

Article

Revised Control Barrier Function with Sensing of Threats from Relative Velocity Between Humans and Mobile Robots

Zihan Zeng ^{1,2}, Silu Chen ^{2,*} , Xiangjie Kong ², Xiaojuan Li ¹, Chi Zhang ² and Guilin Yang ² ¹ School of Mechanical Engineering, Xinjiang University, Urumqi 830047, China² Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo 315201, China

* Correspondence: chensilu@nimte.ac.cn; Tel.: +86-574-86686981

Abstract

The mobile robot, which comprises a mobile platform and a robotic arm, has been widely adopted in industrial automation. Existing safe control methods with real-time trajectory alternation face difficulties in efficiently identifying threats from fast relative motion between humans and robots, causing hazards in environments of dense human–robot coexistence. This work firstly builds a safe mobile robot control framework in the kinematic sense. Secondly, the proximity between parts of a human and a mobile robot is efficiently solved by convex programming with parametric description of skew line segments. It is also no longer required to perform case-by-case analysis of skew line segments' relative pose in space. Thirdly, a novel threatening index is proposed to select the most threatened human parts based on mutual projection of human–robot relative velocity and their common normal vector. Eventually, this index is incorporated into the safety constraint, showing the improved safe control performance in the simulated human–mobile robot coexistence scenario.

Keywords: mobile robot; human–robot interaction; safety constraint; control barrier function; relative velocity; obstacle avoidance



Academic Editors: Paolo Russo and Fabiana Di Ciaccio

Received: 19 May 2025

Revised: 23 June 2025

Accepted: 24 June 2025

Published: 27 June 2025

Citation: Zeng, Z.; Chen S.; Kong, X.; Li, X.; Zhang, C.; Yang, G. Revised Control Barrier Function with Sensing of Threats from Relative Velocity Between Humans and Mobile Robots. *Sensors* **2025**, *25*, 4005. <https://doi.org/10.3390/s25134005>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A mobile robot incorporates a mobile platform and a robotic arm to form a highly flexible and efficient production unit [1]. This integration offers innovative solutions for managing complex tasks [2]. While human–robot collaboration scenarios continue to proliferate, significant challenges remain in establishing reliable safe control frameworks for dynamic environments [3].

Many methods have been proposed to address the challenges of path planning for mobile robots in complex environments, such as potential field [4,5] and danger field-based methods [6]. These methods are widely used for real-time obstacle avoidance due to their high computational efficiency. Control barrier functions (CBFs) are defined in control systems by ensuring that a system's state remains within a predefined safe state set through the design of constrained control inputs [7]. It is particularly efficient in real-time obstacle avoidance if the safe conditions are set up as inequality constraints in quadratic programming. However, CBFs may face issues of high computational complexity and occasional collision with high-dimensional space or low-sampling-rate perception and control, and their definition of obstacle distances may not be comprehensive enough [8]. In recent years, deep reinforcement learning (DRL) [9], neural belief tracking (NBT) [10], and other methods based on artificial intelligence

have also been applied to trajectory planning in the human–machine interaction. By learning and predicting the trajectories of dynamic obstacles, neural networks can significantly enhance the intelligence and adaptability of path planning [11,12]. However, they lack formal safety guarantees, and their performance relies on high-quality training data [13,14]. The CBF-based safe reinforcement learning (CSRL) algorithm has been employed to address target pursuit problems [15]. The combination of CBF and model predictive control (MPC) can address the efficiency issues of online obstacle avoidance in unstructured environments filled with static and dynamic obstacles [16]. The path planning and task allocation in multi-robot systems has recently been optimized by integration of DRL, MPC, and graph neural networks (GNNs) [17]. However, alternating the trajectory in the real time remains a problem when these advanced networks are incorporated.

By sensing and adjustment of the distance from the obstacles, robots can prevent collisions [18]. The trajectory of mobile robots in the environment with static obstacles can be planned through offline programming methods [19]. For a robotic arm with a fixed base, a conservative approach is to define the safe boundary based on robotic arm's own workspace [18]. However, mobile robots sometimes need to deal with crowded, dynamic human–robot environments, where the uncertainty of human motion greatly challenges the real-time trajectory planning [20]. The difficulty of trajectory planning is further increased with mobile robots as the kinematics of the robotic arms couples with that of the mobile platform, especially when dealing with multiple moving humans or obstacles [21]. The determination of minimum human–robot distance is done by case-by-case analysis depending on the relative locations of the skew line segments and their common normals [22]. The “look-backward-and-forward” algorithm is further developed to predict the human trajectory [23], while the estimation of prediction error is further incorporated into the discrete-time control barrier function (DCBF) constraints. However, for crowded environment with dense and fast-moving humans and mobile robots, considering only the minimum distance between the individual parts of a human and a mobile robot is insufficient to indicate the threat in the future. Notably, the human-robot relative velocity critically indicates the level of danger [24] or injury severity [25] when facing potential impacts. They further affect operators' mental workload, anxiety, and risk perception [26]. The robot's velocity is included as a sigmoid-like smooth function term to shape the CBF [27]. A broader range of such constraints are included, such as robot joint velocities and the end-effector's velocity, as well as wall boundaries and cylindrical capsules for human and robot parts [28]. However, how to incorporate the relative velocity into the safety index is not well studied when designing the robot controller.

This paper aims to incorporate the human–robot relative velocity in the control barrier function (CBF), to achieve safe control with real-time trajectory alternation for mobile robots in human–robot coexisting environments. The main contributions are as follows:

1. The coupling kinematics of the mobile robot with a mobile platform and a robotic arm is formed in the kinematic sense rather than solely with the fixed-base robot arm in [22], so that the kinematic control problem in human–robot environments is set up with a discrete-time control barrier function (DCBF) and discrete-time control Lyapunov function (DCLF).
2. By setting up the parametric description of skew line segments, the minimum distance between a pair of human skeletons and a link of a robotic arm (or an outline of a mobile platform) is efficiently solved in real time by convex programming. Compared with [22,28], it is no longer necessary to make a case-by-case analysis depending on the relative locations of the skew line segments and their common normal.
3. By mutual projections of relative velocity of parts of a human and a mobile robot and their common normal vector, two projection indexes are given. Thereafter, a novel

threat index is formed to give a normalized “distance” to select the most threatened human parts. In this way, the relative velocities between parts of humans and mobile robots are successfully incorporated into the DCBF-based constraint rather than just their absolute velocities [27,28].

Simulation of human and mobile-robot environments validates the effectiveness of the proposed safe control method for real-time trajectory alternation.

2. Related Works

Control Barrier Functions (CBFs) have emerged as a formal methodology for guaranteeing system safety in automation and robotics. They construct dynamic safety boundaries by transforming complex safety constraints into computable mathematical conditions, embedding these conditions within controller optimization frameworks to ensure system states remain within safe parameters [29]. Compared to traditional approaches, CBFs uniquely provide rigorous mathematical safety guarantees while meeting the real-time demands of modern control systems. Key research contributions include:

Standard CBFs based on Zeroing Barrier Functions: This class of CBFs define safe sets via differential inequalities governing system trajectories to prevent boundary violations [7]. Applications include collision avoidance for autonomous vehicles, collaborative robots, and UAV formations. Representative implementations encompass event-triggered CBFs for efficient traffic control [30], dual-arm coordination with obstacle avoidance [31], and minimum-intervention collision-free UAV control [32].

High-Order CBFs: This class of CBFs employ recursively defined function chains to convert safety constraints with relative degrees greater than one into feasible conditions explicitly incorporating control inputs [33]. They have been applied to predictive human-robot safety interaction and fall prevention for legged robots. Notable examples include control of articulated robots using dynamic obstacle trajectory prediction [22] and dynamic equilibrium maintenance combined with obstacle-aware trajectory optimization for bipedal robots [34].

Current challenges involve enhancing adaptability and robustness under strong dynamic coupling and unmodeled disturbances [35], alleviating computational bottlenecks (e.g., quadratic programming for multi-agent systems) [36], and mitigating prediction error accumulation in dynamic environments [37]. Potential solutions include cross-disciplinary integration, such as incorporating reinforcement learning into stochastic control frameworks [38]. In view of the above, CBFs are still promoted as a universal design paradigm for safety-critical applications, enabling reliable unmanned systems in industrial automation [39] and medical surgery [40].

3. Problem Formulation

This section firstly models the mobile robot under a discrete-time kinematic control framework. Later, the existing safe control framework is imported with both discrete-time control barrier function (DCBF) and discrete-time control Lyapunov function (DCLF).

3.1. Control of a Mobile Robot

The mobile robot consists of a mobile platform and a robotic arm, as shown in Figure 1.

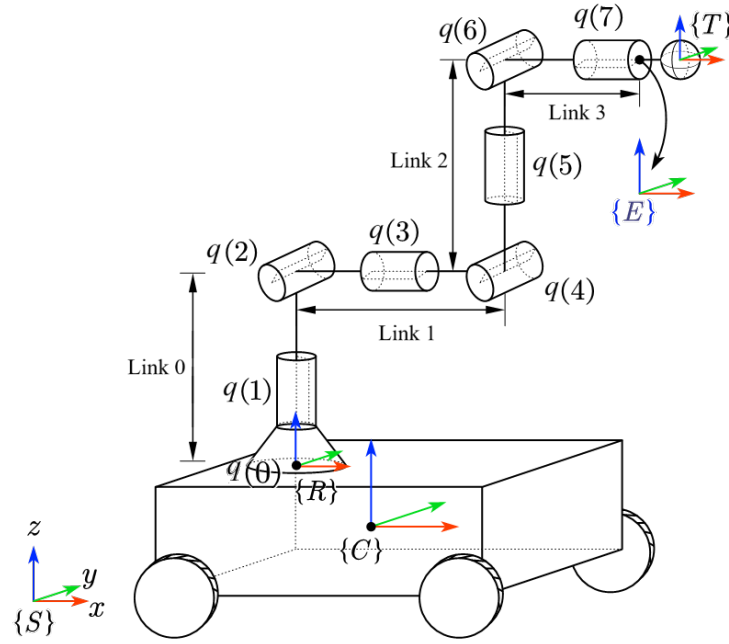


Figure 1. Schematics of a mobile robot.

The mobile platform is equipped with four Mecanum wheels, and its kinematics are given as:

$$p_{k+1} = p_k + v_k \delta_t \quad (1)$$

$$v_k = J_c u_k \quad (2)$$

where the subscript k denotes the k -th instant for all variables, $p \in \mathbb{R}^3$ represents the pose of mobile platform, $v \in \mathbb{R}^3$ is its planar velocity, $u \in \mathbb{R}^4$ is its control input and $J_c \in \mathbb{R}^{3 \times 4}$ is its Jacobian matrix.

Assume a robotic arm with n revolute joints is installed on the mobile platform, and its joint displacements are denoted as $q \in Q \subseteq \mathbb{R}^n$. Hence, its kinematics are given by

$$q_{k+1} = q_k + w_k \delta_t \quad (3)$$

$$a_k = g(\chi_k) = \hat{p}_k T_R^C T_a(q_k) \in SE(3) \quad (4)$$

where $\chi = [q^T \ p^T]^T$, $\bar{p} = [p(1) \ p(2) \ 0 \ 0 \ 0 \ p(3)]^T$, a is the pose in the task space, $T_a(q)$ is the forward kinematics of the robotic arm, T_R^C is the homogeneous transformation matrix (HTM) from the frame of the mobile platform to the base frame of the robotic arm, and w is the control input of the robotic arm, “ $\wedge$ ” is the operator for $(\bullet) \in \mathbb{R}^6 \rightarrow (\hat{\bullet}) \in SE(3)$.

The task is to track a desired pose a_d with the mobile robot's end effector. We can design a discrete-time feedback controller s with a control gain K_s :

$$s_k = K_s [(a_k)^{-1} a_{d,k}]^\vee \quad (5)$$

where s is the velocity of the end effector, “ $\vee$ ” is the operator for $(\bullet) \in SE(3) \rightarrow (\bullet)^\vee \in \mathbb{R}^6$, and

$$s_k = J_{e,k} v_k \triangleq \begin{bmatrix} J_{\text{base},k} & J_{a,k} \end{bmatrix} \begin{bmatrix} u_k \\ w_k \end{bmatrix} \quad (6)$$

In (6), $J_a \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix of the robotic arm, $J_{\text{base}} = [\text{Ad}_{T_a^{-1}(q)}^{\hat{p}-1}] \bar{J}_c$, where $[\text{Ad}_{(\bullet)}]$ is adjoint transformation. In addition,

$$\bar{J}_c = \begin{bmatrix} J_c(1, \bullet) \\ J_c(2, \bullet) \\ 0_{1 \times 4} \\ 0_{1 \times 4} \\ 0_{1 \times 4} \\ J_c(3, \bullet) \end{bmatrix} \in \mathbb{R}^{6 \times 4}.$$

where $J_c(1, \bullet)$ is the first row of J_c , etc. So the velocity (i.e., the control signal) of the entire mobile robot in the joint space v_k is:

$$v_k = J_{e,k}^+ K_s [(a_k)^{-1} a_{d,k}]^V \triangleq w(\chi_k) \quad (7)$$

where J_e^+ is the (pseudo-)inverse of J_e .

3.2. Safety and Stability Constraints

To achieve safe human–robot interaction, a safe set C is defined through a continuous differentiable function $H : Q \rightarrow \mathbb{R}$.

$$C = \{\chi_k \in Q : B(\chi_k) \geq 0\} \quad (8)$$

where $B(\cdot)$ is a discrete-time control barrier function (DCBF). $\Delta B(\chi_k) = B(\chi_{k+1}) - B(\chi_k) \neq 0$, $\chi \in \partial C$ and

$$\Delta B(\chi_k) \geq -\gamma B(\chi_k) \quad (9)$$

with a constant coefficient $\gamma \in (0, 1]$. This ensures the safe operation of the system.

Further, the stability of the system is constrained by the discrete-time Lyapunov function (DCLF) $V(\chi_k)$, and

$$\Delta V(\chi_k) \leq -\alpha V(\chi_k) + \delta \quad (10)$$

where $\Delta V(\chi_k) = V(\chi_{k+1}) - V(\chi_k)$, and $\alpha \in (0, 1]$ is a constant, $\delta \geq 0$ is a slack variable.

Based on the above constraints, the controller under human–robot coexisting environment is synthesized by:

$$v_k^* = \arg \min (\|v_k - \bar{v}_k\|^2 + \beta \delta^2) \quad (11)$$

$$\text{s.t. } \Delta B(\chi_k, v(\chi_k)) + \gamma B(\chi_k) \geq 0 \quad (12)$$

$$\Delta V(\chi_k, v(\chi_k)) + \alpha V(\chi_k) \leq \delta \quad (13)$$

$\bar{v}_k = [\bar{u}_k^T \quad \bar{w}_k^T]^T$ are the desired velocity when there is no human interruption, $\beta > 0$ is the slack variable.

4. Main Results

4.1. Distance Between a Mobile Robot and Human

This subsection aims to effectively seek the minimum distance between a pair of skew line segments by forming a convex optimization problem with parametric description of the points on the line segments.

Obtaining the closest distance between humans and the mobile robots in real time is necessary to establish CBF constraints in (12) for safe human–robot interaction. To calculate their closest distance, one can model the human body as a skeletal structure and simplify the robotic arm as a linkage mechanism, while the moving platform is modeled as a rectangle

with four enclosed line-segments. At this point, the problem is transformed to calculation of the distance between spatial line segments. Taking Figure 2 as an example, let points A and B be the endpoints of the i -th link of the robotic arm or outline of the moving platform, and points C and D be the endpoints of the n -th segment of human skeleton. The parametric equations of the pair of skew line segments (i, n) are:

$$P(s) = A + s(B - A), \quad s \in [0, 1] \quad (14)$$

$$Q(t) = C + t(D - C), \quad t \in [0, 1] \quad (15)$$

where s and t are normalized distances from points A and C , respectively.

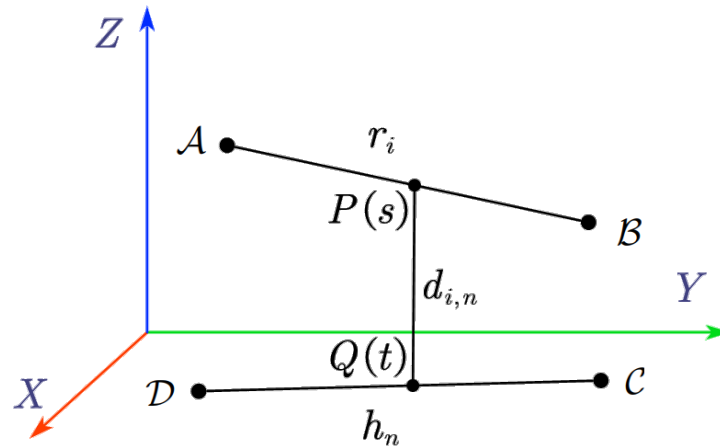


Figure 2. Distance between a pair of skew lines.

The problem is transformed into finding points $P(s)$ and $Q(t)$ such that the distance between them is minimized. This is achieved by the following convex optimization problem:

$$\min f_{i,n}(s, t) \quad \text{s.t.} \quad s, t \in [0, 1] \quad (16)$$

where

$$\begin{aligned} f_{i,n}(s, t) &= \|P(s) - Q(t)\|^2 \\ &= \|(A + s(\overrightarrow{AB})) - (C + t(\overrightarrow{CD}))\|^2 \end{aligned} \quad (17)$$

To deal with the constraints, we construct the Lagrangian function:

$$L = f_{i,n}(s, t) - \lambda_1 s + \lambda_2 (s - 1) - \lambda_3 t + \lambda_4 (t - 1) \quad (18)$$

where Lagrange multipliers $\lambda_1, \dots, \lambda_4$ are all non-negative. The solution (s^*, t^*) to the above optimization problem needs to satisfy the Karush–Kuhn–Tucker (KKT) conditions:

$$\begin{aligned} \frac{\partial L}{\partial s} &= 2(\overrightarrow{AB})^T (\overrightarrow{CA} + s(\overrightarrow{AB}) - t(\overrightarrow{CD})) \\ &\quad + \lambda_2 - \lambda_1 = 0 \end{aligned} \quad (19)$$

$$\begin{aligned} \frac{\partial L}{\partial t} &= -2(\overrightarrow{CD})^T (\overrightarrow{CA} + s(\overrightarrow{AB}) - t(\overrightarrow{CD})) \\ &\quad + \lambda_4 - \lambda_3 = 0 \end{aligned} \quad (20)$$

$$s, t \in [0, 1] \quad (21)$$

$$\lambda_i \geq 0 \quad (22)$$

$$\lambda_1 s = \lambda_2 (s - 1) = \lambda_3 t = \lambda_4 (t - 1) = 0 \quad (23)$$

Thus, the closest distance between all pairs of skew line segments is calculated by

$$d^{\min} = \inf_{i,n} \sqrt{f_{i,n}^*} \quad (24)$$

where $f_{i,n}^*$ is optimal $f_{i,n}$ obtained by solving (16) with the i -th line segment of the mobile robot and n -th segment of the human skeletons.

4.2. The Threatening Index Based on Relative Velocity

Based on the minimum distance between a human and a mobile robot, we can set up the corresponding safety constraint. However, this is insufficient to ensure the safety in real-time under complex interaction scenarios, especially when the humans and robots are moving fast. Thus, this subsection aims to extend the definition of “minimum distance” by considering the factor of the relative velocity between a human and a robot.

As shown in Figure 3, r_i denotes the i -th link of the robot, h_m and h_n represent two segments of the human skeleton, and $V_{r,(\bullet)}$ represents the velocity of $P_{(\bullet)}$. $V_{h,(\bullet)}$ represents the velocity of $Q_{(\bullet)}$. According to the method proposed in Section 4.1, the closest points between r_i and h_m are $P_{i,m}$ and $Q_{i,m}$, while the closest points between r_i and h_n are $P_{i,n}$ and $Q_{i,n}$. It can be observed that $d_{i,m} < d_{i,n}$. However, point $Q_{i,m}$ is moving away from point $P_{i,m}$, while point $Q_{i,n}$ is moving towards point $P_{i,n}$. Under the proximity-based DCBF constraint (12), $d_{i,m}$ is taken as the minimum distance, at which the robot will take evasive action towards point $Q_{i,m}$. However, this ignores the more threatening point $Q_{i,n}$, which may lead to a collision with h_n in the near future. Therefore, when constructing the DCBF constraint, it is also necessary to consider the units that give greater threats in relative motion in addition to the minimum distance between humans and the robot.

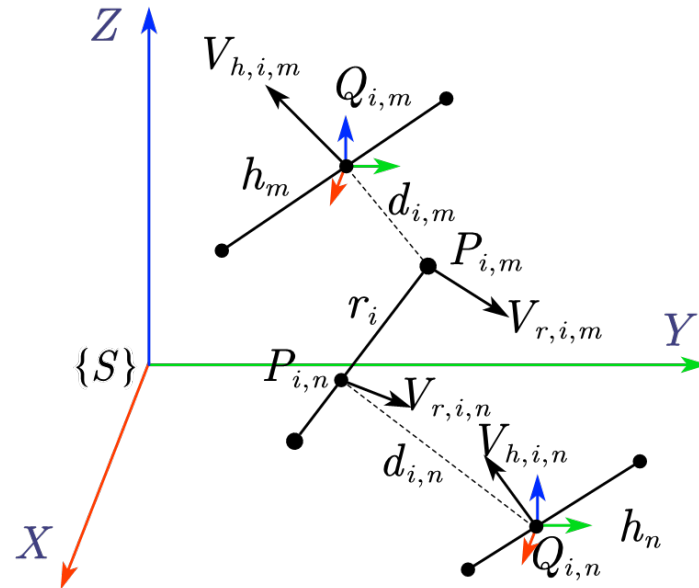


Figure 3. Minimum distance and relative velocity between the skew lines.

Taking r_i and h_n in Figure 3 as an example, the threat level to the system is quantified by the relative velocities between humans and robots. This is given by the threatening index based on the principle illustrated in Figure 4.

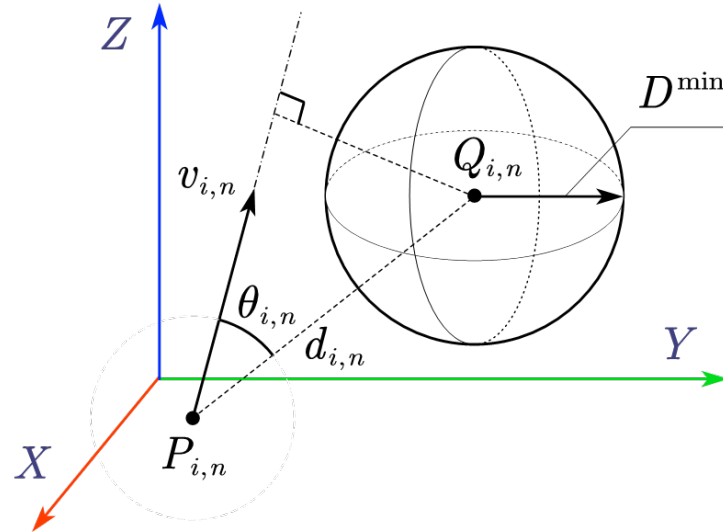


Figure 4. The influence of relative velocity to the safe threat.

As shown in Figure 4, $v_{i,n}$ is the relative velocity between $P_{i,n}$ and $Q_{i,n}$ and

$$v_{i,n} = V_{r,i,n} - V_{h,i,n}$$

θ is the angle between the relative velocity vector and the distance vector (sharing a common starting point), and $D^{\min}$ is the predefined threshold of safe distance between humans and robots. Hence, the threatening index $k_{i,n}$ for skew line segment pair (i, n) is calculated as

$$k_{I,i,n} = \frac{v_{i,n} \delta_t}{d_{i,n} \cos \theta_{i,n}} \quad (25)$$

$$k_{A,i,n} = \frac{D^{\min} - d_{i,n} \sin \theta_{i,n}}{D^{\min}} \quad (26)$$

$$k_{i,n} = \begin{cases} 0, & \text{if } k_{A,i,n} < 0 \\ k_{I,i,n}(k_{A,i,n} + 1), & \text{if } k_{A,i,n} \geq 0 \end{cases} \quad (27)$$

where $k_{I,i,n}$ is related to the intensity of the robot's impact on the human body at the current relative velocity. $k_{A,i,n}$ is related to the angle of impact at the time of collision, and δ_t is the control period.

- When $k_{A,i,n} < 0$, it indicates that the robot moving at the current relative velocity will not intrude into the safety range of the obstacle. Thus, the threatening index is set to zero.
- When $k_{A,i,n} \geq 0$, the robot has the risk of intruding into the safety range of the obstacle at the current relative velocity. In this case, $k_{i,n}$ is set to $k_{I,i,n}(k_{A,i,n} + 1)$, where $(k_{A,i,n} + 1)$ ensures that the threat coefficient is at least $k_{I,i,n}$ when $k_{A,i,n} = 0$. Thus, a non-zero threatening index is issued.

We compute the threatening index $k_{i,n}$ between each pair (i, n) and select the maximum $\bar{k}$. Later, we obtain the minimum $\underline{d}$ across distance $f_{i,n}$ for pair (i, n) with the identical maximum threatening index $\bar{k}$:

$$\bar{k} = \max_{i,n} k_{i,n}, \quad (28)$$

$$\underline{d} = \min\{f_{i,n} \mid k_{i,n} = \bar{k}\} \quad (29)$$

We can further refine the selection of the object pairs (i, n) , all of which have $f_{i,n} = d^{\min}$. This is done by selecting the pair with the highest threatening index:

$$S = \{(i, n) \mid f_{i,n} = d^{\min}\}, \quad (30)$$

$$(i^*, n^*) = \arg \max_{(i,n) \in S} k_{i,n}, \quad (31)$$

4.3. Revised Safety Constraints

After identifying the most threatened human parts with the threatening index based on human–robot relative velocity, a more comprehensive safety constraint is constructed in this section.

Based on the distance and threatening index between humans and robots, we can design multi-objective constraints for the safe set $\mathcal{C}$ defined in (12) with the prescribed safe distance $D^{\min}$ as

$$\Delta B(\chi_k) = \lambda_D B_D + (1 - \lambda_D) B_V \quad (32)$$

where B_D corresponds to the constraint of the closest distance, and B_V corresponds to the distance constraint of the object with the maximum threatening index based on the relative velocity. The specific constraints are as follows:

$$B_D = d^{\min} - D^{\min} \geq 0 \quad (33)$$

$$B_V = \underline{d} - D^{\min} \geq 0 \quad (34)$$

λ_D is the weight distribution coefficients for the two constraints, allocated based on the threatening index:

$$\lambda_D = \frac{1}{\bar{k} + 1} \quad (35)$$

To ensure the forward invariance of the set $\mathcal{C}$, the DCBF-based constraints (12) are further given as:

$$d_{k+1}^{\min} - d_k^{\min} \geq -\gamma(d_k^{\min} - D^{\min}) \quad (36)$$

$$\underline{d}_{k+1} - \underline{d}_k \geq -\gamma(\underline{d}_k - D^{\min}) \quad (37)$$

Unlike prior studies on fixed-base robotic arms that only consider the distance between humans and the robotic arm [22], this composite constraint considers both human–robot distance and relative velocity.

When the mobile robot aims to reach a desired posture after completing the trajectory, we can construct a constraint about $V(\chi_k)$ as

$$d_{g,k} = \|\varphi_k - \varphi_{d,k}\|_2 \quad (38)$$

$$V(\chi_k) \triangleq \epsilon - d_{g,k} > 0 \quad (39)$$

where d_g is the distance between the current end-effector position φ and the desired position φ_d . φ and φ_d are the position vectors extracted from the a in (4) and a_d in (5), respectively. ϵ is a very small constant. So the DCLF-based constraint in (13) is given as:

$$-d_{g,k+1} + d_{g,k} \leq -\alpha(\epsilon - d_{g,k}) + \delta \quad (40)$$

By constructing the aforementioned constraints, the mobile robot can be equipped with a framework based on the hybrid DCBF and DCLF, ensuring safe and efficient operation in dense human-robot coexistence environments. The proposed safe control method for a mobile robot is summarized in Figure 5.

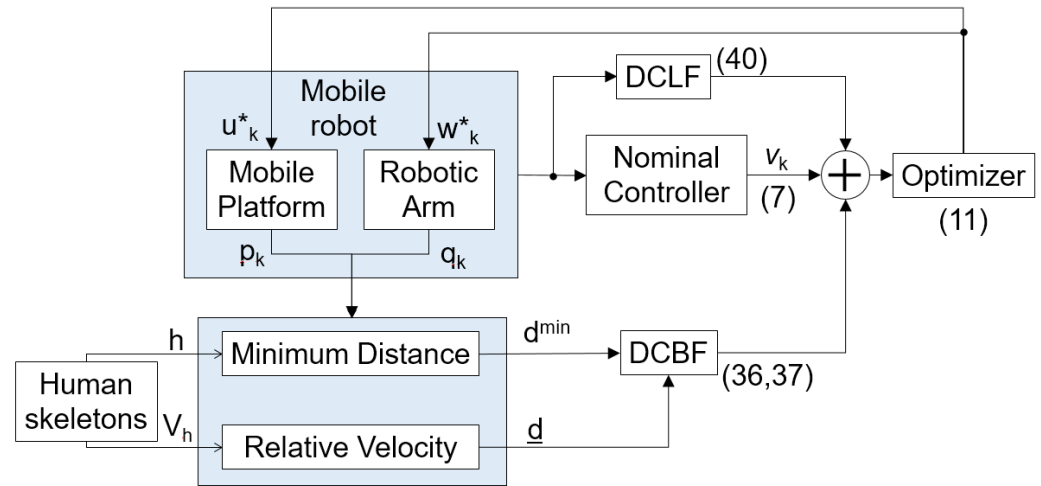


Figure 5. The overview of proposed safe control method.

5. Simulation

5.1. The Setting of Simulation

To verify the effectiveness of the proposed method, two mobile robots are imported for simulation. Each consists of a mobile platform with four Mecanum wheels and a KUKA iiwa robotic arm. The task is to let them approach their corresponding target poses in a human–robot environment, interrupted by three humans. The paths of the humans are manually preset but not known in advance by the mobile robots, and the posture data of human bodies are obtained from a motion capture dataset. The simulation is performed on a workstation equipped with a 12th Gen Intel® Core™ i9-12900H 2.50 GHz processor. The optimization problem is solved by `fmincon` function in MATLAB(2022a), selecting the interior-point method for its low memory footprint and real-time capability. The initial solution is set to the current state values.

Figure 6 illustrates the initial state of the mobile robot. A red dot in the XY-plane marks the origin of the mobile platform’s frame, and the blue lines outline its edges. A robotic arm is mounted on each mobile platform. The joint positions are indicated by red dots, and the links are depicted as black line segments. A pentagram denotes the target for the robotic arm’s end-effector. The mobile robots aim to reach their targets while maintaining safe distances from all humans. Robot 1’s mobile platform starts at $[-3, -3, \pi/2]^T \in SE(2)$, and its arm is initially rested vertically. The goal is to let the end effector move to $[0, 0.97, 0, 3.68, 3, 1]^T \in SE(3)$. Robot 2’s platform starts at $[3, 3, -\pi/2]^T \in SE(2)$ with its arm being initialized in the same way as Robot 1, targeting $[0, 0.97, 0, -2.32, -3, 1]^T \in SE(3)$. The homogeneous transformation matrix from the mobile platform’s frame to the robotic arm base is $(T_R^C)^\vee = [0, 0, 0, 0, 0, 0.3]^T$. Human 1 moves back and forth between the robots’ target points, Human 2 orbits the origin, and Human 3 traverses radially across the mobile robots’ paths.

When screening obstacle-avoidance objects, the mobile platform is simplified as a planar quadrilateral bounding box. For the robotic arm, the rotations of joints 1, 3, and 5 in the KUKA iiwa configuration do not affect the positions of subsequent links. Therefore, the robotic arm can be simplified into four segments: from the base to joint 2 (Link 0 in Figure 1, joint 2 to joint 4 (Link 1), joint 4 to joint 6 (Link 2), and joint 6 to joint 7 (Link 3). Link 0 is fixed to the mobile platform and will not extend beyond the planar quadrilateral bounding box of the mobile platform. The mobile robot is thus simplified into four objects: the planar quadrilateral bounding box of the mobile platform and Link 1, 2, and 3 of the robotic arm.

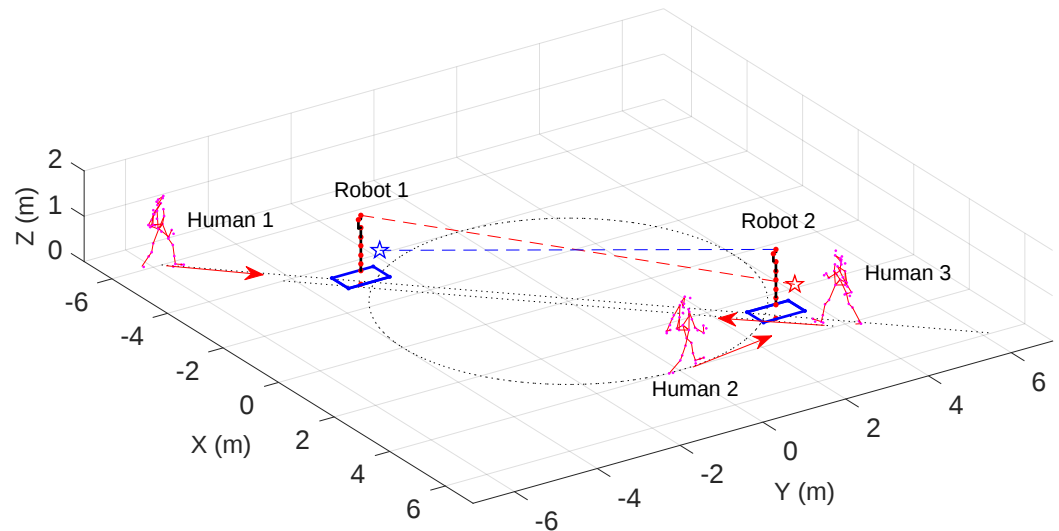


Figure 6. The initial state in the human–robot coexistence environment.

By incorporating the DCBF-based constraints (36), (37) into the optimization formulation (12), we can obtain $\dot{\chi}_k$ to drive a mobile robot to maintain a safe distance from humans or other obstacles during movement. The DCBF constraints based on proximity and threatening index are set up for the mobile platform and Link 1, 2, and 3. Consequently, there are eight DCBF-based constraints to ensure safety in such human–robot coexistence environments. When the end-effector approaches the target point, the DCLF-based constraint (13) is applied to all control methods, ensuring that the end-effector moves as close as possible to a_d if no human is interfering while maintaining stability. Consequently, the optimization problem (11) is formulated, and we impose the following joint limits for the mobile platform:

- Limits of wheels' velocity: [10, 10, 10, 10];
- Limits of joints' velocity for the robotic arm: [85, 85, 100, 75, 130, 135, 135];
- Limits of joints' position limits for the robotic arm: [170, 120, 170, 120, 170, 120, 175].

We would like to compare our safe control method with the conventional one containing DCBF with proximity-based DCBF. Further, we also compare our method with the Dynamic Window Approach (DWA) and Artificial Potential Field (APF) method to accomplish navigation and obstacle avoidance for the mobile platform and robotic arm, respectively [41]. For the method with proximity-based DCBF, the simulation setup is identical to that used for both proximity-based and threatening-index-based DCBF. For the DWA, the prediction horizon is set to six steps, and the scoring weights for the target, velocity, obstacles, orientation, and path smoothness are [0.70, 0.15, 0.05, 0.05, 0.05]. For the APF, the obstacle repulsion and the target attraction coefficients are set to be 0.5 and 1.5 accordingly. The codes for implementation of this simulation example are given in the Supplementary Material.

5.2. Results and Discussions

Figures 7 and 8 show the distances between the two mobile robots and human units. The blue lines correspond to the control with DCBF constraints that consider both the closest point and most threatening point, designated as “DCBF (D&V)”. For comparison, the red lines correspond to the safe control with proximity-based DCBF constraints, designated “DCBF (D)”, and the green lines correspond to the control with DWA&APF method. It can be observed that with our method, all parts of the mobile robots maintain safe distances from humans. In contrast, targets to be dodged are incorrectly selected by proximity-based DCBF and undesired closer human–robot distances at subsequent moments occur,

as proximity-based DCBF neglects the relative velocity during human–robot interactions and it treats humans as static obstacles at the current moment. Meanwhile, the mobile robots' moving trajectories by the DWA&APF method sometimes violate the safety distance when encountering overly dense scenarios or aggressive maneuvers. Additionally, in dense environments, our method achieves a slight temporal advantage over the proximity-based DCBF constraints and less oscillation compared with that by DWA&APF method.

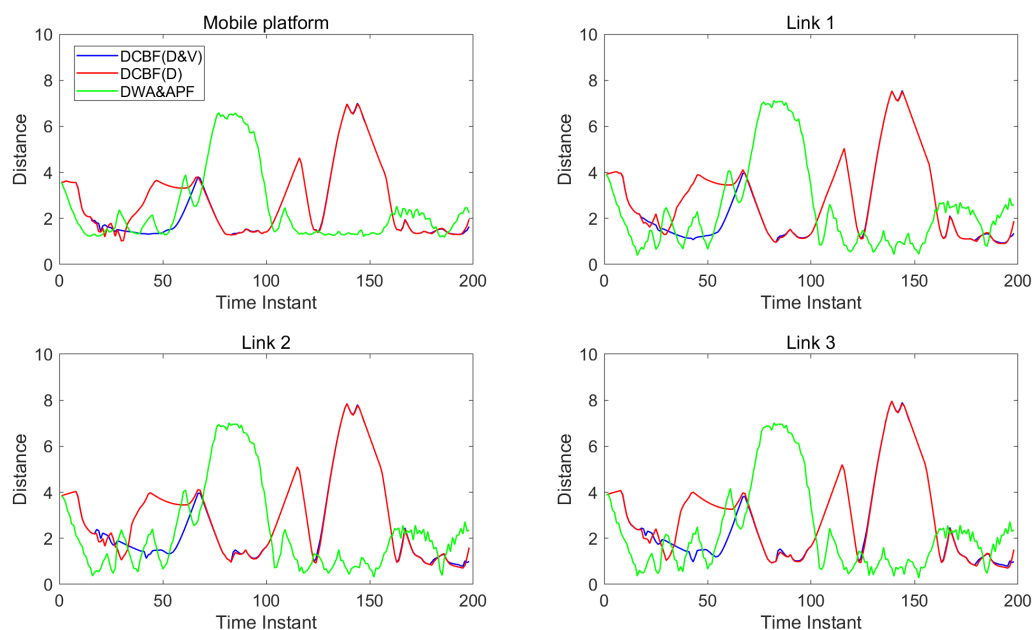


Figure 7. The distance (in meters) between humans and the components of mobile robot 1.

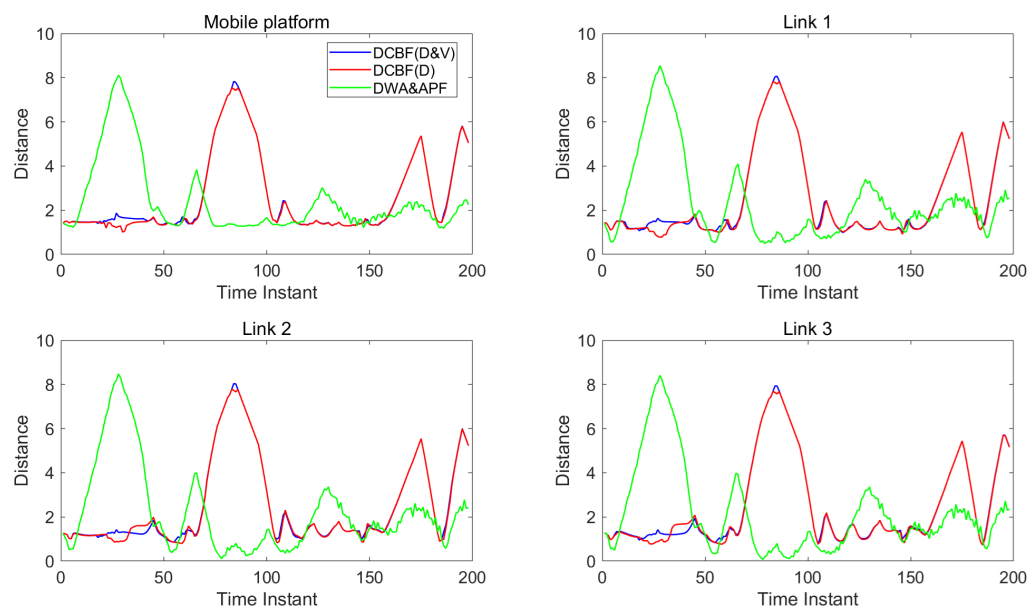


Figure 8. The distance (in meters) between humans and the components of mobile robot 2.

Figure 9 demonstrates the target convergence process of dual mobile robots' end-effectors, where red and blue trajectories represent Robot 1 and 2, respectively. Comparative experiments reveal that under integrated constraints with proximity-based DCBF (36) and highest-threatening-index-based DCBF (37), Robot 1 reaches its target at the 60th time instant, converging 25% faster compared to the distance-constraint-only approach, which requires 80 time instants. Both configurations of constraints enable Robot 2 to

synchronously arrive at its target at the 78th instant, limited by narrow passage dynamic constraints. Meanwhile, by means of the DWA&APF method, Robot 2 successfully initiates its navigation task. However, Robot 1 fails to move towards its target point. This is due to the fact that a human unit is near Robot 1 initially, causing the controller to prioritize obstacle avoidance. Subsequently, as Robot 2 approaches, it acts with other obstacles to obstruct Robot 1, leading Robot 1 to become stuck and unable to complete the navigation. Since Robot 1 is stuck near Robot 2's target point, Robot 2 also fails to reach its destination. This performance enhancement stems from the proactive threat mitigation mechanism embedded in our proposed integrated constraints, which accelerates disengagement from high dense human–robot coexistence zones.

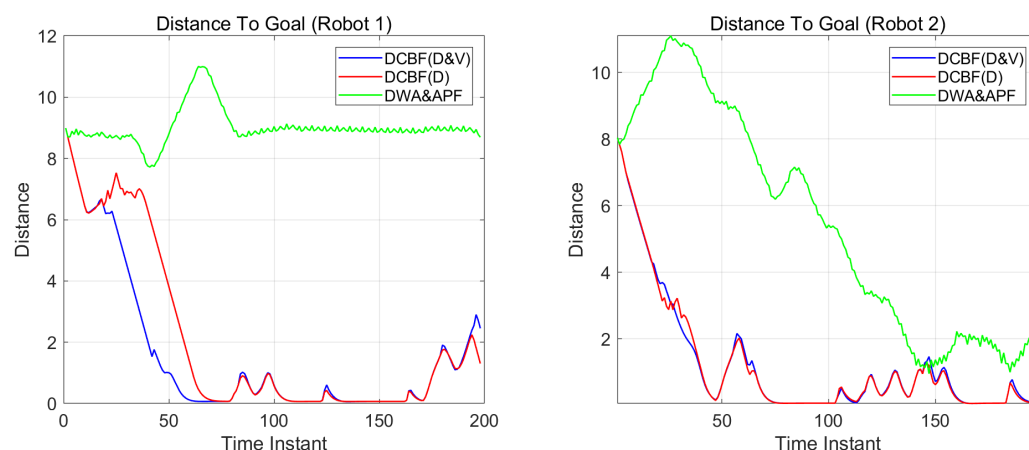


Figure 9. Distances to the goals (in meters) for the end effectors of the two mobile robots.

6. Conclusions

This paper establishes a safe control framework for environments with multiple co-existing mobile robots and humans. The coupled kinematic model for a mobile platform and a robot arm is firstly built with defined discrete-time control barrier function (DCBF) and discrete-time Lyapunov function (DCLF). Subsequently, the closest distances between parts of humans and the mobile robots are solved by convex programming with parametric description of pairs of skew lines. Specifically, the threatening index is defined and incorporated into DCBF by mutual projections of the relative velocity and common normal vector of skew line segments. Simulation of a scenario with two mobile robots and three humans validated the effectiveness of the proposed method. The proposed method can be applied to the safe control of mobile robots in crowded human–robot environments. The kinematic modeling of the current work is applicable to the mobile robot with Mecanum wheels. In future work, we will extend our proposed method to a mobile robot with power caster wheels and dual robotic arms.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/s25134005/s1>.

Author Contributions: Conceptualization, Z.Z. and S.C.; methodology, Z.Z., S.C. and X.K.; software, Z.Z.; validation, Z.Z., S.C. and X.K.; formal analysis, Z.Z., S.C. and X.K.; investigation, Z.Z. and S.C.; resources, S.C., X.L., C.Z. and G.Y.; data curation, Z.Z.; writing—original draft preparation, Z.Z. and S.C.; writing—review and editing, S.C., X.K. and X.L.; visualization, Z.Z.; supervision, S.C. and X.L.; project administration, S.C. and X.L.; funding acquisition, S.C., C.Z. and G.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Nature Science Foundation of China (U20A20282, U21A20121, 92048201, 52127803), Zhejiang Provincial Natural Science Foundation (LD24E050010, LD22E050007), Zhejiang Science and Technology Department (2023C01176), Ningbo Major Research Plan (2023Z041).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

APF	Artificial Potential Field
CBF	Control Barrier Function
CSRL	CBF-based Safe Reinforcement Learning
DCBF	Discrete-time Control Barrier Function
DCLF	Discrete-time Control Lyapunov Function
DRL	Deep Reinforcement Learning
DWA	Dynamic Window Approach
GNN	Graph Neural Networks
MPC	Model Predictive Control
NBT	Neural Belief Tracking

References

1. Ehrmann, C.; Min, J.; Zhang, W. Highly flexible robotic manufacturing cell based on holistic real-time model-based control. *Procedia CIRP* **2024**, *127*, 20–25. [\[CrossRef\]](#)
2. Stillström, C.; Jackson, M. The concept of mobile manufacturing. *J. Manuf. Syst.* **2007**, *26*, 188–193. [\[CrossRef\]](#)
3. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* **2018**, *55*, 248–266. [\[CrossRef\]](#)
4. Park, D.H.; Hoffmann, H.; Pastor, P.; Schaal, S. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In Proceedings of the Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots, Daejeon, Republic of Korea, 1–3 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 91–98. [\[CrossRef\]](#)
5. Merkt, W.; Ivan, V.; Vijayakumar, S. Continuous-time collision avoidance for trajectory optimization in dynamic environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Venetian Macao, Macau, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 7248–7255.
6. Lacevic, B.; Rocco, P.; Zanchettin, A.M. Safety assessment and control of robotic manipulators using danger field. *IEEE Trans. Robot.* **2013**, *29*, 1257–1270. [\[CrossRef\]](#)
7. Ames, A.D.; Grizzle, J.W.; Tabuada, P. Control barrier function based quadratic programs with application to adaptive cruise control. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 6271–6278.
8. Taylor, A.J.; Ames, A.D. Adaptive safety with control barrier functions. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1399–1405.
9. AbuJabal, N.; Baziyad, M.; Fareh, R.; Brahmi, B.; Rabie, T.; Bettayeb, M. A comprehensive study of recent path-planning techniques in dynamic environments for autonomous robots. *Sensors* **2024**, *24*, 8089. [\[CrossRef\]](#)
10. Mrkšić, N.; Séaghdha, D.O.; Wen, T.H.; Thomson, B.; Young, S. Neural belief tracker: Data-driven dialogue state tracking. *arXiv* **2017**, arXiv:1606.03777. [\[CrossRef\]](#)
11. Zou, A.M.; Hou, Z.G.; Fu, S.Y.; Tan, M. Neural networks for mobile robot navigation: A survey. In Proceedings of the Advances in Neural Networks—ISNN 2006, Chengdu, China, 28 May–1 June 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1218–1226.
12. Abdelrahman, A.F.; Valdenegro-Toro, M.; Bennewitz, M.; Plöger, P.G. A neuromorphic approach to obstacle avoidance in robot manipulation. *arXiv* **2024**, arXiv:2404.05858. [\[CrossRef\]](#)
13. Singh, R.; Ren, J.; Lin, X. A review of deep reinforcement learning algorithms for mobile robot path planning. *Vehicles* **2023**, *5*, 1423–1451. [\[CrossRef\]](#)

14. Hart, F.; Waltz, M.; Okhrin, O. Two-step dynamic obstacle avoidance. *Knowl.-Based Syst.* **2024**, *302*, 112402. [\[CrossRef\]](#)
15. Deng, Y.; Gao, J.; Feroskhan, M. Ensuring safety in target pursuit control: A CBF-safe reinforcement learning approach. *arXiv* **2024**, arXiv:2411.17552.
16. Zhang, Y.; Tian, G.; Wen, L.; Yao, X.; Zhang, L.; Bing, Z.; He, W.; Knoll, A. Online efficient safety-critical control for mobile robots in unknown dynamic multi-obstacle environments. *arXiv* **2024**, arXiv:2402.16449. [\[CrossRef\]](#)
17. Li, Z.; Shi, N.; Zhao, L.; Zhang, M. Deep reinforcement learning path planning and task allocation for multi-robot collaboration. *Alex. Eng. J.* **2024**, *109*, 408–423. [\[CrossRef\]](#)
18. Arents, J.; Abolins, V.; Judvaitis, J.; Vismanis, O.; Oraby, A.; Ozols, K. Human–robot collaboration trends and safety aspects: A systematic review. *J. Sens. Actuator Netw.* **2021**, *10*, 48. [\[CrossRef\]](#)
19. Katona, K.; Neamah, H.A.; Korondi, P. Obstacle avoidance and path planning methods for autonomous navigation of mobile robot. *Sensors* **2024**, *24*, 3573. [\[CrossRef\]](#)
20. Finean, M.N.; Petrović, L.; Merkt, W.; Havoutis, I.M.I. Motion planning in dynamic environments using context-aware human trajectory prediction. *Robot. Auton. Syst.* **2023**, *166*, 104450. [\[CrossRef\]](#)
21. Schneier, M.; Schneier, M.; Bostelman, R. *Literature Review of Mobile Robots for Manufacturing*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015.
22. Zhu, Y.; Chen, S.; Zhang, C.; Piao, Z.; Yang, G. Development of adaptive safety constraint by predicting trajectories of closest points between human and co-robot. *J. Intell. Manuf.* **2024**, *35*, 1197–1206. [\[CrossRef\]](#)
23. Chen, S.; Zhu, Y.; Liu, Y.; Zhang, C.; Piao, Z.; Yang, G. A “look-backward-and-forward” adaptation strategy for assessing parameter estimation error of human motion prediction model. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2629–2636. [\[CrossRef\]](#)
24. Giallanza, A.; Scalia, G.L.; Micale, R.; Fata, C.M.L. Occupational health and safety issues in human-robot collaboration: State of the art and open challenges. *Saf. Sci.* **2024**, *169*, 106313. [\[CrossRef\]](#)
25. Han, D.; Park, M.Y.; Choi, J.; Shin, H.; Rhim, S. Analysis of human-robot physical interaction at collision. In Proceedings of the 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Nagoya, Japan, 4–6 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 153–156.
26. Koppenborg, M.; Nickel, P.; Naber, B.; Lungfiel, A.; Huelke, M. Effects of movement speed and predictability in human–robot collaboration. *Hum. Factors Ergon. Manuf. Serv. Ind.* **2017**, *27*, 197–209. [\[CrossRef\]](#)
27. Ferraguti, F.; Landi, C.T.; Costi, S.; Bonfè, M.; Farsoni, S.; Secchi, C.; Fantuzzi, C. Safety barrier functions and multi-camera tracking for human–robot shared environment. *Robot. Auton. Syst.* **2020**, *124*, 103388. [\[CrossRef\]](#)
28. Merckaert, K.; Convens, B.; Wu, C.j.; Roncone, A.; Nicotra, M.M.; Vanderborght, B. Real-time motion control of robotic manipulators for safe human–robot coexistence. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102223. [\[CrossRef\]](#)
29. Ames, A.D.; Xu, X.; Grizzle, J.W.; Tabuada, P. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Autom. Control* **2017**, *62*, 3861–3876. [\[CrossRef\]](#)
30. Sabouni, E.; Ahmad, H.M.S.; Xiao, W.; Cassandras, C.G.; Li, W. Optimal Control of Connected Automated Vehicles with Event-Triggered Control Barrier Functions: A Test Bed for Safe Optimal Merging. *arXiv* **2023**, arXiv:2306.01871. [\[CrossRef\]](#)
31. Shi, K.; Chang, J.; Feng, S.; Fan, Y.; Wei, Z.; Hu, G. Safe Human Dual-Robot Interaction Based on Control Barrier Functions and Cooperation Functions. *IEEE Robot. Autom. Lett.* **2024**, *9*, 9581–9588. [\[CrossRef\]](#)
32. Yang, T.; Miao, Z.; Yi, G.; Wang, Y. Safety-Critical Control of Quadrotor UAVs with Control Barrier Functions. In Proceedings of the 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO), Xishuangbanna, China, 5–9 December 2022; pp. 1074–1079.
33. Xiao, W.; Belta, C. High-Order Control Barrier Functions. *IEEE Trans. Autom. Control* **2022**, *67*, 3655–3662. [\[CrossRef\]](#)
34. Liu, J.; Li, M.; Huang, J.K.; Grizzle, J.W. Realtime Safety Control for Bipedal Robots to Avoid Multiple Obstacles via CLF-CBF Constraints. *arXiv* **2023**, arXiv:2301.01906. [\[CrossRef\]](#)
35. van Wijk, D.E.J.; Coogan, S.; Molnar, T.G.; Majji, M.; Hobbs, K.L. Disturbance-Robust Backup Control Barrier Functions: Safety Under Uncertain Dynamics. *IEEE Control Syst. Lett.* **2024**, *8*, 2817–2822. [\[CrossRef\]](#)
36. Min, X.; Baldi, S.; Shi, Y.; Yu, W. Safe Control of Multi-Agent Systems via Low-Complexity Control Barrier Functions. *IEEE Trans. Autom. Control* **2025**, 1–14. [\[CrossRef\]](#)
37. Buyukkocak, A.T.; Aksaray, D.; Yazıcıoğlu, Y. Control Barrier Functions with Actuation Constraints under Signal Temporal Logic Specifications. In Proceedings of the 2022 European Control Conference (ECC), London, UK, 12–15 July 2022; pp. 162–168.
38. Emam, Y.; Notomista, G.; Glotfelter, P.; Kira, Z.; Egerstedt, M. Safe Reinforcement Learning Using Robust Control Barrier Functions. *arXiv* **2022**, arXiv:2110.05415. [\[CrossRef\]](#)
39. Garg, K.; Usevitch, J.; Breeden, J.; Black, M.; Agrawal, D.; Parwana, H.; Panagou, D. Advances in the Theory of Control Barrier Functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems. *Annu. Rev. Control* **2024**, *57*, 100945. [\[CrossRef\]](#)

-
40. Qin, W.; Yi, H.; Fan, Z.; Zhao, J. Haptic Shared Control Framework with Interaction Force Constraint Based on Control Barrier Function for Teleoperation. *Sensors* **2025**, *25*, 405. [[CrossRef](#)] [[PubMed](#)]
 41. Cao, L.; Tang, L.; Cao, S.; Sun, Q.; Zhou, G. Smooth Optimised A*-Guided DWA for Mobile Robot Path Planning. *Appl. Sci.* **2025**, *15*, 6956. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.