International Journal of Advanced Robotic Systems

OPEN ACCESS ARTICLE

A Cross-domain Survey of Metrics for Modelling and Evaluating Collisions

Review Paper

Jeremy A. Marvel[1,*] and Roger Bostelman[1]

1 Intelligent Systems Division, Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA
* Corresponding author E-mail: jeremy.marvel@nist.gov

Received 17 Sep 2013; Accepted 02 Jul 2014

DOI: 10.5772/58846

Abstract This paper provides a brief survey of the metrics for measuring probability, degree, and severity of collisions as applied to autonomous and intelligent systems. Though not exhaustive, this survey evaluates the state-of-the-art of collision metrics, and assesses which are likely to aid in the establishment and support of autonomous system collision modelling. The survey includes metrics for 1) robot arms; 2) mobile robot platforms; 3) nonholonomic physical systems such as ground vehicles, aircraft, and naval vessels, and; 4) virtual and mathematical models.

Keywords Collision Metrics, Collision Modelling, Robot Collisions, Mobile Robot Collisions, Vehicular Collisions

1. Introduction

Accurately detecting, predicting, and avoiding collisions with objects are key safety functions for automated physical systems. These functions enable mechanical systems to operate in complex environments while simultaneously protecting personnel and equipment from harm. Moreover, the ability to understand the consequences of these collisions enables protective systems to be designed that minimize the potential hazards incurred as a result of collisions. These hazards become increasingly prevalent as the use and nature of automation extends beyond manufacturing and into human-occupied healthcare and service environments. However, the environmental and operational conditions that make collision detection and avoidance necessary also give rise to large variability in the mechanisms for measuring and modelling collisions.

In any physical system, a given pair of objects has three possible proximal states: *separate*, *touching*, and *colliding*. Colliding differs from touching; colliding results in the deformation or destruction of one or both objects, while touching does not. Most common metrics measuring separation are useful for collision avoidance. However, they are of little help when quantifying actual or potential collision severity. Separation metrics, however, remain the prevalent scoring method for safety systems due to computational constraints and practical considerations. Specifically, most would rather see collisions avoided than quantified.

In this review, we provide a summary of the metrics identified for modelling, detecting, and avoiding collisions across multiple domains. Section two outlines metrics used with robot arms, which are focused on maintaining safe distances between the robot and any obstacles inside its work volume. Section three discusses mobile robot safety systems, which attempt to navigate through an unstructured and variable world. Sections four and five extend the scope of investigation, and explore the metrics used in fields directly related to robotics. Section four reviews collision metrics used in manned and semi-autonomous vehicular systems such as automobiles, aircraft, and naval vessels, while section five reviews the metrics of collisions and penetrations in virtual systems.

2. Robot Arm Collision Metrics

The open-chain manipulator paradigm of a robot arm attached to an affixed pedestal or rail (e.g., Figure 1) has been the prominent focus of robot safety literature for the past several decades. These robots are limited in reach and are physically confined to a set position. Despite their limited reach, injuries and deaths worldwide have been attributed to accidents involving traditional industrial robot arms [1-3].

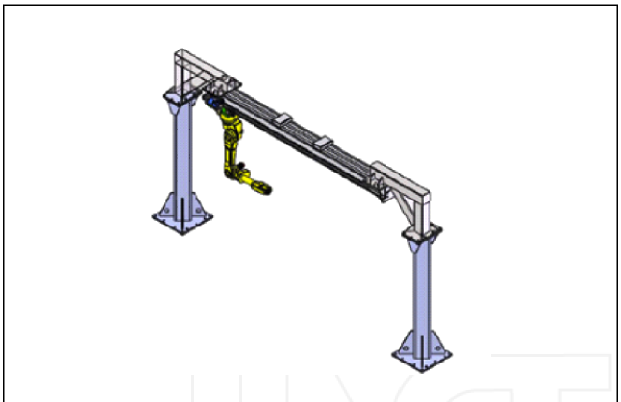

Figure 1. An example robot configuration where a robot arm is underslung on a linear rail for an increased work envelope

Traditionally, robot safety has focused on workcell ergonomics, designed specifically to minimize the possibility of collisions between the robot and outside elements such as walls or supporting beams, machinery, or people [4]. With the advent of modular and agile manufacturing, this focus has since shifted toward robotic controllers and safety systems that can monitor dynamically defined workspaces to assess safety hazards [5]. As the working environment changes, new potentials for collisions involving robots emerge.

The perception of possible collisions between a robot arm and an outside element results in one of two possible actions: an adjustment of the arm's trajectory to move around the potential collision (*active obstacle avoidance*), or a modulation of the arm's velocity along its current trajectory to allow the conflict state to clear itself (*velocity scaling*). A potential collision is detected by means of distance checks between a model of the robot and the sensed obstacles.

Typically, actual collisions are not modelled because they constitute constraint violations, resulting in the robot reverting to a known failsafe model (e.g., an emergency stop). When collisions are modelled, the goal is not to estimate the degrees of state space violations. Rather, the goals are centred on capturing the effects and potential damages to the robots (e.g., [6]) or humans (e.g., [7, 8]). These models, however, can be used to evaluate and tune hazard metrics for determining danger zones for alternative safety mechanisms such as power and force limiting. In Ogorodnikova's work [9], for example, the author simulated single degree of freedom, dynamic, mass-spring models of forces and accelerations in collisions to tune end-effector velocities to minimize discomfort and injury.

2.1 Active Obstacle Avoidance

Adjusting a robot's position and path trajectory based on sensed hazards has been an active topic of research for decades. Implementations typically fall into one of two possible categories: planning-level trajectory changes, and reaction-based trajectory modifications. The former modifies the initial trajectory prior to the robot moving based on *a priori* knowledge of obstacles. The latter adjusts the motions of the robot on-the-fly.

2.1.1 Stationary Obstacles

From the breadth of literature on the topic, common implementations of dynamic trajectory modulation involve navigating a robot arm around and amongst sensed, static objects in the work zone. Some algorithms use the robot's current state to generate fully formed trajectories around objects based on the perceived occupied spaces. This requires the robot to utilize maps of its environment and imposes additional computational and memory overhead for map generation and maintenance. These algorithms have a high probability of finding a solution vector to a goal state. Other algorithms create a baseline trajectory to a given goal state and then add permutations as the robot's inverse kinematic solution brings it closer to sensed objects. These processes require less overhead and are more capable of responding to changing environmental conditions. However, the algorithms are more susceptible to local optima and can steer the robot into a conflict state.

One of the earliest successful—and widely modified—active, obstacle-avoidance algorithms was based on potential fields [10]. This algorithm simultaneously drives the robot effector toward a goal state and repels the robot away from obstacles present in the workspace. As the distance from the goal state increases, so too does the attractive pull toward it. Similarly, as the distance to an object decreases, the repulsive radial push away from the object increases (see Figure 2). Implementations of this algorithm have two important features. First, the processes of path planning and obstacle avoidance are combined at a low level. Second, both processes can be accomplished in real time. Potential fields, however, have a significant limitation: the virtual repelling fields neither penalize nor expressly prevent collisions. This limitation exists because the basis for motion along a given trajectory is the balance between attraction toward a desired position and repulsion away from a perceived obstacle.

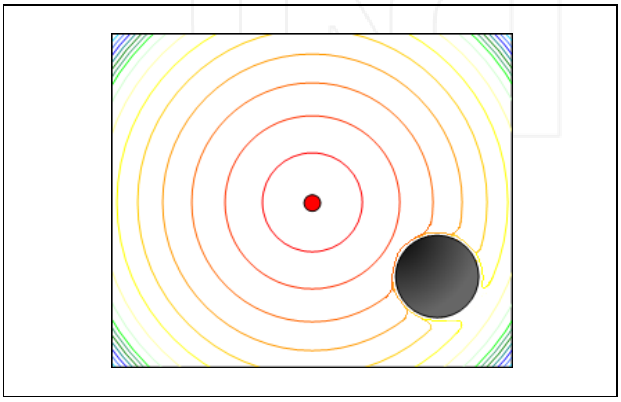

Figure 2. The attractive strength of potential fields increases as the robot approaches the target position (red central dot), and is likewise repulsed by obstacles (dark grey sphere). Here, the intensity of the target's attractive field is indicated by the colour of the concentric circles. Redder lines indicate stronger attraction to the target than the blue, yellow, and green lines. The robot follows a gradient path based on the strength (distance) of the fields. Because the obstacle warps the attractive fields, the robot's trajectory is changed to move around the potential collision.

Related to potential fields are *reflexive* and *virtual force* controllers. Reflexive controllers accept or reject high-level commands based on rapid evaluations of configuration space (C-Space) maps that define boundary regions based on clearances to nearby obstacles (e.g., [11]). Distances from these boundary regions drive limits on speed and motion to avoid collisions. Virtual force controllers (e.g., [12, 13]) quantify distances between the robot and mapped obstacles as simulated forces. These forces act against the robot being controlled by common, compliant, motion-control algorithms. As the distances decrease, the motion controller increases the counteracting virtual forces. Unlike potential fields, the virtual force implementation attempts to adhere to a predetermined trajectory. However, the forces in a simulated force-controlled motion can override this trajectory.

When *a priori* knowledge of the obstacles in the robots' work volume is not available, collision-avoidance processes must rely on sensors to perceive changes in the environment. The research of Hosoda, Sakamoto, and Asada [14] demonstrated this capability by using 2D image-plane data to move in collision-free paths (see Figure 3). This method does not require the reconstruction of three-dimensional geometry because it enforces a constraint that forbids the projected trajectory from intersecting with the projected obstacles.

Though immobile, these unmapped obstacles still make it difficult to provide smooth and stable trajectories. This difficulty arises because active sensing systems provide constant feedback to the obstacle-avoidance path planner. The planner uses this feedback to make frequent changes to the trajectory, which can result in jitter and instability. While it is possible to use the sensors to map obstacles for smoother trajectory planning, a number of researchers have shown that such mapping is not required if the proximity to obstacles can be measured accurately. For example, it has been shown in simulation [15] that a manipulator with a series of link or joint sensors could actively avoid multiple potential collisions while simultaneously limiting trajectory oscillations. Both can be achieved even while attempting to avoid only the closest collision. Similar results are seen (e.g., Feddema and Novak [16]) when using arm-mounted, capacitance-based proximity sensors to adjust the commanded joint velocities along the normal axis of the sensors.

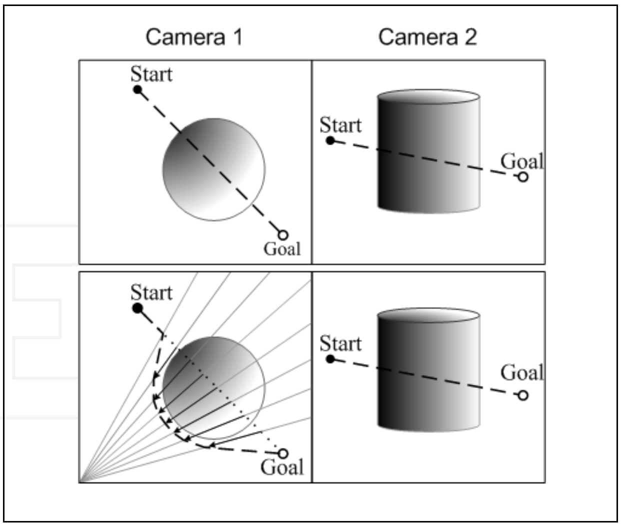

Figure 3. Multiple-camera systems can detect possible collisions (top left and right) and can generate collision-free paths without reconstructing three-dimensional geometries provided that a path according to any camera is clear of any collisions (lower left). The rays originating in the lower left corner represent a constraint, in image space, on the trajectory shift. This constraint is used as a fast mechanism for path planning around potential collisions.

Instead of maximizing the separation distances, however, maintaining a set distance between the robot and potential collisions could be just as effective [17]. By using an artificial neural network, the inverse kinematics of an arm can be computed to keep the obstacles a minimum distance from the robot. The resulting solution treats obstacles as bounding spheres, and forces the robot to follow the contour of an object as it makes progress toward a goal state.

2.1.2 Non-Stationary Obstacles

Obstacles that are moving within the workspace pose an additional challenge for robot safety. Just as with the stationary obstacles, the safety systems must actively and safely adjust the motions of the robots. Due to the dynamic nature of the non-stationary obstacles, the safety system must also track the active elements within reach of the robot. Researchers have attempted to simplify the problem by focusing on sensing objects and making obstacle avoidance a factor of reaction rather than premeditation. The potential fields method, for example, has been extended successfully to provide obstacle avoidance for dynamic objects. The system proposed by Newman and Hogan [18] uses dynamic attractive and repulsive fields to perform high-speed tasks in the presence of moving obstacles. Virtual forces are exerted on the robot based on logical field combinations in both Cartesian and joint-space configurations. Similarly, Park *et al.* [19] extended the implementation from Khatib [10] by making the potential fields gradient-based rather than distance-based. As a result, *dynamic potential fields* are generated for obstacle avoidance.

A benefit to potential fields and virtual forces is that they can be applied at a low level, and thus provide real-time response to potential collision events. However, they suffer from the same limitations as their static counterparts in that collisions are not, strictly speaking, avoided entirely. The repulsive field of one obstacle may therefore cause the robot to move through another obstacle that has a smaller repulsion. Moreover, the active nature of both the obstacles and the robot's responses to those obstacles makes it difficult to prove *a priori* trajectory verification, and cannot therefore predict collision-free paths. Without additional checks, the likelihood of the robot moving into a bad or dangerous state is increased.

Other approaches are more direct in implementing obstacle avoidance. One system by Liu, Deng, and Zha [20], for instance, uses established path-planning algorithms to navigate around a simulated human upper torso making random arm movements. In simulation, this system creates a C-space mapping around cylinders representing the robots. The system then uses rudimentary distance metrics (based on *safe*, *dangerous*, and *invalid* edge distinctions) to perform an A*-like graph search. Another system by Bosscher and Hedman [21] provides collision avoidance for two industrial robots that have overlapping workspaces modelled as spherical shells. Taking into account the known kinematics of one robot, the other actively avoided collisions with the spherical shell to 1) maintain or exceed a set minimum separation between the two robots, and 2) remain within the limits of joint angles and velocities. In stark contrast to both approaches, the solution proffered by De Luca *et al.* [22] reacts to sensed collisions using lightweight robots. These robots then conform around the collisions utilizing Cartesian force information.

A limitation on all dynamic collision-avoidance algorithms lies with the reliance on the accurate sensing and identification of obstacles. Many implementations of dynamic collision avoidance require having perfect information of obstacle pose, volume occupancy, and direction and speed of travel. Uncertainty in the sensing and timing of object motions may lead to errant or otherwise unpredictable robot behaviour that may not actually avoid collisions. Moreover, distinguishing obstacles from work objects is also problematic. Typically, the robot is expected to make physical contact with an object within its working volume to accomplish a task. The standard test pieces used to evaluate robot safety are not biomimetic, and may even resemble the robot's work piece [23]. Additional safeguards and administrative steps may be required, which ultimately lessens the importance of implementing advanced collision-avoidance algorithms.

2.2 Velocity Scaling

Rather than adjusting trajectories to skirt around potential collisions, robots may be programmed to scale their velocities to slow down or stop until the threat of collision has disappeared. The internal mechanisms and metrics for this form of obstacle avoidance are largely similar to the dynamic avoidance discussed earlier. Rather than actively moving the robot around an obstacle, however, velocity scaling methods assume that the unexpected obstacle will move away independent of the robot's motions. While more predictable in outcome, such methods are less predictable in time. Robots can deadlock as they wait for the obstacle to move outside of some defined border region. This, as a result, may reduce productivity throughput. Nevertheless, velocity scaling represents the majority of safety systems driven by non-contact sensors (see Section 2.3).

As with active collision avoidance, velocity-scaled safety systems rely on separation metrics to determine and maintain safe distances. In one example [24], first- and second-order instantaneous approximations are used to compute *time-to-collision*. This computation drives the collision detection, the end-effector velocity scaling, and the coordinated null-space optimization across multiple robots in a shared space. Another approach is to treat separations from static and mobile regions of sensor uncertainty as potential hazard states [25]. This has been demonstrated to be useful particularly in instances when sensors are unable to provide full information of the operational environment. In yet another instance, Kulić and Croft use a *danger index* [26] in a multi-tiered safety system. This index is a function of inertia and separation distance, and incorporates long-, medium-, and short- term safety goals. Long-term safety goals centre on safe planning, while medium- and short-term safety goals focus on trajectory scaling, and safe control, respectively. The trajectory scaling component, itself, is a function of the desired end-effector velocity and the calculated danger index.

In contrast, some researchers have taken the position that physical interactions between humans and robots in collaborative environments are inevitable. One such system by Haddadin *et al.* [27] scales the robot's trajectory by making the time element of the current trajectory a function of the output of a workspace observer. This system includes a safety mechanism that limits the transfer of kinetic energy from a moving robot to a human operator to minimize the risk of injury. Similarly, a report by the German Institute for Occupational Safety and Health [28] outlined pressure, force, and compression constant limits to minimize risk of injury. These limits are based on a literature survey of injury studies going back as far as the 1940s.

2.3 Commercial Solutions

Numerous instantiations of active obstacle avoidance and dynamic velocity scaling zones have been developed. However, relatively few are commercially available or are implemented in actual manufacturing environments. Instead, most implementations rely on static safety zones based on a distance metric for velocity scaling purposes. While research and experimental safety implementations are permitted to use arbitrary separation distances, distances for industrial systems are regulated by means of standards. One such standard often applied to manufacturing equipment is the International Organization of Standards (ISO) reference 13855 [29]. This standard provides a simple metric based on three variables: K, C, and T. K is expected maximum speed of the robot. C is the reach of a human operator. T is the distance the human can travel in the time necessary to safely stop the robot. These variables are then used to calculate the *minimum separation distance* using equation (1).

$$S = (K \times T) + C \tag{1}$$

If the distance between the machinery and the human falls below the value of S, the system brings the machine to a safe, controlled stop. The safety zone calculations of ISO 13855 provide the basis for the safety zones for robot cells defined in ISO Technical Specification 15066 [30], and, consequently, both parts of ISO 10218 [31, 32]. In these implementations, however, the equation is extended to include factors such as human travel speed, braking distance, braking time, and sensing uncertainty. Following the standards guidelines enables easier verification and validation. Productized variants of the velocity-scaling paradigm from robot vendors include [33-35]. After-market and integrated safety systems include camera- and laser-scanner-based solutions (e.g., [36] and [37], respectively).

3. Mobile Robot Collision Metrics

As with robot arms, most safety metrics for mobile robots and automated guided vehicles (AGVs) are based on task-specific performance factors rather than collision severity. Such factors include path and task optimization [38-40]. For path optimization, robot control laws focus on achieving the goal state without colliding with elements in the environment. Implementations of obstacle avoidance are more qualitative than quantitative. As a result, measurements of obstacle avoidance are Boolean in nature: either the robot avoided colliding with objects or it did not. Measures of obstacle avoidance rely almost exclusively on the distance to the nearest obstacle on the robot's path. This makes the direct comparison of collision avoidance algorithms nearly impossible. Comparisons are thus limited to computational metrics such as 'time to complete a task', 'number of path nodes explored', and 'lengths of paths' [41].

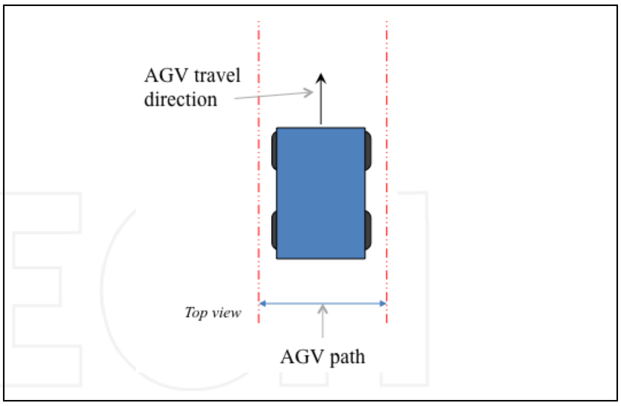

Figure 4. Top view of an AGV as it moves through the environment. Current safety standards mandate that the path of travel remain clear of all obstacles for a distance commensurate with the AGV's speed.

Not surprisingly, securing a buffered distance between the robot and any obstacles is the standard for mobile robotic platforms. The American National Standards Institute/Industrial Truck Standards Development Foundation (ANSI/ITSDF) standard B56.5-2012 [42] defines a *safety zone* for AGVs. In B56.5-2012, the *safety zone* is defined to be a distance buffer projected along the vehicle path (see Figure 4)—including potential instantaneous changes in direction—commensurate with the AGV's speed. Although there are no defined stopping distances for industrial vehicles, standard test methods must initiate a vehicle stop prior to the vehicle structure contacting a standard test piece.

Attempts to provide methods for evaluating stopping distances have produced similar metrics. However, their implementations vary considerably. For example, the study by Amato *et al.* [43] investigated numerous *distance metrics* for a probabilistic roadmap methodology. These metrics are used to select the next target location to which their local path planner should connect. Interestingly, the best distance metric was chosen because of its computational performance and roadmap connectivity rather than any quantifiable safety criterion.

In contrast, Alvarez's *Security Metrics* [44] attempt to quantify the safety of the robot passing through an obstacle-ridden environment. Security Metrics is based on three different measurements: SM_1, SM_2, and *Min*. SM_1 is the mean distance between the robot and all of the obstacles at all points in time for every sensor on the robot. SM_1 is used to identify when the robot is passing through obstacle-free areas. SM_2 is the minimum mean distance to the obstacles. SM_2 quantifies the risk taken in terms of the proximity of the robot to obstacles throughout the entire mission. *Min* is the minimum distance between the robot and any obstacle throughout the mission. *Min* measures the maximum risk taken.

The *Safety Cost Function* of the work of Sisbot, Marin, and Alami [45] operates under the pretext that the further away a robot is from an object (or human), the safer the interaction between the two will be. Every possible configuration of the robot has an associated cost. That cost is inversely proportional to the distance to the human. Moreover, the cost is a function of the human's associated state (such as standing, sitting, etc.).

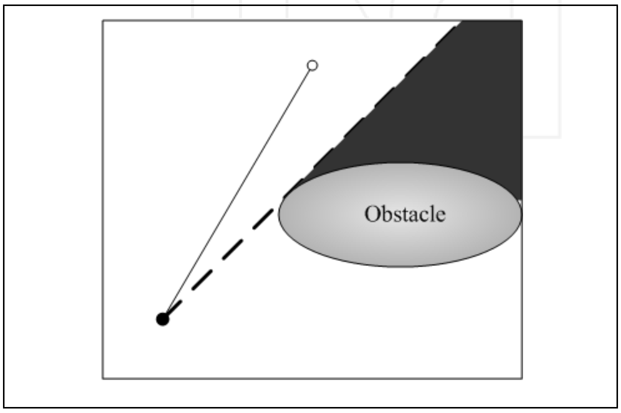

Figure 5. As a robot (black dot, lower-left) moves toward its target destination (white dot, upper-centre), it approaches unmapped or occluded regions in the operational space. To avoid collisions with anything in the unmapped region (black shadow), the robot's velocity is reduced to allow for sensor exploration.

Similarly, the *Collision Danger*, as defined by Toussaint [46], is calculated based on the heavy-side function that takes two arguments: the shortest distance between a pair of collideable objects and a predefined margin of safety.

A fundamental component of all collision-avoiding algorithms is the reliability of the robot's sensor suite. In instances of sensor uncertainty or severe clutter, the actions of the robot may be further tempered in order to verify a degree of certainty of a collision-free path. This is illustrated in the *Safety Criterion* of Miura, Negishi, and Shirai [47] where the motions of a mobile robot are slowed as it approaches an unmapped region (Figure 5). This gives the sensor suite sufficient time to determine that a given region in front of the vehicle is either occupied or not. This determination is typically made by means of feature identification or abstraction (e.g., [48]), or similarity to previously explored regions (e.g., [49]).

In contrast to the previous predictive approaches, some methodologies and metrics may permit minor slips in avoiding collision states. Probabilistic navigational systems such as the one described by Fulgenzi *et al.* [50] calculate a probability of collision based on cumulative uncertainties of the model and perception measurements. Such systems may enter collision states (or perceived collision states) if the sensor data becomes excessively noisy. As another metric, algorithms may be rated on both the maximum penetration of the robot into the collision state, and the maximum time spent in the collision state [51]. If both times are sufficiently (and arbitrarily) small, said collisions may even be forgivable.

4. Vehicular Collision Metrics

Vehicular-collision metrics are the basis for the warning systems on manned and semi-autonomous vehicles. These metrics account for many of the same environmental and configuration parameters as their robotic counterparts. The associated warning systems model operator behaviour, track and project vehicle characteristics and kinematics, issue warnings, and cause evasive procedures when warranted.

Much effort has gone into the modelling of collisions, including the effects on the chassis, environment, and passengers and drivers. These models have proved intrinsically useful for the vehicular systems for which they were designed. Some researchers, however, have raised concerns that the models do not accurately predict the severity of potential injuries inflicted on humans by robots [52]. Regardless, the metrics utilized for collision detection and avoidance draw on the same principles of physical systems that govern robot installations.

4.1 Land-Based Vehicular Collisions

In contrast to the metrics for robots, most evaluations of land-based vehicular collisions are based on modelling, testing, and assessing the physics of actual collisions. Data from these crash tests are used exclusively to evaluate and improve the safety features of vehicles for the passengers inside. Such data, however, are used only sparingly in collision detection and avoidance—except for pre-processing and visualizing potential crash severities (e.g., [53]) and injury criteria (e.g., [54]). Also considered are the human factors such as health and fatigue that play roles in collisions involving human-operated vehicles. A review of the social factors that both promote accidents and result in the adoption of new vehicular safety systems provides the basis for such considerations [55].

Automobile manufacturers have made considerable progress in integrating sensor-based collision detection and collision avoidance systems into their products. As will be discussed shortly, many common forms of these systems either provide warnings to the driver or take partial control over the car's cruise control. The driver-warning systems signal the car's operator of a potential collision, while intelligent cruise control causes the car to slow automatically when a potential collision is detected.

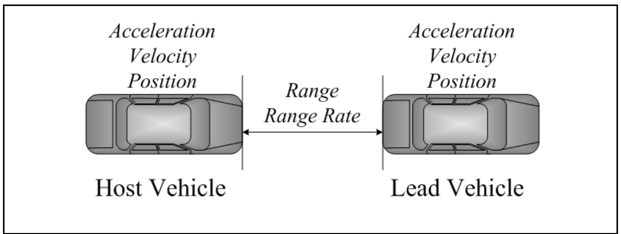

Figure 6. An illustration of some variables of interest in motor vehicle forward-collision warning systems. The raw data of the forward (lead) and following (host) vehicles are evaluated by a variety of safety systems when making braking decisions.

It is difficult to identify which algorithms perform better or more reliably than others without a common set of metrics. To address this, *time headway margins* [56] are defined to separate resulting behaviours into *safe* and *threatening* state classifications. The distinction is based on collected velocity, braking, and range data from manual tests utilizing lead and host vehicles. These data are then used to measure the percentage of time a given algorithm spent in each state (Figure 6). This method is validated based on tests involving several commercial and research systems [57-60]. Each of these systems provides braking logic [57-59] or driver warnings [58-60] based on metrics such as braking range [57], reaction time [58], time-to-impact [59], and braking time [60]. Moreover, each of these systems takes as inputs the velocities of the lead and host vehicles [57-60], relative rate of approach [57-60], relative distance [59, 60], and relative accelerations [59], and host vehicle kinematics [60].

The common, measurable factors utilized in the algorithms mentioned above are used in a number of additional collision detection, warning, and avoidance algorithms. For instance, a framework for collision avoidance decision-making in [61] selects from different reaction scenarios based on time-to-collision calculations. Meanwhile, the system described in [62] uses the lead and host vehicle velocities to determine how much time remains for the driver or the control system to avoid a rear-end collision with a lead vehicle. Other approaches exploit information and models not measurable at the time of a potential collision incident. For instance, the system described in [63] extends the methods of [57-60] to account for human factors, manoeuvres of adaptive cruise control, and the performances of previous systems.

Due to the pervasiveness of ground vehicles in modern society, new systems supporting collision detection, warning, and avoidance continue to be developed and deployed. For example, many vehicles are now equipped with rear-facing cameras and range sensors to give warnings of obstacles directly behind a car. Other common systems monitor traffic intersections for safety evaluations (e.g., [64]), or automatically park cars based on distance-measuring sensors and vehicle kinematics (e.g., [65-67]).

4.2 Aircraft Collisions

The safety of aircraft traffic is also centred upon minimum distance-separation metrics [68]. Given the high speeds and nonholonomic nature of aircraft motions, it is necessary that these metrics be thoroughly tested to better understand and minimize risks. For instance, the airspace evaluation equation in [69] calculates a probability of collision between two converging aircraft based on a benchmark probability that accounts for situational difficulty and operator inattention. In contrast, the probability of collision in [70] is a function of the horizontal, vertical, and lateral overlap probabilities. These overlap probabilities are based on the aircrafts' dimensions, nominal separation distances, and relative vertical velocities.

Two related survey papers identify a number of metrics used in aircraft collision warning and avoidance [71, 72]. Most of these metrics are based on separation measurements and calculations such as *predicted miss distance*, *range*, and *predicted time to closest point of approach*. Many other metrics consider *probability of collision* calculations. Less directly tangible metrics include *computational cost*, *collision rate*, and *utility*. These surveys also serve to pinpoint a number of deficiencies of mitigating circumstances in the metrics that are normally considered during actual instances of free flight. Concepts such as uncertainty, acceptance and implementation, robustness and validation requirements, multiple collision-avoidance capacities, and coordination and computational requirements were not addressed in the literature reviewed.

4.3 Naval Collisions

Because of the impact maritime travel has on the world economy, the maintenance of its safety is a priority. It is because of this that naval collision detection and avoidance has also been a significant focus of study. In fact, entire infrastructures and measurement systems have been developed to enable the safe, directed traversal of vessels both in open waters and close to shore.

The development of these systems has been based on a large number of collision risk assessment studies. These studies take advantage of the two-dimensional representation of the naval region (Figure 7). Risks of collision are indicated through the utilization of the areas surrounding the ship(s) and any nearby obstacles. For example, the system developed by Goralski and Gold [73] uses static and dynamic kinetic voronoi diagrams to represent the environment for both distance representation and nearest-neighbour queries. Another method by Tam and Bucknall [74] classifies encounter types and collision-avoidance manoeuvres based on collision regulations [75]. This method also features a categorization of obstacles based on their heading with respect to the heading of the ship. Even though the regulations in [75] are written as precisely as possible, automating collision avoidance is difficult as the regulations are often reliant on human interpretation and common sense [76].

The means by which vessels represent collision detection and avoidance vary somewhat. Nevertheless, these methods are ultimately based on the same basic two principles. First, they maximize the passing distance between the vessel and any potential hazard. Second, they minimize the deviation from the original intended route. The method proposed by Yongqiang and Chen [77], for instance, focuses the optimization of ship control for collision avoidance on a weighted fitness function that balances these two principles. More complex approaches take into account the sometimes-considerable effects of motion on the water surface. For example Shtay and Gharib [78] use models of the inertial effects on steering to train fuzzy models for collision avoidance. In contrast, the system described by Bandyophadyay, Sarcione, and Hover [79] considers other environmental factors such as tidal currents and waves into the collision detection and avoidance algorithms.

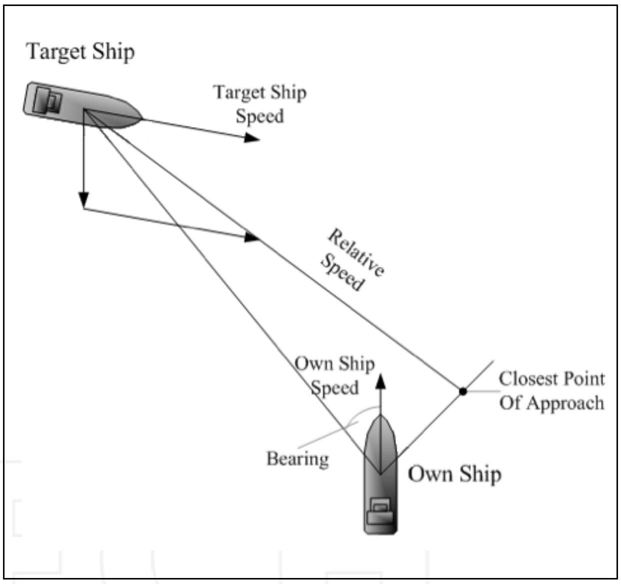

Figure 7. The meeting situation between ships approaching one another. Collision-avoidance steps are taken only if passing distances are too small for safe passage.

Sailing vessels present a unique problem because they are not self-powered. As such, they cannot directly navigate at will in any given direction. Research trends focus on collision detection and avoidance that use reactive steering to minimize directional changes while maintaining positive motion toward a goal location (e.g., [80, 81]).

5. Simulation and Graphics Collisions

Robot collision evaluation is tightly linked with the fields of computer graphics, machine vision, and simulation. Real-world testing of prototype robot systems and control algorithms is subject to several mitigating constraints. Such constraints include prototyping, costs, time, and danger. Initial trials, therefore, are typically carried out virtually. Similarly, geometrical representations replace real obstacles and real robot components for dynamic trajectory planning and collision testing. Parallels between the physical and virtual worlds can easily be drawn between the degree of object penetration and the severity of impact.

5.1 Mathematical Models of Collisions

We have described a number of algorithms for collision detection and measurement in this report, but we have not provided the technical details of these algorithms. Such details, which are an important factor in task-specific algorithm selection and implementation, have been reviewed in depth in other studies. One such survey by Lin and Gottschalk [82] focuses on techniques and algorithms for collision detection, specifically for geometric models and processing schemes for multiple objects. Another survey by Jiménez, Thomas, and Torras [83] provides a comparison of collision detection algorithms for different three-dimensional object representations. Yet another survey by Kockara *et al.* [84] provides a broad overview of common collision detection paradigms and their limitations.

5.1.1 Separation Metrics

The measurements and limitations of separations in simulations change as functions of the representation of object volumes. The separation of convex volumes defined by affinely independent sets of points (i.e., 'simplexes'), for example, is computed by the comparison of closest points between two convex hulls. The best-known example of simplex-based algorithms is the Enhanced Gilbert, Johnson and Keerthi (GJK, [85]) algorithm. Other algorithms rely on specific means of defining shapes to accommodate separation and collision detection. The most common of which include bounding volumes of spheres (e.g., [86-88]) and axis-aligned boxes (e.g., [89, 90]) given the simplicity of evaluating overlap.

In contrast, the separation of volumes defined by geometric features is measured by calculating the distances between elements like points (e.g., Voronoi Clip V-Clip, [91]), or other defining components like polyhedral faces, edges, and vertices (e.g., [92]). Algorithms for closed objects defined in image-space (e.g., [93]) and volume-space (e.g., [94]) utilize ray-casting methods to test for image space occlusions to detect collisions, but cannot measure separation distances.

There have been efforts to generalize the measurement of separation, and make the process independent of surface representation. One such effort by Bernabeu and Tomero [95] computes the *minimum translational distance* by first applying a Hough transform to determine if a given point is inside, outside, or on a spherical surface. The actual distance between said point and surface, however, was computed using GJK. Others have formulated guidelines for the functional inclusion of a myriad of bounding volume types. The implementation described by Johnson and Cohen [96] uses a framework based on geometric reasoning for minimum distance computations for several surface representations. The *lower-upper bound tree* framework mandated that each surface representation provides a set of common operations, such as bounding volume creation, lower bounds on distance computations, upper bounds on minimum distance computations, bounding column refinement, and methods to determine computational termination.

An important factor of collisions overlooked by graphics and simulations algorithms is the element of time. Metrics actually involving a time element do so not as a basis of collision metrics, but as an optimization tool. For instance, the systems in [97, 98] compute a time-to-collision for scheduling the order of collision testing. Another system described by Herzen, Barr, and Zatz [99] subdivides domains of time-varying object surfaces to define bounding regions on the scope of the sub-regions' ranges in virtual space for limiting collision queries.

5.1.2 Metrics of Collision Severity

One can readily see the relationship between collision severity and object surface penetration. A non-zero separation between surfaces means that there is no collision. In contrast, touching or penetration implies a collision has occurred. Measuring penetration, however, grows more computationally expensive and difficult as the complexity of the objects increases.

The penetration distance is the shortest relative translation of two or more objects that causes the objects to have no common interior points. The evaluation of this metric, however, is computationally expensive. To address this, *growth distances* [100] measure both surface separation and penetration by 'growing' two objects from fixed points until their interiors just touch. When the grown objects are larger than the original objects, this growth measures separation; when they are smaller, they measure penetration. Similarly, using two different object representations, *half-spaces* and *edge lists*, different classes of measures for the penetration of different representations of three-dimensional convex polyhedrons along a single axis can be defined [101]. These penetration measurements, when combined with a minimum Euclidean distance measure, can also be used to detect collisions.

A special note should be made regarding the Minkowski Difference [102],

$$C = A \ominus B, \qquad (2)$$

f two N-dimensional polygons, A and B, where $c \in C$: $c = a - b$, $a \in A$, $b \in B$ (Figure 8). If $\exists c_i$ such that $c_i = \{0, 0, \ldots, 0\}$, then it can be shown that A and B overlap in at least one point. This property of the Minkowski Difference has been exploited to great advantage by a substantial number of collision-detection algorithms including the GJK algorithm (e.g., [103-105]). The reason is simple: theoretically, it can be a useful metric for collision testing between two N-dimensional polygons. Computationally, however, it can be expensive, with exponential complexity for convex and general polyhedra. Additionally, the existence of $c = \{0, 0, \ldots, 0\}$ indicates only that the two polyhedra are touching, and does not specify the degree or direction of amount of penetration.

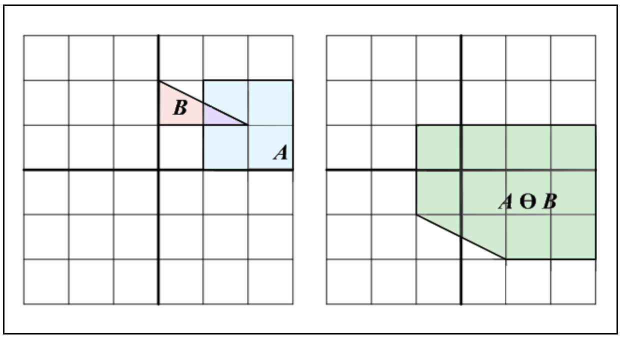

Figure 8. The Minkowski difference of two regions, *A* and *B* (left), results in a super-set area (right) that intersects the (0, 0) coordinate if *A* and *B* overlap

5.1.3 Algorithm Comparison Metrics

One of the largest factors limiting full utilization of collision detection algorithms is the lack of a common basis for comparison between the efficacy of collision and separation metrics. As we discussed earlier with the metrics for robotics, the process of comparing two or more virtual collision metrics consists entirely of comparing the computation costs of each algorithm. An example of such archetypal metrics is a cost function [106] for ray tracing bounding volumes:

$$T = N_V C_V + N_P C_P. \qquad (3)$$

Here, the total cost, T, is computed based on the costs for testing pairs of bounding volumes for overlap, C_V, and pairs of primitives for contact, C_P. These costs are scaled based on the number of bounding volumes, N_V, and primitives, N_P. Some researchers noted [107] that older methods were lacking in generality and were limited only to bounding volumes. As a result, they derived a new method for comparing collision detection for graphic-primitives algorithms. Rather than focusing only on computational cost, their method includes direct comparisons of three quantitative metrics (*performance*, *scalability*, *robustness)* and one qualitative metric *(ease of implementation).* They used the aforementioned GJK and V-Clip collision-detection algorithms to validate their method for the quantitative metrics. They readily admit, however, that there is no simple way to compute or validate their method for the qualitative metric.

5.2 Simulations and Virtual Agents

In many cases, simulations of physical agents either use or evaluate existing collision-avoidance and detection algorithms. However, as research in robotics turns toward collaborative human-robot interactions, efforts in collision avoidance will focus more on modelling virtual agents in complex scenarios. The method reported in [108], for example, develops and validates models of human collision avoidance. These models are based on existing multi-robot planning (e.g., as described in Section 3) and real-world biomechanical data of humans walking. Similarly, the system described in [109] uses agent-based vehicle guidance and collision-avoidance systems modelled after the perception and cognition of human drivers. The kinematic capabilities and statistical probabilities of motions of the human models provide inputs into the collision-modelling algorithms. As these systems are refined, it is expected that the characterization and evaluation of collisions will also evolve.

6. Current Trends and Next-Generation Systems

The studies in collision detection and avoidance have led to several unique and useful algorithms for separation measurement and assessment. Despite extensive research and increased automation, however, there is no single unified metric for measuring collision potential. Different application domains apply weights to different aspects of the separation problem. Land-based vehicles and mobile robots, for instance, maintain following distance measurements for automated braking problems. Aircraft safety systems, on the other hand, track the likelihood of mid-air collisions, and implement early evasive manoeuvres to minimize the possibility of impact.

Furthermore, the main goal for robotics and automation has been to maintain operator safety. Because the interactions between man and robot will likely increase, the importance of this goal will only grow. However, there is little in the current literature on which to judge the actual effectiveness of a given collision-avoidance algorithm. Nor is there any method to directly compare two different implementations to determine which is safer. Though comparative metrics exist for assessing the safety of specific systems, most are neither generalizable nor fully applicable across domains. Such metrics focus on equating safety with single-focus factors such as separation distance [110, 111], impact force [112-115], system configuration and velocity [116], inertia [117], and cost [118]. As a result, these metrics risk falling short of their full potential by not taking into account the lessons learned by other fields of study.

Modern automated systems are becoming increasingly complex. They feature multi-dimensional parameterized models of world events and statistical predictions of future occurrences. From the perspective of autonomous systems, we expect future collision-modelling systems to embody more hybridized and intelligent forms. We suggest the following practices will be likely candidate features for the next generation of performance metrics for collision modelling:

- Separation metrics, based on distance or time, compose a significant portion of the metrics for collision-detection and avoidance. However, a single metric is almost always insufficient for addressing the complexity of dynamic robot systems. This is because a single metric does not provide adequate information about the capabilities of the autonomous system or the obstacles. Metrics that combine separation metrics and statistical projections based on expected kinematics are likely to provide more robust collision-avoidance results.

- Computational complexity is becoming an increasingly common metric for direct comparison of algorithms. In such comparisons, the most computationally inexpensive algorithm always wins. Regardless, such simple algorithms may not be able to capture the full complexity of the operational environment. However, more computationally efficient, distributed, and agent-like approaches can reduce environmental uncertainties. These approaches will become more common with more advances in distributed computing and multi-core processor designs.
- Velocity scaling and pre-emptive steering constitute a majority holding of the collision-avoidance algorithms commercially available for land-based vehicular systems. Similar trends in robot safety systems are also apparent. Velocity scaling is far easier to implement, understand, and predict. As such, it is intuitively less likely to result in bad functional states than trajectory modulation. It is also much easier to recover from such bad states. Although active obstacle avoidance has long been singled out as the future of collision-avoidance mechanisms, robust and demonstrably safe implementations have yet to emerge.
- Identifying potentially hazardous states and computing the probability of collision have intrinsic value. Moreover, they may be useful in identifying key areas for directed risk assessments, which are necessary to improve safety performance. They are not likely, however, to be useful for guaranteeing safety in shared environments.
- Separation and penetration of virtual bounding regions are strong candidates for use in post-process collision evaluations. Through these, one can measure and model the severity of failures to provide adequate separation between robots and obstacles. Existing methods are computationally expensive, however, and are impractical for real-time evaluation for complex environments. As evaluative tools, however, penetration metrics can aid in the qualification of results of algorithm parameter tuning, and serve as a basis for system verification and validation.
- Human-machine interactions are on the rise in many domains. Human factors such as situational awareness, focus of attention, intention models, and biomechanical limitations are commonly integrated into automation safety systems. Models of human collision avoidance have been developed and validated based only on simple planning algorithms and existing biomechanical data [108]. Developing richer models of the interactions and hazards inherent in human-occupied environments will only improve the safety of autonomous systems.
- New metrics and algorithms for robot safety systems will continue to appear in the future. As the strict separation of humans and robots in various domains dissolves, the need for standardized test methods will increase. These test methods will provide the technical foundation for comparing two or more safety algorithms. Additionally, the development of, and support for, standardized uses cases and test and evaluation configurations to identify the dynamics of the collaborative robot-human tasks will also be necessary.

7. References

[1] Sugimoto N (1977) Safety Engineering on Industrial Robots and their Draft Standard Safety Requirements. Proc. 7th int. symp. ind. rob.: 461-470.

[2] Jiang B.C, Gainer C.A (1987) A Cause-and-Effect Analysis of Robot Accidents. J. occup. accid. 9(1): 27-45.

[3] Malm T, Viitaniemi J, Latokartano J, *et al.* (2010) Safety of Interactive Robotics – Learning from Accidents. Int. j. soc. rob. 2: 221-227.

[4] Rehg J (1985) Introduction to Robotics: A Systems Approach. Prentice-Hall, Inc., New Jersey.

[5] Fryman J, Matthias B (2012) Safety of industrial robots: From conventional to collaborative applications. Proc. ROBOTIK: 1-5.

[6] Zheng Y, Hemami H (1985) Mathematical Modeling of a Robot Collision with its Environment. J. rob. syst. 2(3): 289-307.

[7] Bicchi A, Tonietti G. (2004) Fast and "Soft-Arm" Tactics. IEEE rob. & autom. mag. 11(2): 22-33.

[8] Haddadin S, Albu-Schaeffer A, Hirzinger G (2008) The Role of the Robot Mass and Velocity in Physical Human-Robot Interaction - Part 1: Non-Constrained Blunt Impacts. Proc. IEEE int. conf. rob. autom.: 1331-1338.

[9] Ogorodnikova O (2009) How Safe the Human-Robot Coexistence Is? Theoretical Presentation. Acta polytech. Hung. 6(4): 51-74.

[10] Khatib O (1986) Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. Int. j. rob. res. 5(1): 90-98.

[11] Wickman T.S, Branicky M.S, Newman W.S (1993) Reflexive Collision Avoidance: A Generalized Approach. Proc. IEEE int. conf. rob. autom.: 31-36.

[12] Seraji H, Bon B (1999) Real-Time Collision Avoidance for Position-Controlled Manipulators. IEEE trans. rob. autom. 15(4): 670-677.

[13] Wen S, Zheng W, Zhu J, *et al.* (2012) Elman Fuzzy Adaptive Control for Obstacle Avoidance of Mobile Robots Using Hybrid Force/Position Incorporation. IEEE trans. syst., man, cybern., part C: appl. and rev. 42(4): 603-608.

[14] Hosoda K, Sakamoto K, Asada M (1995) Trajectory Generation for Obstacle Avoidance of Uncalibrated Stereo Visual Servoing without 3D Reconstruction. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 29-34.

[15] Boddy C.L, Taylor J.D (1993) Whole-Arm Reactive Collision Avoidance Control of Kinematically Redundant Manipulators. Proc. IEEE int. conf. rob. autom.: 382-387.

[16] Feddema J.T, Novak J.L (1994) Whole Arm Obstacle Avoidance for Teleoperated Robots. Proc. IEEE int. conf. rob. autom.: 3303-3309.

[17] Mao Z, Hsia T.C (1997) Obstacle Avoidance Inverse Kinematics Solution of Redundant Robots by Neural Networks. Robotica. 15(1): 3-10.

[18] Newman W.S, Hogan N (1987) High Speed Robot Control and Obstacle Avoidance Using Dynamic Potential Functions. Proc. IEEE int. conf. rob. autom.: 14-24.

[19] Park D, Hoffmann H, Pastor P, *et al.* (2008) Movement Reproduction and Obstacle Avoidance with Dynamic Movement Primitives and Potential Fields. Proc. 8th IEEE-RAS int. conf. humanoid rob.: 91-98.

[20] Liu H, Deng X, Zha H (2005) A Planning Method for Safe Interaction Between Human Arms and Robot Manipulators. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 2724-2730.

[21] Bosscher P, Hedman D (2009) Real-Time Collision Avoidance Algorithm for Robotic Manipulators. Proc. IEEE int. conf. tech. pract. rob. appl.: 113-122.

[22] De Luca A, Albu-Shaffer A, Haddadin S, *et al.* (2006) Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 1623-1630.

[23] Marvel J, Bostelman R (2013) Towards Mobile Manipulator Safety Standards. Proc. IEEE int. conf. rob. sensors env.: 31-36.

[24] Spencer A, Pryor M, Kapoor C, *et al.* (2008) Collision Avoidance Techniques for Tele-Operated and Autonomous Manipulators in Overlapping Workspaces. Proc. IEEE int. conf. rob. autom.: 2910-2915.

[25] Shackleford W, Norcross R, Marvel J, *et al.* (2012) Integrating Occlusion Monitoring into Human Tracking for Robot Speed and Separation Monitoring. Proc. workshop perform. metrics intell. syst.: 168-173.

[26] Kulic D, Croft E (2007) Pre-Collision Safety Strategies for Human-Robot Interaction. Auton. rob. 22(2): 149-164.

[27] Haddadin S, Albu-Schaeffer A, De Luca A, *et al.* (2008) Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 3356-3363.

[28] BGIA (2011) BG/BGIA Risk Assessment Recommendations According to Machinery Directive: Design of Workplaces with Collaborative Robots. U001/2009e October 2009 edition.

[29] International Organization of Standardization (2010) ISO 13855 Safety of Machinery – Positioning of Protective Equipment with Respect to the Approach Speeds of Parts of the Human Body.

[30] International Organization of Standardization (ISO) (2013) ISO Technical Specification 15066 Robots and Robotic Devices – Collaborative Robots (Draft).

[31] ISO (2011) ISO 10218 Robots and Robotic Devices – Safety Requirements – Part 1: Robots.

[32] ISO (2011) ISO 10218 Robots and Robotic Devices – Safety Requirements - Part 2: Robot Systems and Integration.

[33] Behnisch K (2008) Safe Collaboration with ABB Robots: Electronic Position Switch and SafeMove.

[34] Heiligensetzer P (2005) Safe Operation - Safe Handling. OTS-Workshop. FpF - Verein zur Foerderung produktionstechnischer Forschung.

[35] Kochan A (2006) Robots and Operators Work Hand in Hand. Ind. rob.: An int. j. 33(6): 422-424.

[36] Davies S. (2007) Watching Out for the Workers [Safety Workstations]. Manuf., IET. 86(4): 32-34.

[37] SICK, Inc. (2009) Industrial Safety Systems: Safety Laser Scanners for Increased Dynamism and Efficiency.

[38] Dorst L, Trovato K.I (1988) Optimal Path Planning by Cost Wave Propagation in Metric Configuration Space. Proc. SPIE: mobile robots III. 1007: 187-197.

[39] Steinfeld A, Fong T, Kaber D, *et al.* (2006) Common Metrics for Human-Robot Interaction. Proc. 1st ACM SIGCHI/SIGART conf. human-rob. interact.: 33-40.

[40] Ceballos N.D.M, Valencia J.A, Ospina N.L. (2010) Quantitative Performance Metrics for Mobile Robots Navigation. Mobile rob. navig.: 485-500.

[41] Evans J, Patron P, Smith B, *et al.* (2008) Design and Evaluation of a Reactive and Deliberative Collision Avoidance and Escape Architecture for Autonomous Robots. Auton. rob. 24(3): 247-266.

[42] American National Standards Institute (2012) ANSI/ITSDF B56.5-2012 Safety Standard for Driverless, Automatic Guided Industrial Vehicles and Automated Functions of Manned Industrial Vehicles.

[43] Amato N.M, Bayazit O.B, Dale L.K, *et al.* (1998) Choosing Good Distance Metrics and Local Planners for Probabilistic Roadmap Methods. Proc. IEEE int. conf. rob. autom.: 630-637.

[44] Alvarez J.C (1998) Planificacion del movimiento de vehiculos autonomos basada en sensores. Universidad de Oviedo, Oviedo, Espana.
[45] Sisbot E.A, Marin L.F, Alami R (2007) Spatial Reasoning for Human Robot Interaction. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 2281-2287.
[46] Toussaint M (2009) Robot Trajectory Optimization Using Approximate Inference. Proc. 26th ann. int. conf. mach. learn.: 1049-1056.
[47] Miura J, Negishi Y, Shirai Y (2006) Adaptive Robot Speed Control by Considering Map and Motion Uncertainty. Rob. auton. syst. 54(2): 110-117.
[48] Thrun S, Montemerlo M, Dahlkamp H, *et al.* (2006) Stanley: The Robot that Won the DARPA Grand Challenge. J. field rob. 23(9): 661-692.
[49] Bostelman R, Chang T, Hong T, *et al.* (2006) Unstructured Facility Navigation by Applying the NIST 4D/RCS Architecture. Proc. 3rd int. conf. cybern. inf. tech., syst. appl.
[50] Fulgenzi C, Tay C, Spalanzani A, *et al.* (2008) Probabilistic Navigation in Dynamic Environment Using Rapidly-Exploring Random Trees and Gaussian Processes. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 1056-1062.
[51] Singh S, Kapadia M, Faloutsos P, *et al.* (2009) SteerBench: A Benchmark Suite for Evaluating Steering Behavior. Comput. anim. virtual worlds. 20(5-6): 533-548.
[52] Haddadin S, Albu-Schaeffer A, Hirzinger G (2007) Safety Evaluation of Physical Human-Robot Interaction via Crash-Testing. Rob: Sci. syst. conf. RSS2007: 217-224.
[53] Desrosiers D, Birdsong C, Schuster P (2007) A Pre-Crash Simulator to Evaluate Vehicle Collision Prediction Algorithms. Proc. 5th IFAC symp. adv. autom. control.
[54] Alami R, Albu-Shaeffer A, Bicchi A, *et al.* (2006) Safe and Dependable Physical Human-Robot Interaction in Anthropic Domains: State of the Art and Challenges. Proc. IEEE/RSJ int. conf. intell. rob. syst. workshop: phys. human-rob. interact.
[55] Nmngani A, Akyurt M (2002) A Review of Vehicle Collision Avoidance Systems. Proc. 6th Saudi eng. conf., KFUPM: 413-428.
[56] Lee K, Peng H (2005) Evaluation of Automotive Forward Collision Warning and Collision Avoidance Algorithms. Veh. syst. dyn. 43:10 pp. 735-751.
[57] Doi A, Butsuen T, Niibe T, *et al.* (1994) Development of a Rear-End Collision Avoidance System with Automatic Brake Control. JSAE Rev. 15(4): 335-340.
[58] Fujita Y, Akuzawa K, Sato M (1995) Radar Brake System. Proc. ann. meetings ITS America: 95-101.
[59] Barber P, Clarke N (1998) Advanced Collision Warning Systems. IEE colloq. ind. autom. control: appl. automot. ind. 234: 2/1-2/9.
[60] Brunson S.J, Kyle E.M, Phamdo N.C, *et al.* (2002) Alert Algorithm Development Program. NHTSA Rear-End Collision Alert Algorithm. Final Report. DOT HS 809 526.
[61] Jansson J, Gustafsson F (2008) A Framework and Automotive Application of Collision Avoidance Decision Making. Automatica. 44(9): 2347-2351.
[62] Zhang Y, Antonsson E.K, Grote K (2006) A New Threat Assessment Measure for Collision Avoidance Systems. Proc. IEEE intell. transp. syst. conf.: 968-975.
[63] Ararat O, Kural E, Cuvenc B.A (2006) Development of a Collision Warning System for Adaptive Cruise Control Vehicles Using a Comparison Analysis of Recent Algorithms. Proc. intell. veh. symp.: 194-199.
[64] Kowshik H, Caveney D, Kumar P.R (2011) Provable Systemwide Safety in Intelligent Intersections. IEEE trans. vehic. tech. 60(3): 804-818.
[65] Paromtchik I.E, Laugier C (1996) Motion Generation and Control for Parking an Autonomous Vehicle. Proc. IEEE int. conf. rob. autom.: 3117-3122.
[66] Cuesta F, Gomez-Bravo F, Ollero A (2004) Parking Maneuvers of Industrial-Like Electrical Vehicles With and Without Trailer. IEEE trans. ind. electron. 51(2): 257-269.
[67] Zhao Y, Collins E.G Jr. (2005) Robust Automatic Parallel Parking in Tight Spaces via Fuzzy Logic. Rob. auton. syst. 51(2-3): 111-127.
[68] International Civil Aviation Organization (2010) Rules of the Air and Air Traffic Services, TP 14371E.
[69] Knecht W.R, Hancock P.A (1997) Parameterizing a Metric of Midair Collision Risk. Proc. 41st ann. meeting human factors ergon. soc.: 9-12.
[70] Yuling Q, Songchen H (2010) A Method to Calculate the Collision Risk on Air-Route. Proc. int. conf. manage. serv. sci.: 1-4.
[71] Kuchar J.K, Yang L.C (1997) Survey of Conflict Detection and Resolution Modeling Methods. Proc. AIAA guidance, navig, control conf.: 1388-1397.
[72] Palumbo N.F, Blauwkamp R.A, Lloyd J.M (2010) Modern Homing Missile Guidance Theory and Techniques. Johns Hopkins APL tech. digest. 29(1): 42-59.
[73] Goralski I.R, Gold C.M (2007) Maintaining the Spatial Relationships of Marine Vessels Using the Kinetic Voronoi Diagram. Int. symp. voronoi diagrams sci. eng.: 84-90.
[74] Tam C, Bucknall R (2010) Collision Risk Assessment for Ships. J. mar. sci. tech. 15(3): 257-270.
[75] U.S. Coast Guard Commandant. (1999) International Regulations for Prevention of Collisions at Sea, 1972 (72 COLREGS).
[76] Benjamin M.R, Curcio J.A (2004) COLREGS-Based Navigation of Autonomous Marine Vehicles. IEEE/OES auton. underwater veh.: 32-39.

[77] Yonqiang Z, Chen G (2010) The Coordinate Control of Ship Steering and Main Propulsion in Constrict Waters for Collision Avoidance. Proc. int. conf. logist. syst. intell. manage.: 568-572.
[78] Shtay A.D, Gharib W (2009) An Intelligent Control System for Ship Collision Avoidance. Int. j. eng. tech. 9(10): 36-41.
[79] Bandyophadyay T, Sarcione L, Hover F.S (2010) A Simple Reactive Obstacle Avoidance Algorithm and Its Application in Singapore Harbor. Springer tracts adv. rob. 62: 455-465.
[80] Sauze C, Neal M (2010) A Raycast Approach to Collision Avoidance in Sailing Robots. Proc. 3rd int. rob. sailing conf.
[81] Stelzer R, Jafarmadar K, Hassler H, *et al.* (2010) A Reactive Approach to Obstacle Avoidance in Autonomous Sailing. Proc. 3rd int. rob. sailing conf.
[82] Lin M.C, Gottschalk S (1998) Collision Detection Between Geometric Models: A Survey. Proc. IMA conf. math. surf.: 602-608.
[83] Jiménez P, Thomas F, Torras C (2001) 3D Collision Detection: A Survey. Comp. graphics. 25(2): 269-285.
[84] Kockara S, Halic T, Iqbal K, *et al.* (2007) Collision Detection: A Survey. Proc. IEEE int. conf. syst, man cybern.: 4046-4051.
[85] Gilbert E.G, Foo C (1990) Computing the Distance Between General Convex Objects in Three-Dimensional Space. IEEE trans. rob. autom. 6(1): 53-61.
[86] Hubbard P.M (1995) Collision Detection for Interactive Graphics Applications. IEEE trans. visual. comput. graphics. 1(3): 218-230.
[87] Palmer L, Grimsdale R (1995) Collision Detection for Animation Using Sphere-Trees. Comput. graphics forum. 14(2): 105-116.
[88] Larsen E, Gottschalk S, Lin M, Manocha D (1999) Fast Proximity Queries with Swept Sphere Volumes. Dept. of Computer Science, UNC, Chapel Hill, NC, Tech. Rep. TR99-018.
[89] Beckmann N, Kriegel H.P, Schneider R, Seeger B (1990) The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. Proc. ACM SIGMOD int. conf. manage. data: 322-331.
[90] Sulaiman H.A, Othman, M.A, Ismail M.M, *et al*. (2013) Distance Computation Using Axis Aligned Bounding Box (AABB) Parallel Distribution of Dynamic Origin Point. Proc. int. conf. microelectron., commun. renewable energy: 1-6.
[91] Mirtich B (1998) V-Clip: Fast and Robust Polyhedral Collision Detection. ACM trans. graphics. 17(3): 177-208.
[92] Lin M.C, Canny J.F (1991) A Fast Algorithm for Incremental Distance Calculation. Proc. IEEE int. conf. rob. autom.: 1008-1014.
[93] Knott D (2003) CInDeR: Collision and Interference Detection in Real Time Using Graphics Hardware. The University of British Columbia.
[94] Guendelman E, Bridson R, Fedkiw R (2003) Nonconvex Rigid Bodies with Stacking. Proc. ACM siggraph.: 871-878.
[95] Bernabeu E.J, Ternero J (2002) Hough Transform for Distance Computation and Collision Avoidance. IEEE trans. rob. autom. 18(3): 393-398.
[96] Johnson D.E, Cohen E (1998) A Framework for Efficient Minimum Distance Computations. Proc. IEEE int. conf. rob. autom.: 3678-3684.
[97] Lin M.C (1993) Efficient Collision Detection for Animation and Robotics. University of California, Berkeley.
[98] Lin M.C, Manocha D (1995) Faster Interference Detection Between Geometric Models. The visual computer. 11(10): 542-561.
[99] Von Herzen B, Barr A.H, Zatz H.R (1990) Geometric Collisions for Time-Dependent Parametric Surfaces. Comput. graphics. 24(4): 39-48.
[100] Ong C.J, Gilbert E.G (1996) Growth Distances: New Measures for Object Separation and Penetration. IEEE trans. rob. autom. 12(6): 888-903.
[101] Sridharan K, Keerthi S.S (2001) Computation of a Penetration Measure Between 3D Convex Polyhedral Objects for Collision Detection. J. rob. syst. 18(11): 623-631.
[102] Oks E, Sharir M (2006) Minkowski Sums of Monotone and General Simple Polygons. Discrete comput. geom. 35(2): 223-240.
[103] Li Z, Milenkovic V (1993) A Compaction Algorithm for Non-Convex Polygons and Its Applications. SCG Proc. 9th Ann. symp. comput. geom.: 153-162.
[104] Thomas F, Turnball C, Ros L, *et al.* (2000) Computing Signed Distances Between Free-Form Objects. Proc. IEEE int. conf. rob. autom.: 3713-3718.
[105] Gange G, Marriott K, Stuckey P.J (2008) Smooth Linear Approximation of Non-Overlap Constraint. Lect. notes comput. sci. 5228: 45-59.
[106] Gottschalk S, Lin M.C, Manocha D (1996) OBBTree: A Hierarchical Structure for Rapid Interference Detection. Proc. 23rd ann. conf. comput. graphics interact. tech.: 171-180.
[107] Levey E, Peters C, O'Sullivan C (1999) New Metrics for Evaluation of Collision Detection Techniques. Proc. Winter School Comput. Graphics: 140-146.
[108] Guy S.J, Lin M.C, Manocha D (2010) Modeling Collision Avoidance Behavior for Virtual Humans. Proc. 9th int. conf. auton. agents multiagent syst.: 575-582.
[109] Kong Z, Mettler, B (2013) Modeling Human Guidance Behavior Based on Patterns in Agent-Environment Interactions. IEEE trans. human-mach. syst. 43(4): 371-384.
[110] Chen M, Zalzala M.S (1997) A Genetic Approach to Motion Planning of Redundant Mobile Manipualtor Systems Considering Safety and Configuration. J. rob. syst. 14(7): 529-544.

[111] Trautman P, Krause A (2010) Unfreezing the Robot: Navigating in Dense, Interacting Crowds. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 797-803.

[112] Bicchi A, Peshkin M.A, Colgate J.E (2008) Safety for Physical Human-Robot Interaction. Springer handb. rob.: 1335-1348.

[113] Nokata M, Ikuta K, Ishii H (2002) Safety-Optimizing Method of Human-Care Robot Design and Control. Proc. IEEE int. conf. rob. autom.: 1991-1996.

[114] Ikuta K, Ishii H, Nokata M (2003) Safe Evaluation Method of Design and Control for Human-Care Robots. Int. j. rob. res. 22(5): 281-297.

[115] Haddadin S, Albu-Schaeffer A, Hirzinger G (2009) Requirements for Safe Robots: Measurements, Analysis and New Insights. Int. j. rob. res. 28(11-12): 1507-1527.

[116] Lacevic B, Rocco P (2010) Kinetostatic Danger Field - A Novel Safety Assessment for Human-Robot Interaction. Proc. IEEE/RSJ int. conf. intell. rob. syst.: 2169-2174.

[117] Kulic D, Croft E. (2004) Safe Planning for Human-Robot Interaction. Proc. IEEE int. conf. rob. autom.: 1882-1887.

[118] Althoff D, Kuffner J.J, Wollherr D, *et al.* (2012) Safety Assessment of Robot Trajectories for Navigation in Uncertain and Dynamic Environments. Auton. rob. 32(3): 285-302.