Multiagent Systems With CBF-Based Controllers: Collision Avoidance and Liveness From Instability

Mrdjan Jankovic, Fellow, IEEE, Mario Santillo, Senior Member, IEEE, and Yan Wang, Member, IEEE

Abstract—Assuring system stability is typically a major control design objective. In this brief, we present a system where instability provides a crucial benefit. We consider multiagent collision avoidance using control barrier functions (CBFs) and study trade-offs between safety and liveness—the ability to reach a destination without large detours or gridlocks. We consider centralized, full information policy as the benchmark, two standard decentralized policies with only the local (host) control available, and the predictor-corrector for collision avoidance (PCCA) in which each agent optimizes for everyone using incomplete information, and corrects for discrepancies. One contribution of this brief is proving feasibility for the centralized and PCCA policies. Monte Carlo simulations show that decentralized, host-only control policies lack liveness while the PCCA policy performs as well as the centralized avoiding gridlocks. We explain the observed results by considering two agents negotiating passing order through an intersection. We show that the structure and stability of the resulting equilibria correlates with the observed propensity to gridlock—the policies with unstable equilibria avoid gridlock while those with stable ones do not.

Index Terms—Control barrier functions (CBFs), gridlock, multiagent control.

I. INTRODUCTION

With the recent advances in automated driver assisted systems, autonomous vehicles, and multirobotic systems, management of agent-to-agent interactions has received a lot of attention. Each agent must be capable of planning and executing paths in real time while assuring collision-free operation. A challenge arises from operating scenarios with multibrand, multirobot factories, or heterogeneous driving situations with fully autonomous, semi autonomous, and human-driven vehicles that both compete and cooperate. This can lead to complex feedback loops that are only partially controllable from each agent’s perspective, presenting an opportunity for the rigors of feedback control.

In recent years, control barrier functions (CBFs) [1], [2], [17] have shown great promise in providing a computationally efficient method that is both provably safe and able to handle complex scenarios. Similar to model-predictive control (MPC), CBF is a model-based control design method that can be formulated as a quadratic program (QP) and solved online using real-time capable solvers, for example, [8]. A noteworthy difference is that CBF-based optimization remains convex even for nonlinear (input affine) system dynamics and nonconvex obstacles. Many other methods for multiagent systems, such as interactive Gaussian process (IGP) [15] and optimal reciprocal collision avoidance (ORCA) [13], are nonconvex.

With all agents having the ability to communicate, a centralized controller—an off-board computer that takes in all agents’ inputs (such as their desired acceleration), calculates, and relays the optimal action for each agent—may be employed [18]. One contribution of this brief is a proof that the centralized, distance-CBF-based QP problem is always feasible. In less controlled scenarios, such as vehicles operating on roadways, a decentralized controller may be necessary where each agent computes and executes the best control for itself given incomplete information. The decentralized follower (DF) method [4] assumes that each agent takes full responsibility for collision avoidance while the decentralized reciprocal (DR) method [16] assigns each agent a fraction of responsibility. Robust CBF (RCBF) [10] was used as the basis for the development of the predictor–corrector for collision avoidance (PCCA) algorithm [14]. The PCCA performs “co-optimization”—compute the best course of action for every agent with local, incomplete information—and corrects the result by feeding back the error between the actual and locally computed actions for other agents.

After introducing each algorithm, we assess feasibility and its effect to online algorithm implementation. We then proceed to compare algorithm metrics on liveness, collisions, and feasibility through randomized five-agent Monte Carlo simulation trials for each method (each set up identically). It turns out that the two decentralized algorithms exhibit gridlocks as well as generally slower arrival to the destination compared to the centralized and PCCA policies, which show consistently fast convergence with no gridlocks.

Based on the observed behavior, we conjectured that the equilibrium structures and their stability play the main role in determining propensity to gridlock. Gridlocks for multiagent systems have been studied before in robotics literature. The results typically deal with decentralized policies (DF, DR like) and propose methods to deconflict the agents once a gridlock is detected—see, for example [6], [9], [16] and references therein. Here, we are interested in trying to explain the observed differences between control policies. We analyze the equilibrium structure in the joint space for a simple problem of two agents negotiating passing order through an intersection or a merge point. The decentralized policies have a stable set of equilibria, which explains their propensity to gridlock.
In contrast, the Centralized and PCCA policies have unstable equilibria and, thus, only initial conditions from a set of measure zero end up in a gridlock. Moreover, the instability is exponential, which suggests fast movement away from the gridlock point even for trajectories that start fairly close to the stable manifold of the equilibrium set.

The paper is an extension of the conference paper [11]. Besides many improvements and clarifications throughout the text, the main addition is Section VI that contains equilibrium structure analysis for the different control policies and explains the role instability plays in avoiding gridlocks. Due to length limits, the gridlocking, co-optimization-based, complete constraint set (CCS) policy included in [11] has been dropped.

The brief is organized as follows. Section II reviews (R)CBF-based control. Section III introduces the dynamic constraint set (CCS) policy included in [11] has been dropped. Section IV introduces CBF-based controllers for collision avoidance. The simulations in Section V consider randomized trials of five interacting agents. The equilibrium analysis is provided in Section VI.

**Notation:** For a differentiable function \( h(x) \) and a vector \( f(x) \), \( L_f h(x) \) denotes \( \partial h/\partial x \). A continuous function \( \alpha(\cdot) \) is of class \( K \) if it is strictly increasing and satisfies \( \alpha(0) = 0 \). We additionally assume \( \alpha \in K \) is Lipschitz continuous. A function \( \gamma(t, \varepsilon) \) is said to be \( O(\varepsilon) \) if \( |\gamma(t, \varepsilon)| \leq \kappa|\varepsilon| \) for some \( \kappa > 0 \) and for all sufficiently small \( \varepsilon \).

**II. RCBFs Reviewed**

In this section, we briefly review the concepts of CBFs – introduced in [17] and later combined with QPs (see for example, [1], [2]) – and of RCBFs introduced in [10]. CBFs apply to nonlinear systems affine in the control input

\[
\dot{x} = f(x) + g(x)u
\]

with \( x \in \mathbb{R}^n \), \( u \in \mathbb{R}^m \), \( f(x) \), and \( g(x) \) Lipschitz continuous. RCBFs extend CBFs to systems with a bounded external disturbance \( w(t) \), \( \|w(t)\| \leq \tilde{w} > 0 \), of the form

\[
\dot{x} = f(x) + g(x)u + p(x)w
\]

where \( p(x) \) is also Lipschitz continuous.

One control objective is to regulate the system to the origin or suppress the disturbance [i.e., achieve input-to-state stability (ISS)] and we assume that there is a known baseline control \( u_0 \in \mathbb{R}^m \) that achieves the objective. The other control objective is to keep the state of the system in an admissible (i.e., safe) set defined by \( C = \{ x \in \mathbb{R}^n : h(x) \geq 0 \} \) where \( h(x) \) is a sufficiently smooth function. Here we combine definitions of CBF and RCBF into one.

**Definition 1 (CBF and RCBF):** A function \( h(x) \) is a CBF for the system (1) if there exists a function \( \alpha_K \in \mathbb{R}^+ \) such that

\[
L_x h(x) = 0 \Rightarrow L_f h(x) + \alpha_K(h(x)) > 0. \tag{3}
\]

The function \( h(x) \) is an RCBF for the system (2) if

\[
L_x h(x) = 0 \Rightarrow L_f h(x) - \|L_p h\| \tilde{w} + \alpha_b(h(x)) > 0. \tag{4}
\]

In the CBF case, the definition asks that, when the control over the evolution of \( \dot{h} = L_f h + L_x h u \) is lost \( (L_f h = 0) \), the rate of decrease of \( h \) to 0 is not faster than \( \alpha_b(h) \).

Similarly, for the system with disturbance, the bound on the rate of decrease applies for the worst case disturbance.

One advantage of CBFs for control affine systems is that they naturally lead to linear constraints on the control input \( u \) that could be enforced online. A QP is set up to enforce the constraint, while staying as close as possible to the baseline (performance) control input \( u_0 \).

(R)CBF QP Problem: Find the control \( u \) that satisfies

\[
\min_u \|u - u_0\|^2 \quad \text{s.t.} \quad F_i u \geq 0, \quad i = 0, 1, \text{ or } 2 \tag{5}
\]

where we select \( F_0 = L_f h + L_x h u + \alpha_b(h(x)) \) if \( h \) is a CBF for the system (1); \( F_1 = L_f h - \|L_p h\| \tilde{w} + L_x h u + \alpha_b(h(x)) \) if \( h \) is an RCBF for the system (2) with an unknown disturbance; or \( F_2 = L_f h + L_p h \tilde{w} + L_x h u + \alpha_b(h(x)) \) when an estimate/measurement \( \tilde{w} \) of the disturbance \( w \) is available.

The available results (e.g., [1], [2], [10]) guarantee that the resulting control is Lipschitz continuous, the barrier constraint \( F_i \) is satisfied, which implies that \( h(x(t)) \geq 0 \) for all \( t \) and the safe set \( C \) is forward invariant. Note that strict “>” is needed in the (R)CBF definition (3) and (4) to guarantee Lipschitz continuity of the control law [10], or, alternatively, \( L_g h \neq 0 \), for all \( x \in C \) needs to be assumed as in [1].

In the distance-based barrier functions considered in the rest of the brief, the control input does not appear in their first derivative as assumed in the definition of (R)CBF. The inputs appear in the second derivative of \( h \), so we follow the ideas of [12] for dealing with higher relative degree: instead of enforcing \( \dot{h} + \alpha_b(h) \geq 0 \), we switch to linear barrier dynamics and enforce

\[
\dot{h} + l_1 \tilde{h} + l_0 h \geq 0 \tag{6}
\]

as the QP constraint. The parameters \( l_0, l_1 \) should be selected so that the two roots \([\lambda_1, \lambda_2]\) of the polynomial \( s^2 + l_1 s + l_0 = 0 \) are negative real \( (\lambda_1, \lambda_2 = (l_1 \pm \sqrt{l_1^2 - 4l_0})/2)/2 \).

It is a matter of straightforward calculation to show that, if the barrier constraint (6) holds, the set \( C^* = \{(x) : h(x) \geq 0, h(x) \geq -(1/\lambda_1)\tilde{h}(x)\} \), where \( \lambda_1 \) is either of the two roots, is forward invariant. With \( C^* \subset C \), the original constraint \( h(x) \geq 0 \) will be satisfied and \( C^* \to C \) as \( \lambda_1 \to \infty \).

For the second-order barrier, the QP constraints that need to be enforced for \( C^* \) to be forward invariant are

\[
F_0 \leftarrow L_f^2 h + L_x L_f h u + l_1 L_f h + l_0 h \geq 0 \tag{7}
\]

in the case of the CBF for the system without disturbance

\[
F_1 = L_f^2 h - \|L_p L_f h\| \tilde{w} + L_x L_f h u + l_1 L_f h + l_0 h \geq 0 \tag{8}
\]

for an RCBF with unknown disturbance bounded by \( \tilde{w} \); or

\[
F_2 = L_f^2 h + L_p L_f h \tilde{w} + L_x L_f h u + l_1 L_f h + l_0 h \geq 0 \tag{9}
\]

for an RCBF with disturbance estimate \( \tilde{w} \).

**III. HOLONOMIC AGENT MODEL**

Agents are typically modeled either with holonomic double integrators in the xy plane or as a nonholonomic “unicycle” or “bicycle” model. For simplicity, we consider the first option
with an agent \(i\) modeled as a disk of radius \(r_0\) with the center motion given by the double integrator in each dimension

\[
\dot{x}_i = v_{xi}, \quad \dot{y}_i = v_{yi} \\
\ddot{x}_i = u_{xi}, \quad \ddot{y}_i = u_{yi}.
\]  

(10)

The relative motion between two agents \(i\) and \(j\) is given by the following equation:

\[
\dot{\xi}_{ij} = v_{ij} \\
\dot{u}_{ij} = u_i - u_j
\]

(11)

where \(\xi_{ij} = [x_i - x_j, y_i - y_j]^T\) is the center-to-center displacement between the two agents, \(v_{ij} = [v_{xi} - v_{xj}, v_{yi} - v_{yj}]^T\) is their relative velocity, and \(u_i = [x_{xi}, u_{yi}]^T\) is agent \(i\)'s control. Our goal is to keep \(||\xi_{ij}||\) larger than \(r \geq 2r_0\) (the distance \(r\) strictly greater than \(2r_0\) provides a "radius margin," see [19]). To this end, we define a distance-based CBF

\[
h(\xi_{ij}) = \xi_{ij}^T \xi_{ij} - r^2.
\]

(12)

The advantages of this CBF over alternatives, such as the position-velocity one used in [4], and [16], is that 1) we can prove feasibility of the Centralized and the PCCA policies and 2) it allows a radius (barrier) margin because the calculation does not collapse when \(h(\xi_{ij}) < 0\). A disadvantage is that \(h\) has relative degree two from all inputs. Because of this, we apply the approach described in Section II and form a CBF constraint

\[
F_{ij} := \dot{h} + l_i \dot{h} + l_0 h = a_{ij} + b_{ij} (u_i - u_j) \geq 0
\]

(13)

where \(a_{ij} = 2v_{ij}^T v_{ij} + 2 l_i \xi_{ij}^T v_{ij} + l_0 (\xi_{ij}^T \xi_{ij} - r^2), b_{ij} = 2\xi_{ij}^T u\). The function \(h\) is a CBF for the system (11) because \(L_d h = 2b_{ij} \neq 0\) unless the two agents completely overlap (that is, are well past the point of collision). As a result, with only two agents, we can always enforce positive invariance of the admissible set \(C^+_0 = \{ (\xi_{ij}, v_{ij}) : h(\xi_{ij}) \geq 0, h(\xi_{ij}) \geq -(1/\lambda) h(\xi_{ij}, v_{ij}) \}\), with \(\lambda\) one of the two roots of \(s^2 + l_1 s + l_0 = 0\), as discussed above. Without loss of generality, we use \(l = 1\).

As we shall see below, with the Centralized controller, we can guarantee that the agents do not collide even if there is no barrier margin: \(r = 2r_0\). Noncentralized controllers (DF, DR, PCCA) might have constraint violations because two agents \(i\) and \(j\) compute \(u_i\) and \(u_j\) independently, based on different information available to them. With \(r > r_0\), the agents have still not collided when \(h < 0\) (this happens if \(h \leq l_0 (r_0^2 - r^2)\)), and there is a repelling force pushing them apart.

IV. CBF-BASED COLLISION AVOIDANCE ALGORITHMS

We assume that each agent has its own preferred control action \(u_{0i}\), computed independently of the collision avoidance algorithm. In this brief, each agent knows its own final destination and uses linear quadratic regulator (LQR) to compute \(u_{0i}\). We first set up the Centralized controller that controls all the agents, knows everyone’s \(u_{0i}\)’s, and uses the barrier constraints between \(N_a\) agents as shown in (13).

Centralized QP: Find the controls \(u_i, i = 1, \ldots, N_a\)

\[
\min_{u_i, u_0} \sum_{i=1}^{N_a} \|u_i - u_{0i}\|^2 \\
\text{s.t.} \ a_{ij} + b_{ij} (u_i - u_j) \geq 0 \ \forall \ i, j = 1, \ldots, N_a, \ i \neq j
\]

(14)

where \(a_{ij}\) and \(b_{ij}\) are defined after (13).

The QP could be solved by a central node and communicated to the agents, or each agent could solve the QP independently, which still requires communication between them. If this QP problem is feasible, and this is proven below, the control action would satisfy all the barrier constraints and guarantee collision-free operation (see [16]).

Without communication, the base control action \(u_{0j}\) for the target (i.e., other) agents are not available to the host \(i\) (the agent doing the computation). In this case, each agent could implement an on-board decentralized controller. One version, included here because it resembles many defensive driving policies considered in the literature, is for each agent to accept full responsibility for avoiding all the other agents. Borrowing nomenclature from the Game Theory, we refer to this policy as “DF.”

DF QP (Agent \(i\)): Find the control \(u_i\) for the agent \(i\) that satisfies

\[
\min_{u_i} \|u_i - u_{0i}\|^2 \\
\text{s.t.} \ a_{ij} + b_{ij} u_i \geq 0 \ \forall \ j = 1, \ldots, N_a, \ j \neq i
\]

(15)

This formulation is essentially the same as in [4] with a different barrier function used. The agent \(i\) has only its own action to avoid all other agents and there are no guarantees that the DF QP is feasible. Even when it is feasible there are no collision avoidance guarantees. The reason is that each agent knows only its own acceleration \(u_{0i}\), and may assess it safe to apply. That is, if \(a_{ij} + b_{ij} u_{0i} \geq 0\), for all \(j\), agent \(i\) would consider \(u_{0i}\) safe to apply. Similarly, agent \(j\) might find that \(u_{0j}\) is safe to apply. However, \(a_{ij} + b_{ij} u_{0i} \geq 0\) and \(a_{ij} - b_{ij} u_{0j} \geq 0\) do not imply \(a_{ij} + b_{ij} (u_{0i} - u_{0j}) \geq 0\), which would actually guarantee collision avoidance. Indeed, even with only two agents, simulations show that there could be a collision as illustrated in [11].

To improve the DF performance, the “DR” policy was introduced in [16].

DR QP (Agent \(i\)): Find the control \(u_i\) for the agent \(i\) that satisfies

\[
\min_{u_i} \|u_i - u_{0i}\|^2 \\
\text{s.t.} \ \frac{1}{2} a_{ij} + b_{ij} u_i \geq 0 \ \forall \ j = 1, \ldots, N_a, \ j \neq i
\]

(16)

The only difference from the DF version is the (1/2) factor multiplying \(a_{ij}\) – that is, each agent assumes half the responsibility for avoiding a collision (we assume all the agents are the same). The method was shown in [16] to guarantee constraint adherence\(^1\) and, hence, collision avoidance as long as it

\(^1\)Stackelberg games are also referred to as “leader–follower.” In the algorithm we consider, every agent assumes the role of a follower.

\(^2\)The constraint in (16) is agent \(i\)’s contribution to satisfying the CBF constraint \(a_{ij} + b_{ij} (u_i - u_j) \geq 0\) needed for collision avoidance.
is feasible. When it is not feasible, [16] proposed a braking action. Braking, however, works only if all agents, even those with feasible QP, apply it simultaneously. To illustrate the issue, consider two moving agents passing the stationary one in the middle as shown in Fig. 1. The DR QP becomes infeasible for the agent in the middle, but, because it was already stationary, braking has no effect. The other two have feasible QPs and keep moving forward expecting, in vain, the middle agent to contribute its part toward collision avoidance. Possible remedy is for each agent to run separate DR-QPs for everyone and brake when any one turns infeasible (unknown \(u_{0,j}\)’s have no impact on feasibility because they do not impact the constraints). Besides increased computational footprint, one has to decide when to stop braking: as soon as all QPs become feasible or after all agents have stopped. In the simulation section, we have allowed the QP solver to resolve the feasibility issue by selecting control with the smallest constraint violation.

Instead of \(N_a\) DR-QPs being solved by the host, one could set up a single QP that includes all the control variables and all the constraints – a quasi-centralized policy with incomplete information. Zeros are used instead of unknown other agents’ \(u_{0,j}\)’s. The RCBF with known disturbance concept (9) is applied, leading to development of the “PCCA” method [14].

PCCA QP (Agent i): Find control actions \(u_{ij}, j = 1, \ldots, N_a\) that satisfy

\[
\min_{u_{i1}, \ldots, u_{in}} \left( \|u_{ii} - u_{0i}\|^2 + \sum_{j=1, j \neq i}^{N_a} \|u_{ij}\|^2 \right) \\
\text{s.t. } a_{ij} + h_{ij}(u_{ij} + \hat{w}_{ij} - u_{ik} - \hat{w}_{ik}) \geq 0 \\
\forall j, k = 1, \ldots, N_a 
\]

(17)

and implement its own: \(u_i = u_{i0i}^*\). The (fictitious) disturbance terms \(\hat{w}_{ij}\) have been added to the \(u_{ij}\). They represent uncertainty of agent \(i\)’s computation of agent \(j\)’s acceleration. Instead of considering the worst case disturbance (8) leading to a conservative control, PCCA uses the known disturbance approach (9) with the disturbance estimate as the difference between the control action for agent \(j\) (\(u_{ij}^*\)) computed by the agent \(i\) and the action agent \(j\) actually implemented (\(u_{ij}\))

\[
\hat{w}_{ij} = u_{ij} - u_{ij}^* 
\]

(18)

Because \(u_{ij}^*\) requires knowing \(\hat{w}_{ij}\) and vice versa, an algebraic loop is created. To break this loop, one could use either the value from the previous sample or a low pass filter.

For two agents, Santillo and Jankovic [14] proved that possible constraint violation is of the order of the sample time \(\Delta T\) – the smaller the sample time, the smaller the error. Even if one agent is not cooperating, the other agent eventually takes over full responsibility for collision avoidance, also reducing constraint violations to the order of \(\Delta T\). Note that PCCA assumes information (measurement) of other agents’ acceleration that can generally be obtained by lead filtering velocity measurements/estimates or from an estimator such as the one proposed in [3]. Using a filter to break the algebraic loop provides the designer another knob to suppress measurement noise. The barrier margin would have to increase with increasing time constant.

We now show that the Centralized and PCCA QPs are feasible. To the best of our knowledge, this is a new result, but specific to the distance-based barrier function. The problem is nontrivial because there are situations where we have more active QP constraints than the linearly independent (row) vectors multiplying control inputs.

**Proposition 1:** The Centralized QP (14) and PCCA QP (17) are always feasible in the admissible set \(C^* = \{ x \in \mathbb{R}^n | h_{ij}(x) \geq 0, h_{ij}(x) \geq -(1/\lambda_1)\dot{h}_{ij}(x), i, j \in \{1, \ldots, N_a\}, i \neq j \}\) and the solution in each case is unique.

**Proof:** Consider the Centralized policy constraint \(F_{ij} := a_{ij} + h_{ij}(u_{i} - u_{j})\). From the definition of \(a_{ij}, h_{ij}\) we have

\[
F_{ij} \geq 2\|v_{ij}\|^2 + 2\xi_{ij} (u_{i} - u_{j}) + 2\lambda_1 v_{ij}^T v_{ij} 
\]

where the last term is obtained by using \(h_{ij} \geq -(1/\lambda_1)\dot{h}_{ij}\) (from the definition of \(C^*\)) and \(l_1 - l_0/\lambda_1 = \lambda_1\). From here, we construct a feasible \(u\) by selecting one that satisfies

\[
u_{i} - u_{j} = -\lambda_1 v_{ij} \quad (19)
\]

which results in \(F_{ij} \geq 2\|v_{ij}\|^2 \geq 0\).

We proceed by using mathematical induction. For the first two agents, pick any \(u_1\) and \(u_2\) that satisfy (19) where, in this case, \(i = 1\) and \(j = 2\). For example, we could select \(u_1 = 0, u_2 = \lambda_1 v_{12}\). Proceeding with the induction argument, assume that for the first \(l - 1\) agents we have selected control inputs \(u_1, \ldots, u_{l-1}\) such that the condition (19) holds for all \(i, j, l, l - 1, i \neq j\). Adding the \(l\)th agent, we first consider \(F_{l1} \geq 2\|v_{l1}\|^2 + 2\xi_{l1} (u_{l} - u_{1} + \lambda_1 v_{l1})\). Selecting \(u_{l} = u_{1} + \lambda_1 v_{l1}\), makes \(F_{l1} \geq 0\) and we need to show that all the other constraints are satisfied. Because \(v_{ij} = v_{ij} + v_{ij}, v_{ij} = -v_{ij}\), and (19) holds for \(i, j, l, l - 1, i \neq j\) by the induction assumption, for all \(i = 2, \ldots, l - 1\) we have

\[
\lambda_1 v_{l1} + u_{l} - u_{l} = -u_{l} = 0
\]

Thus, for all \(i = 1, \ldots, l - 1, F_{li} \geq 2\|v_{il}\|^2 \geq 0\) and the induction argument completes the feasibility part. Because the optimal program has (strictly convex) quadratic cost and linear constraints, there is a unique solution. Feasibility of PCCA follows because, by changing variables, the constraint set takes the same form as that of the centralized controller with only the cost function being different.

**Remark 1:** From the proof of Proposition 1 it is clear that we have one extra degree of freedom to assure feasibility and collision avoidance even if one agent is nonresponsive.
but with acceleration known to others. Second, each agent braking proportional to its velocity, \( u_i = -\lambda_1 v_i \), is a feasible action. The QP problem remains feasible even if agent deceleration is limited, provided its speed is appropriately limited too.

The result carries over to circular agents with unicycle kinematics and the distance-based CBF. If velocity is the control variable, as is often assumed in robotics literature, then obviously \( v = 0 \) is a feasible action. With acceleration \( a \) the control variable, a proportional braking \( a_i = -\lambda_1 v_i + (\theta_i - \omega_i) 0 \) works similar to the holonomic case considered above.

V. SIMULATION RESULTS

We now compare the CBF collision-avoidance algorithms reviewed above in Monte Carlo simulations of five agents maneuvering in an enclosed area. The agents are modeled as disks of radius \( r_0 = 2 \) with the center motion given by a double integrator in two dimensions as in (10). A static outer circle of radius \( R_0 = 11 \) acts as an additional (soft) barrier constraint to enclose the space containing the agents. The controller sample time is \( \Delta T = 50 \text{ ms} \), and the baseline controller \( u_0 \) for each agent is computed by LQR with \( Q = 0.2I_4 \) and \( R = I_2 \). For computation of the QP constraints (13), we choose \( l_0 = 6 \) and \( l_1 = 5 \) to satisfy \( l_i^2 \geq 4l_0 \) (i.e., negative real eigenvalues). All the algorithms use this same set of parameters.

For each simulation run, agents are assigned random beginning and end locations somewhere within the outer circle enclosure. These initial/final locations are screened for any agent-to-agent overlap as well as overlap with the outer circle, assigning new ones until 100 feasible initial and final positions are generated. Each algorithm then ran from the same 100 initial to the corresponding final positions. Fig. 2 shows a time snapshot of one of these runs—the agents, their beginning and end locations, as well as their past and future paths.

To assess the algorithms, a set of metrics was used to compare liveness, collisions, and feasibility. Liveness is measured by convergence time; we assess how long it takes for all the agents to reach within 0.1 units from their destinations with the velocity magnitude less than 0.1 units/s. Each simulation was run for 100 s and assessed for convergence. It was found that all nonconvergent runs at 100 s had gridlocked and were not expected to converge. We did not use the deconfliction algorithms for gridlocks because they need a preferred passing direction agreed up front [9], [16] or determined on line with agent-to-agent communication [6].

Table I shows the aggregated results for the Centralized (14), DF (15), DR (16), and PCCA (17) policies without any additional radius margin added. PCCA was implemented with either a sample delay or a low-pass filter with a time constant \( \tau = 0.2 \text{ s} \) to break the algebraic loop. Table I shows the minimum convergence time to be similar for each algorithm, while the maximum is quite varied. The max and mean values do not include the gridlocked simulation runs for DF and DR. In general, DF and DR algorithms exhibit less liveness and take longer as previously reported in the literature (e.g., [16]). We see this better in Fig. 3, where the simulation trials are sorted by the average convergence time for each trial over all algorithms from maximum to minimum.

It has been established above that the Centralized and PCCA controllers are always feasible and the simulations confirmed this. Nearly a third of the DF and DR simulations exhibited infeasible QPs at some point. In this case, the QP solver [8] was configured to return the “least infeasible” solution. Except for slacked constraints on the outer static circle, the algorithms were implemented in pure form without alternative mechanisms to handle infeasibility.

For collision avoidance in the multiagent case, the centralized controller exhibits the best results. While the barrier is shown to be violated (i.e., \( h_{\text{min}} = -0.002 \); \( h_{\text{min}} \) is the minimum agent-to-agent barrier value over all the agents and all 100 trials), this is due to the selection of sampling time. When the sampling time was reduced, barrier violation disappeared as expected. A pictorial comparison shown in Fig. 4 displays the minimum agent-to-agent barrier value sorted by average of \( h_{\text{min}} \) over all the algorithms. DF has larger,

![Fig. 2. Five-agent simulation time snapshot with past and future paths.](image)

### TABLE I

| Method       | Converge Time (sec) | \( h_{\text{min}} \) | # gridlock | # infeasible |
|--------------|---------------------|-----------------------|------------|-------------|
| Centralized  | 7.45                | 0.002                 | 0          | 0           |
| DF           | 7.55                | -2.84                 | 3          | 27          |
| DR           | 7.55                | -1.53                 | 4          | 32          |
| PCCA         | 7.35                | -0.015                | 0          | 0           |
| PCCA_0.2     | 7.35                | -0.007                | 0          | 0           |

![Fig. 3. Convergence time for 100 Monte Carlo simulation runs, sorted by average from max to min excluding gridlocked trials.](image)
persistent violations. Both the Centralized and PCCA controllers exhibit minimal barrier violation and are almost indistinguishable in the plot.

We reran the Monte Carlo simulations using the worst case agent-to-agent barrier violations $h_{\text{min}}$ recorded in Table I. This $h_{\text{min}}$ is converted to $r$ to add the specific radius margin to each algorithm. The results in Table II show the barrier violations with the agents’ actual size $r_0$; all methods now effectively avoid collision with radius margin added. Note also that the radius margins added for DF and DR grew the agent sizes enough to induce additional gridlocks and infeasibility. One could iterate on the barrier margin required for both DF and DR to achieve $h_{\text{min}}$ closer to zero, similar to the other methods. For the Centralized algorithm and potentially for the PCCA with unit delay, one could use one of the barrier margin formulas for sampled data systems from [5], but they are unlikely to work for the other three.

It has been known that pure decentralized policies (DF/DR) may produce gridlocks – gridlock (deadlock) resolution methods [9], [16] were proposed for just such policies. It has also been known that the Centralized policy does not gridlock, though we do not think that an explanation has been offered. The results above show that PCCA will not gridlock either and its liveness is practically indistinguishable from the Centralized policy. Next, we try to explain these findings on a related problem simplified enough to make it analytically tractable.

VI. EQUILIBRIUM INSTABILITY AND LIVENESS

In this section, we consider a simpler problem with only two agents. To make the problem nontrivial for our objective, we give them only one degree of freedom – longitudinal acceleration. The agents will have to negotiate which one goes first through an intersection. It will turn out that the gridlocking algorithms (DF, DR) create a set of stable equilibrium (gridlock) points, while for those that did not gridlock (Centralized, PCCA), the equilibria are unstable. In other words, liveness comes from instability.

The problem considered is depicted in Fig. 5. The control action is the velocity, that is, each agent is the single integrator

$$\dot{x}_i = v_i, \quad i = 1, 2. \quad (20)$$

Here, $x_1$ and $x_2$ denote the distance of the two agents from the origin in the intersection point. The goal of each agent is to move at their desired velocity $v_i > 0$, assumed constant, while avoiding collision with the other agent. For collision avoidance, we use the same distance-based CBF

$$h(x) = x_1^2 + x_2^2 - r^2 = x^T x - r^2 \quad (21)$$

with $r \geq 2r_0$, $x = [x_1, x_2]^T$. The barrier constraint is

$$2x_1v_1 + 2x_2v_2 + \lambda h(x_1, x_2) \geq 0 \quad (22)$$

where $\lambda > 0$ is a design parameter – the barrier bandwidth. Because $x_1$ and $x_2$ cannot be simultaneously 0 in the admissible set $C$ if both control actions $v_1$ and $v_2$ are available for collision avoidance, the problem is feasible and $h$ is a CBF.

A. DF AND DR EQUILIBRIA AND THEIR STABILITY

For the DR policy, each agent assumes half of the responsibility for satisfying the barrier constraint while having only its own control at its disposal. For agent 1, the QP is

$$\min_{v_1} \|v_1 - v_1\|^2 \quad \text{such that} \quad \frac{\lambda}{2} h + 2x_1v_1 \geq 0. \quad (23)$$

Note that converting the DR into the DF policy would amount to removing (1/2), or equivalently multiplying $\lambda$ by 2. Hence, all the conclusions apply to DF as well. The solution to (23) is

$$v_i^* = \begin{cases} v_0, & \text{if } \frac{\lambda}{2} h + 2x_1v_1 \geq 0 \\ -\frac{\lambda}{4x_1} h, & \text{if } \frac{\lambda}{2} h + 2x_1v_1 < 0. \end{cases} \quad (24)$$

The case for agent 2 is symmetric.

Now, we consider the critical case, in terms of gridlocks, where the barrier constraints for both agents are active

$$\dot{x}_1 = -\frac{\lambda}{4x_1} h$$

$$\dot{x}_2 = -\frac{\lambda}{4x_2} h. \quad (25)$$
First, because \( x_i < 0 \), both agents’ constraints will activate while \( h(x) > 0 \). After they activate, \( \dot{h} = -\lambda h \), which results in \( h(x(t)) > 0 \) for all \( t \), that is, guarantees collision avoidance. Second, the only way for the two agents satisfying (25) to move backward is if \( \dot{h}(x) < 0 \), but this has just been ruled out. Therefore, once they enter the shaded region in Fig. 6 they are stuck because neither moves backward. Thus, any trajectory that hits the shaded curvilinear triangle will result in a gridlock. The equilibria of (25) in the quarter-plane \( \{ x_1 < 0, x_2 < 0 \} \) satisfy \( h(x) = 0 \), the arc or radius \( r \) as shown in Fig. 6. With \( \dot{h} = -\lambda h \), the arc is attractive from the points outside of it. Linearizing around an equilibrium point \( x_e \), we obtain the linear system \( \dot{\xi} = A_{dr} \xi \) in the \( \xi = x - x_e \) coordinates, where

\[
A_{dr} = \begin{bmatrix}
-\frac{\lambda}{2} & \frac{\lambda x_2}{2x_e} \\
\frac{\lambda x_1}{2x_e} & -\frac{\lambda x_2}{2}
\end{bmatrix}.
\]  

(26)

The eigenvalues of the matrix \( A_{dr} \) are zero and \( -\lambda \). This does not necessarily prove that the equilibrium \( x_e \) is stable because the system (25) is nonlinear. However, because the agents do not go backward, when initialized near any equilibrium point on the arc, they stay close to it, proving stability.

**B. Centralized Controller Equilibrium Analysis**

With the Centralized controller, the agents have full information about each others’ intentions and solve the same QP

\[
\min_{v_1, v_2} \| v_1 - v_01 \|^2 + \| v_2 - v_02 \|^2 \quad \text{such that} \quad \lambda h + 2x_1 v_1 + 2x_2 v_2 \geq 0.
\]

(27)

After the constraint activates, the coupled dynamics becomes

\[
\begin{align*}
\dot{x}_1 &= -\frac{\lambda x_1}{2\|x\|^2} h + \frac{x_2^2}{\|x\|^2} v_01 - \frac{x_1 x_2}{\|x\|^2} v_02 \\
\dot{x}_2 &= -\frac{\lambda x_2}{2\|x\|^2} h - \frac{x_1^2}{\|x\|^2} v_01 + \frac{x_2^2}{\|x\|^2} v_02.
\end{align*}
\]

(28)

By adding and subtracting the right-hand sides of (28) and setting them to 0 we obtain the equilibrium set, defined by the intersection of the arc \( h(x) = 0 \) and the line \( x_2 v_01 - x_1 v_02 = 0 \) as shown in Fig. 7 – that is, the equilibrium is a single point.

Linearizing around the equilibrium \( x_e \), we obtain the dynamics \( \dot{\xi} = A_{dr} \xi \), with the state \( \xi = x - x_e \) as above, and the eigenvalues of the state matrix \( \text{eig}(A_r) = [-\lambda, ((v_{01}^2 + v_{02}^2)^{1/2})/r] \) there we have used the solution for the equilibrium: \( x_{1e} = -(v_01 r)/((v_{01}^2 + v_{02}^2)^{1/2}), i = 1, 2 \). Note that the second eigenvalue is positive and the equilibrium point \( x_e \) is unstable, explaining the lack of gridlocks for the Centralized policy. Only trajectories that start exactly on the stable manifold – a set (in this case a line) of measure 0 – would end up at the equilibrium (that is, gridlocked), while all other trajectories move away exponentially fast, allowing agents to clear the intersection.

**C. PCCA Equilibrium Analysis**

As described above, the PCCA policy performs co-optimization, uses 0 instead of the unknown desired velocity for the other agent, and feeds the filtered difference between the other agent’s observed and host computed velocities as the known “disturbance”

\[
\left[ v_{11}^*, v_{12}^* \right] = \arg \min_{v_1, v_2} \| v_1 - v_01 \|^2 + \| v_2 - v_02 \|^2 \quad \text{such that} \quad \lambda h + 2x_1 v_1 + 2x_2 v_2 + \mu_1 \geq 0
\]

where

\[
\begin{align*}
\dot{w}_1 &= 1/\tau (-w_2 - v_2 - v_{12}^*), \\
\dot{w}_2 &= 1/\tau (-w_1 - v_1 - v_{11}^*).
\end{align*}
\]

(29)

The closed form solution to the PCCA QP is given by the following equation:

\[
\left[ v_{11}^*, v_{12}^* \right] = \begin{cases} [v_01, 0], & \text{if } \mu_1 \geq 0 \\ [v_01, 0] - \frac{\mu_1}{2\|x\|^2} x^T, & \text{if } \mu_1 < 0 \end{cases}
\]

(30)

with \( \mu_1 = \lambda h + 2x_1 v_01 + 2x_2 v_02 \). The control \( v_{11}^* \) is then implemented as the velocity for agent 1. The equations for agent 2 are symmetric.

After both agents activate constraints, the dynamics for the combined system is given by the following equation:

\[
\begin{align*}
\dot{x}_1 &= -\frac{\lambda h}{2\|x\|^2} x_1 + \frac{x_2 v_{10} - x_1 v_{20}}{\|x\|^2} x_2 \\
\dot{x}_2 &= -\frac{\lambda h}{2\|x\|^2} x_2 - \frac{x_2 v_1 - x_1 v_2}{\|x\|^2} x_1 \\
\tau \dot{w}_1 &= \frac{x_2}{\|x\|^2} (x_2 v_{10} + x_1 v_{20} - x_2 v_1 - x_1 v_2) \\
\tau \dot{w}_2 &= \frac{x_1}{\|x\|^2} (x_2 v_{10} + x_1 v_{20} - x_2 v_1 - x_1 v_2).
\end{align*}
\]

(31)

Note that we should use a fast filter (i.e., the time constant \( \tau \) small) because the constraint adherence is within \( O(\tau) \) margin of error. Returning to the equilibrium analysis of (31), we obtain that the 1-D equilibrium set in \( \mathbb{R}^4 \) is defined by the following equation:

\[
\begin{align*}
h(x_e) &= x_{1e}^2 + x_{2e}^2 - r^2 = 0 \\
x_{1e} v_{2e} &= x_{2e} v_{10}, \\
x_{2e} w_{1e} &= x_{1e} v_{01}.
\end{align*}
\]

(32)
To analyze stability of an equilibrium point \((x_e, w_e)\) that belongs to the set defined by (32), we use the ratio \(\mu_e = (x_1e/x_2e)\) and change the variables

\[
\eta_1 = x_1 - x_1e - \mu_e(x_2 - x_2e)
\]
\[
\eta_2 = \mu_e(x_1 - x_1e) + x_2 - x_2e
\]
\[
\eta_3 = \mu_e(w_1 - w_1e) - (w_2 - w_2e)
\]
\[
\eta_4 = w_1 - w_1e + \mu_e(w_2 - w_2e).
\]

The linearized system around \((x_e, w_e)\) in the \(\eta\)-coordinates is

\[
\dot{\eta} = A_p \eta
\]

with the first eigenvalue being positive. Note that the 0 eigenvalue has moved from the physical states \(x\) to the controller states \(w\), while the exponentially unstable state \(\eta_1 = x_1 - x_1e - \mu_e(x_2 - x_2e)\) belongs to the physical space \((x_1, x_2)\) plane. The projection of trajectories onto the physical plane \((x_1, x_2)\) should resemble behavior shown in Fig. 7 and helps explain the lack of gridlock for the PCCA policy in the simulations shown in Section V.

Remark 2: The equilibria structure and their stability for the intersection considered in this section is summarized in Table III. This analysis correlates closely with the observed simulation results from Section V and points to the actual mechanism for gridlock avoidance. The policies that have unstable equilibria (Centralized and PCCA) produced no gridlocks in the five-agent Monte Carlo simulations. In contrast, the policies with stable equilibria had gridlocks observed. Existence of equilibria themselves cannot be avoided with nonconvex, compact obstacles, and (Lipschitz) continuous policies.

### VII. Conclusion

This brief compared several CBF-based algorithms for their performance in multiagent scenarios. The simulation results confirmed that the Centralized and PCCA algorithms are always feasible. They also showed minimal barrier violations while the DF and DR methods had a few larger violations due to infeasibility. More importantly, the Centralized and PCCA were faster and did not gridlock, while DF and DR did. To explain the observed behavior, we analyzed equilibria structures and stability on a simpler two-agent problem. It turned out that the policies exhibiting lack of gridlocks have unstable equilibria, while policies with observed gridlock have stable equilibria. Beyond the examples provided in this brief, it is not known to the authors how to design a method that has unstable equilibria.

### References

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” IEEE Trans. Autom. Control, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.
[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in Proc. 18th Eur. Control Conf. (ECC), Jun. 2019, pp. 3420–3431.
[3] A. Ansari and D. S. Bernstein, “Input estimation for nonminimum-phase systems with application to acceleration estimation for a maneuvering vehicle,” IEEE Trans. Control Syst. Technol., vol. 27, no. 4, pp. 1596–1607, Jul. 2019.
[4] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, “Control barrier certificates for safe swarm behavior,” in Proc. IFAC Conf. Anal. Design Hybrid Syst., Oct. 2015, pp. 68–73.
[5] J. Breeden, K. Garg, and D. Panagou, “Control barrier functions in sampled-data systems,” IEEE Control Syst. Lett., vol. 6, pp. 367–372, 2022.
[6] F. Celi, L. Wang, L. Pallottino, and M. Egerstedt, “Deconfliction of motion paths with traffic inspired rules in robot-robot and human–robot interactions,” IEEE Robot. Automat. Lett., vol. 4, no. 2, pp. 2227–2234, Apr. 2019.
[7] J. Chen, A. Singletary, and A. D. Ames, “Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach,” IEEE Control Syst. Lett., vol. 5, no. 1, pp. 127–132, Jan. 2021.
[8] G. Cimini, A. Bemporad, and D. Bernardini, (Sep. 2017). ODYS QP Solver. [Online]. Available: http://odys.it/qp
[9] J. Grover, C. Liu, and K. Sycara, “The before, during, and after of multi-robot deadlock,” Int. J. Robot. Res., vol. 42, no. 6, pp. 317–336, May 2023.
[10] M. Jankovic, “Robust control barrier functions for constrained stabilization of nonlinear systems,” Automatica, vol. 96, pp. 359–367, Oct. 2018.
[11] M. Jankovic and M. Santillo, “Collision avoidance and liveness of multi-agent systems with CBF-based controllers,” in Proc. 60th IEEE Conf. Decis. Control (CDC), Dec. 2021, pp. 6822–6828.
[12] Q. Nguyen and K. Sreenath, “Exponential control barrier functions for enforcing high-relative-degree-safety-critical-constraints,” in Proc. Amer. Control Conf. (ACC), Jul. 2016, pp. 322–328.
[13] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, “The hybrid reciprocal velocity obstacle,” IEEE Trans. Robot., vol. 27, no. 4, pp. 696–706, Aug. 2011.
[14] M. Santillo and M. Jankovic, “Collision free navigation with interacting, non-communicating obstacles,” in Proc. Amer. Control Conf., May 2021, pp. 1637–1643.
[15] P. Trautman and A. Krause, “Unfreezing the robot: Navigation in dense, interacting crowds,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Oct. 2010, pp. 797–803.
[16] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” IEEE Trans. Robot., vol. 33, no. 3, pp. 661–674, Jun. 2017.
[17] P. Wieland and F. Allgower, “Constructive safety using control barrier functions,” in Proc. IFAC Symp. Nonlinear Control Syst., 2007, pp. 462–467.
[18] S. Wilson et al., “The Robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems,” IEEE Control Syst. Mag., vol. 40, no. 1, pp. 26–44, Feb. 2020.
[19] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” IFAC-PapersOnLine, vol. 48, no. 27, pp. 54–61, 2015.