock. We will do this by showing that the distance between the robots is non-decreasing, once phase three control starts. We break this proof into three parts consistent with the three phases:

**Phase 1 \( \rightarrow \) Phase 2:**
Let \( t = t_1 \) be the time at which phase 1 ends (and phase 2 starts) \( i.e. \) when robots fall in deadlock. In theorem 1 we showed that in deadlock \( \| \Delta p_{21} \| = D_s \), and in theorem 2 we showed that the positions of the two robots and their goals are collinear. So at the end of phase 1, \( \Delta p_{21}(t_1) = D_s \hat{\beta}_{\beta+\pi} \). The goal vector \( \Delta p_{d_{21}}(t) := p_{d_2} - p_{d_1} = D_G \hat{\beta} \forall t > 0 \). Moreover, since the robots are static in deadlock, \( \Delta v_{21}(t_1) = 0 \)

**Phase 2 \( \rightarrow \) Phase 3:**
The initial condition of phase two is the final condition of phase one \( i.e. \) \( \Delta p_{21}(t_1) = D_s \hat{\beta}_{\beta+\pi} \) and \( \Delta v_{21}(t_1) = 0 \). In phase two, we rotate the assembly of the robots making sure that the distance between them stays at the safety distance, until the orientation of the vector \( \Delta p_{21}(t) = D_s \hat{\beta}_{\theta(t)} \) aligns with the orientation of \( \Delta p_{d_{21}} = D_G \hat{\beta} \). Assuming that the control strategy accomplishes this, \( i.e. \theta \rightarrow \beta, \exists \) a time \( t_2 \) at which \( \theta(t_2) = \beta \). Additionally, at this time, the robots are no longer moving, hence their velocities are zero. So at this time, \( \Delta p_{21}(t_2) = D_s \hat{\beta} \) and \( \Delta v_{21}(t) = 0 \). These states mark the final condition for phase 2 and initial for phase 3.

**Phase 3 \( \rightarrow \infty \):**
In this phase, the initial conditions are \( \Delta p_{21}(t_2) = D_s \hat{\beta} \) and \( \Delta v_{21}(t_2) = 0 \). Also, note that the dynamics of phase 3 control are specified by the prescribed PD controllers. The dynamics of relative positions and velocities are:

\[
\Delta p_{21} = \Delta v_{21} \quad \Delta \hat{v}_{21} = -k_p (\Delta p_{21} - \Delta p_{d_{21}}) - k_v \Delta v_{21},
\]

(38)

where \( \Delta p_{d_{21}} = D_G \hat{\beta} \).

Now, we will do a coordinate change as described next. Let \( \Delta p_{21} := R_{-\beta} \Delta p_{21} \) and \( \Delta v_{21} := R_{-\beta} \Delta v_{21} \). The initial conditions in these coordinates are \( \Delta p_{21}(t_2) = (D_s, 0) \) and \( \Delta v_{21}(t_2) = 0 \) \( i.e. \) \( \Delta \hat{p}_{21}(t_2) = D_s, \Delta \hat{p}_{d_{21}}(t_2) = 0, \Delta \hat{v}_{21}(t_2) = 0 \) and \( \Delta \hat{v}_{d_{21}}(t_2) = 0 \). The dynamics in new
coordinates are:
\[
\Delta \dot{p}_{21} = \Delta \dot{x}_{21}
\]
\[
\Delta \dot{x}_{21} = -k_p(\Delta \dot{p}_{21} - R_{\beta} \Delta p_{d_{21}}) - k_v \Delta \dot{x}_{21}.
\]
Using these coordinates, note that \(R_{\beta} \Delta p_{d_{21}} = (D_G, 0)\). Note from the dynamics and the initial conditions for the \(y\) components of relative position and velocities that the only solution is the zero solution i.e. \(\Delta \dot{p}_{21}(t) \equiv 0 \) and \(\Delta \dot{x}_{21}(t) \equiv 0 \) \(\forall t \geq t_2\). As for the \(x\) component, we can compute the solution to be
\[
\Delta \dot{p}_{21}(t) = c_1 e^{\omega_1 (t-t_2)} + c_2 e^{\omega_2 (t-t_2)} + D_G
\]
and \(\Delta \dot{x}_{21}(t) = c_1 \omega_1 e^{\omega_1 (t-t_2)} + c_2 \omega_2 e^{\omega_2 (t-t_2)}\) where
\[
\omega_{1,2} = \frac{1}{2} \left( -k_v \pm \sqrt{k_v^2 - 4k_p} \right), \quad c_1 = \frac{\omega_1 (D_G - D_x)}{\omega_1 - \omega_2}, \quad c_2 = -\frac{\omega_1 (D_G - D_x)}{\omega_1 - \omega_2},
\]
and \(\omega_1 - \omega_2 = -\sqrt{k_v^2 - 4k_p}\). After substituting these values, we get,
\[
\Delta \dot{p}_{21}(t) = \frac{k_p(D_x - D_G)(e^{\omega_1 (t-t_2)} - e^{\omega_2 (t-t_2)})}{\sqrt{k_v^2 - 4k_p}}.
\]
Now, from the assumptions that PD controllers are overdamped i.e \(k_v, k_p^2 - 4k_p > 0\) and \(D_G > D_x\), it follows that \(\Delta \dot{p}_{21}(t) = \frac{k_p(D_x - D_G)(e^{\omega_1 (t-t_2)} - e^{\omega_2 (t-t_2)})}{\sqrt{k_v^2 - 4k_p}} > 0\) and \(\Delta \dot{x}_{21}(t) = \frac{D_G - D_x}{\sqrt{k_v^2 - 4k_p}}(\omega_1 e^{\omega_1 (t-t_2)} - \omega_2 e^{\omega_2 (t-t_2)}) + D_G \geq D_x > 0\).

Finally, note that \(\frac{d}{dt} |\Delta p_{21}(t)| = \Delta \dot{p}_{21}(t) \Delta \dot{x}_{21}(t) \geq 0\). Hence, the distance between the robots is non-decreasing i.e. the robots never fall in deadlock and safety is guaranteed. Additionally, since the robots use a PD-type controller, their positions exponentially stabilize to their goals.

VII. CONCLUSIONS

In this paper, we analyzed the characteristic properties of deadlock that results from using CBF based QPs for avoidance control. We demonstrated how to interpret deadlock as a subset of the state space and proved that in deadlock, the robots are on the verge of violating safety. Additionally, we showed that this set non-empty and bounded. We developed a three-phase control algorithm to force the robots out of deadlock and ensure task completion while staying safe. There are several directions we would like to explore in future. Firstly, our analysis for deadlock was restricted to the case of two robots. We found that the exponential increase in the possible values of Lagrange multipliers limit the scalability of deriving characteristics of deadlock. However, we still claim that the problem of \(N\) robot deadlock can be decomposed into subproblems of pairwise deadlock in two-robot. We will pursue this approach and extend our analysis to the \(N\) robot case. Secondly, we would like to compute the basin of attraction of the deadlock set i.e. the formally characterize the set of initial conditions that lead the robots to deadlock. We envision that tools from backwards reachability set calculation can be used to formally characterize the basin of attraction. Finally, we focused on CBF based QPs for analysis. In future, we would like to extend this to other reactive methods such as velocity obstacles and tools using value functions, and explore the properties that make a particular algorithm immune to deadlock.

VIII. ACKNOWLEDGEMENTS

Authors would like to thank Michael Cheng and Fan Jia for help with experiments. This research was supported by the DARPA Cooperative Agreement HR00111820051.

REFERENCES

[1] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, “Distributed search and rescue with robot and sensor teams,” in *Field and Service Robotics*, pp. 529–538, Springer.
[2] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
[3] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 376–386, 2005.
[4] P. Ogren, M. Egerstedt, and X. Hu, “A control lyapunov function approach to multiagent coordination,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 847–851, 2002.
[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
[6] U. Bormann, L. Wang, A. D. Ames, and M. Egerstedt, “Control barrier certificates for safe swarm behavior,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.
[7] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
[8] S. Petti and T. Fraichard, “Safe motion planning in dynamic environments,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2210–2215, IEEE, 2005.
[9] P. O’Donnell, “Deadlockfree and collision-free coordination of two robot manipulators,” in *Proc. IEEE International Conference on Robotics and Automation*, pp. 484–489, 1989.
[10] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
[11] G. Leitmann and J. Skowronski, “Avoidance control,” *Journal of optimization theory and applications*, vol. 23, no. 4, pp. 581–591, 1977.
[12] G. Leitmann and J. Skowronski, “A note on avoidance control,” *Optimal Control Applications and Methods*, vol. 4, no. 4, pp. 335–342, 1983.
[13] D. M. Stipanović, P. F. Hokayem, M. W. Spong, and D. D. Šiljak, “Cooperative avoidance control for multiagent systems,” *Journal of dynamic systems, measurement, and control*, vol. 129, no. 5, pp. 699–707, 2007.
[14] P. F. Hokayem, D. M. Stipanović, and M. W. Spong, “Coordination and collision avoidance for lagrangian systems with disturbances,” *Applied Mathematics and Computation*, vol. 217, no. 3, pp. 1085–1094, 2010.
[15] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
[16] J. Van den Berg, M. Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” in *2008 IEEE International Conference on Robotics and Automation*, pp. 1928–1935, IEEE, 2008.
[17] J. Van den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in *Robotics research*, pp. 3–19, Springer, 2011.
[18] D. Wilkie, J. Van Den Berg, and D. Manocha, “Generalized velocity obstacles,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5573–5578, IEEE, 2009.
[19] H. Yamaguchi, “A cooperative hunting behavior by mobile-robot troops,” in *the International Journal of Robotics Research*, vol. 18, no. 9, pp. 931–940, 1999.
[20] M. Jager and B. Nebel, “Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots,” in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No. 01CH37180)*, vol. 3, pp. 1213–1219, IEEE.
[21] Y. Li, K. Gupta, and S. Payandeh, “Motion planning of multiple agents in virtual environments using coordination graphs,” in Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 378–383, IEEE, 2005.

[22] E. J. Rodríguez-Seda, D. M. Stipanović, and M. W. Spong, “Guaranteed collision avoidance for autonomous systems with acceleration constraints and sensing uncertainties,” Journal of Optimization Theory and Applications, vol. 168, no. 3, pp. 1014–1038, 2016.

[23] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.