Master's Thesis

Robot motion planning in dynamic environment: A comparative study

Dharmin Bakaraniya

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Erwin Prassler

Dr. Cesar Lopez Martinez

January 2019

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

_____   _____
          Date   Dharmin Bakaraniya

Abstract

INCOMPLETE

Acknowledgements

INCOMPLETE

Contents

List of Figures

List of Tables

1

Introduction

- Robot motion planning can be considered solved for static environment[1].

- As the robot does not have the perfect information about the environment, the problem is considered NP-hard[2].

- Motion planning in dynamic environment has a lot more application compared to one in static environment as the robot can be deployed in real environment among humans and other robots. This need is increasing since the availability of cheap and easily accessible robots.

1.1 Motivation

By solving problem of motion planning in dynamic environment, we can ensure

- Safe environment for humans and for robots

- Cost effective transportation of goods

Additionally, by comparing different approaches on same constraints and parameters, we can form a well informed decision regarding the best approach for a given application.

1.2 Challenges and Difficulties

1.3 Problem Statement

- Motion planning in dynamic environment needs to perform the following task

  - Reach the desired target position

  - Avoid static and moving obstacle

  - Consider actuator constraints

  - Consider geometry of robot

- It needs to perform these tasks in *fast* and *efficient* manner.

- The *fast* nature of planner dictates that the planning of motion should be at least performed in real time for the robot to react to its environment.

- An approach is *efficient* if the planner

  - results in no collision

  - takes minimal time to reach goal position

  - does not require a large amount of sensor data

  - does not require a lot of computational resources

- Most of the approaches have been tested on a single type of robot kinematics (mostly circular, holonomic).

- Most of the approaches do not even address how the objects needs to be perceived. They do not address how errors from perception and control would influence the planner's efficiency.

- The state of the art approaches test their methods and algorithm with different parameters such as

  - kinematics of robot (holonomic, differential drive or car-type)

  - availability of map

  - motion of moving obstacles

- – accuracy of location information

- – amount of information available to robot about the obstacles

- This makes the task of determining the better approaches from the rest difficult and the result may be unreliable.

- This work will provide a brief overview of the current state of the art approaches for motion planning in time varying environment.

- Additionally, this work will provide a comparative evaluation on 3 approaches of motion planning tested on same parameters.

State of the Art

2.1 Related work

This work provides a comparative study of approaches in robot motion planning in dynamic environment. This field of research cannot be considered new in any respect. There are several existing survey article:

- Mohanan et al.[3] covers 101 research papers that were published between 1985 and 2016 in the field of motion planning in dynamic environment. This survey is very close to the current work in terms of publication date. They have considered most of the work in this field and presented it in different sections on the basis of school of thought. Even though this work covers many approaches, the comparison is lacking. The only comparison is within a school of thought and that too is quite shallow.Additionally, the authors do not provide proper reasoning behind any of their claims and neither does their work contain any experimental study.

- Hoy et al.[4] provides a survey for algorithms which provide collision free navigation for robots. This work is fairly detailed. This work cannot be considered very close to our work because it mainly covers obstacle avoidance comparison, but it does contain a section for dynamic obstacle avoidance. They have give quite a lot of importance of collision avoidance using boundary following. Even though it provides a comparison between main approaches

based on numerous criterion, it still does not evaluate these approaches on a standard uniform test. They have provided their comparison solely based on the information provided by the original research papers.

- Keshmiri et al.[2] provides a survey specifically for motion planning in dynamic environment. It covers all the approaches presented in research papers published between 1986 and 2008 totalling up to 150 papers. They have provided a comparison on how much contribution has been made in motion planning field based on different approach but regarding the actual approaches itself, only a summary of at most 23 sentences for each approach is provided. This is quite short for readers to get a clear picture. Also, they have not performed any test on the approaches. This survey was meant for just providing a survey for contribution in different school of thought.

- Surveys [5] and [6] are quite dated and does not cover any state of the art approaches in motion planning for dynamic environment.

Existing approaches generally provide critique and deficiencies on their previous works. These are generally helpful for the reader at the time but they can only provide a critique on the approaches that were published before them. Additionally, their critique are generally about the problem that they have solved in their work. Therefore, though these comparisons are helpful at that time for the reader to better understand the presented work, they might not be completely useful for an unbiased comparison.

2.2 Motion planning approaches

2.2.1 Velocity based

Dynamic window

- Fox et al.[7] proposed the original idea which was simply optimizing a function which balances robot's distance from goal (*heading*), distance from nearest obstacle (*dist*) and current velocity (*vel*).

$$G(v, w) = \sigma(\alpha \cdot heading(v, w) + \beta \cdot dist(v, w) + \gamma \cdot vel(v, w))$$

  Here α, β, γ and σ are constant that can be tuned according to need.

- This is a reactive approach, which means that the algorithms only thinks about the next iteration.

- It takes care of the kinematic and dynamic constraints by converting the current sensor readings in velocity space.

- Velocity space is a 2D space where X axis is angular velocity of the robot and Y axis is linear velocity.

- The robot is represented as a dot in this velocity space. The controls are bounded by the rectangle around this dot which represents the dynamic constraints of the robot.

- This approach, despite being robust, simple and fast did not work for dynamic environment. This method also had local minima problem.

- Later, Brock et al.[8] extended this approach for global path planning and for dynamic environments by combining it with NF1 algorithm. This eradicated the problem of local minima.

- The experiments did not contain dynamic objects.

- This approach is later extended by [9] where they provide a version of DWA which is convergent but they only consider static environment.

- The current best extension of dynamic window approach (time varying dynamic window (TVDW)) was proposed by [10]. This approach provides very robust obstacle avoidance method for dynamic environment.

- The authors have combined FD* path planning algorithm with dynamic window approach.

- This method uses moving cells for representing obstacles. This relieves the method from considering the shape of the obstacle.

- The obstacles are represented in the grid map at the location where the planner thinks they will collide. This helps the algorithm to calculate alternate trajectories efficiently.

- Another extension was proposed by [11], where they emphasise on the safety of the robot. They create a map with collision distance ($d_{collision}$) which takes into account any obstacles that are outside the field of view or occluded by other static objects. This approach provides a trajectories with longer travel time but the overall safety of the robot is far more compared to classical DW approach.

Velocity obstacle

- Originally developed by Fiorini et al.[12], this approach proposes to avoid obstacles by choosing velocity outside *collision cone*.

- The robot and the obstacles are considered circular. The robot contains the information about the position, orientation and the velocity of the obstacle.

- The obstacles are believed to travel at constant velocity and same orientation.

- The set of relative velocity of the robot and obstacle which result in collision is termed as a collision cone.

- By selecting an appropriate velocity outside the collision cone (avoidance velocity) of all the moving and static obstacles, the robot can ensure collision avoidance. Additionally, the choice of admissible velocity is restricted to the

velocity allowed by the kinematic and dynamic constraints. These are called *admissible avoidance velocities*.

- This approach unifies the representation for avoiding static and dynamic obstacles.

- This idea has since been transformed to incorporate many scenarios [13].

- One of the extensions, NLVO, was proposed by Shiller et. al. [14]. They extend the classical velocity obstacle approach that deals with obstacles moving linearly with constant velocity to propose *non linear velocity obstacles* that can deal with obstacles moving on arbitrary path at constant velocity.

- The concept of NLVO was used very successfully in [1], where they successfully implemented and test their extension of NLVO on a car-like robot. They have adapted the method such that it can deal with non circular robot and obstacles. As this can be computationally expensive, they used graphics library to speed up the collision calculation.

- A reactive method for VO was developed in [15]. They propose the idea of virtual plane which is obtained by transforming moving obstacles into static obstacles and transforming robot to a virtual robot. This plane is used to calculate the time to collision and collision position. By using this information, even a simple planner can be used to avoid collision. The authors state that any of the previous approaches that were used to solve static motion planning can be used here for desired outcome. They implemented and test their approach with multiple moving obstacles with acceleration.

- For multi robot systems, Van den berg et al.[16][17] have extended VO approach. The agents calculates the collision cone for another agents that are on the course of collision with it and chooses an admissible avoidance velocity that makes the changes only half of what is required. As all the agents are running the same algorithm and both of them will make half the effort, they will avoid collision without communicating with each other.

9

Velocity space

- An approach using velocity time space was proposed by [18][19]. If the obstacles move with constant velocity in same direction then the collision time and distance can be calculated. This information is mapped to velocity time space. This creates a surface in this 3D space which is used to choose a velocity for the robot so that it does not collide with this obstacle. The target is also mapped to velocity space.This space was named Dynamic velocity space (DVS). Simulated experiments were performed with 3 dynamic obstacles.

2.2.2 ICS based

- "Inevitable collision states (ICS) are states that will end up in a collision, no matter which trajectory is followed by the robot"[20]. This formal definition and a few important ground work was provided in [20]. This works on a state time space concept. Later, [21] declared that ICS is compliant with the safety criteria necessary for a robot to move safely in a dynamic environment.

- These safety criteria are as follows

    1. "The robot should consider its own dynamics

    2. The robot should consider the future behaviour of environment objects

    3. The robot system should reason over an infinite time horizon"[21]

- A partial motion planner combined with ICS-checker was proposed in [22]. They provide an additional property which proves that if the trajectory is collision free and the end state is not an ICS, then the whole trajectory is ICS-free. This allows them to calculate partial path and only check the final state for ICS. Their collision checker is rapidly exploring random tree based and only checks for braking trajectories of the robot. The algorithm was implemented on a car-like robot in simulation.

- Later [23] proposes ICS-AVOID, an ICS based collision avoidance approach. They use the ICS-checker proposed by [24] and combine it with *safe control*

kernel. This guarantees that the sampling output will always contain at least one safe trajectory. This enables their algorithm to go from one ICS-free state to another without failure. This is later compared with state of the art approaches based on dynamic window (TVDW[10]) and velocity obstacle (NLVO[1]). These experiments were performed in simulation and the information about the future trajectories of the obstacles (up to a limited time horizon) were provided to each of the algorithm. The results show that when same amount of information is provided, ICS-AVOID performs far better than TVDW and quite better than NLVO.

- An on-line planner based on the concept of NLVO and ICS was proposed by [25][26]. It is based on selecting an appropriate time horizon such that the robot will either pass ahead of the obstacle or stop before the obstacle. The NLVO till that carefully calculated time horizon guarantees that the robot never ends up in an ICS state. The approach was implemented on a robot in simulation and it was tested with 70 moving obstacles.

- The ICS method was extended to probabilistic domain by [27]. This method was later extended by the same authors in [28]. They introduce probabilistic collision state (PCS). So, instead of checking for collision, they are calculating the probability for collision. This probability is simply the minimum probability of a collision occurring by following a trajectory out of all the possible trajectory. This number determines the probability of that state being an ICS. The probabilities are calculated by considering the motion model of the obstacles and sampling from the probability density function of their resultant position at the end of some time interval. They have used only braking trajectories for collision checking since the infinite time horizon is infeasible. The authors have also considered collisions that occur among the obstacles excluding the robot. The authors have performed a string of experiments to check the effects of different parameters on PCS collision avoidance system on a car-like system.

- Since guaranteed ICS free navigation is impossible, Braking ICS was proposed[29]. The authors state that Braking ICS free navigation is the next best solution.

11

Their algorithm PASSAVOID is *provably passively safe* meaning that the robot will be at rest if a collision occurs. A braking ICS is a state such that no matter what braking trajectory is executed by the robot, it will still collide. They provide this algorithm under the assumption of limited field of view in an unknown dynamic environment. Simulated experiments were performed for a car like robot in the middle of 22 arbitrarily moving obstacles.

- An extension of PCS based navigation was developed by [30]. The robot constantly updates a probabilistic collision matrix based on the model of the environment. For each obstacle, sampling is done based on the previous model which considers its velocity, acceleration, direction of movement, etc. For trajectory calculation, authors have used time dependent Dijkstra algorithm. In exchange for a little higher collision probability, the authors get a huge computational speedup along with a shorter trajectory length by setting a minimum probability restriction. This means that if the collision probability of a cell is lower than a predefined threshold, then that cell is considered safe to traverse. The authors have performed extensive simulation experiments with trajectory generation with different parameters in order to explain the algorithm. For online planning, the authors have modified their algorithm so that the robot only follows the planned trajectory for certain period of time and alters the plan based on the newly available sensor data. The algorithm was tested on a real robot and pedestrians were used as obstacles. The robot was able to get from one position to another with very less collision probability.

2.2.3 Roadmap based

- Probabilistic roadmap approach was proposed by [31]. "A PRM planner samples the robots configuration space at random and retains the collision-free samples as milestones. It then tries to connect pairs of milestones with paths of pre-defined shape (typically straight-line segments in configuration space) and retains the collision-free connections as local paths. The result is an undirected graph, called a probabilistic roadmap, whose nodes are the

milestones and the edges are the local paths"[31]. This is a state space based approach. The sampling is done while considering kinematic and dynamic constraints imposed by the robot. The authors proved that as the number of sample increases, the probability of finding a path increases exponentially. Their approach is mainly for path planning, but they have provided with some additional techniques like sensing error counter measure, trajectory optimization, safe mode planning and on-the-fly replanning which makes their approach quite suitable for dynamic environment. They performed experiments to prove that a robot with any shape, kinematic and dynamic constraint was able to traverse. Additionally they also experimented with moving obstacle with random trajectory.

- An approach[32] was proposed for motion planning in dynamic environment where the global roadmap for static obstacles is already calculated. The authors provide a local and a global planning method which uses the previously given roadmap and only considers the dynamic obstacles. The problem is thus reduced to 1D state space. The local planner only considers going from one vertex to another in the roadmap, while the global planner employs multiple *probes* that individually run the local planner algorithm to find the global path. Experiments were performed for a 6 DOF robot in a 3D space in a simulation.

- In [33], the authors propose a method to calculate and recalculate path to the goal in dynamic environment. This approach is similar to the extension from [31]. The method creates an anytime plan from the current position to the goal position which is constantly updated based on the sensor information. Anytime D* algorithm is used to calculate the trajectory in state time space. Experiments were performed for a point sized robot in an simulated dynamic environment. The moving obstacles can change their direction randomly but this information is not available to the robot. The robot still only collides approx. 20% of the time.

2.2.4 Other approach

Nearness diagram

- A reactive navigation approach was proposed in [34]. They employ *situated activity* paradigm. The robot chooses from a set of 5 behaviour based on the sensor information. The sensor information is like vector field histogram. The robot behaves differently when there are object on its sides compared to where it is directly aligned to the goal. This algorithm was proved to not have local minima. The authors state that the proposed method only has one parameter to be selected compared to other approaches where numerous parameters determine the behavior of the robot. Experiments were performed on a real holonomic robot in a cluttered environment.

Potential field

- A potential field approach for dynamic environment was proposed in [35]. Here the target and obstacles are moving and they generate attractive and repulsive forces respectively. Problem of local minima was addressed. Calculations for holonomic robot and differential drive robot were provided. Experiments were performed in simulation and on real robots.

Biologically inspired

- A biologically inspired approach for differential drive robots was presented in [36]. Here the robot calculate the angle made by the obstacle in the field of sensing and calculates the linear and angular velocity that it must obey in order to avoid the obstacle. The robot follows two type of motion: avoidance motion and straight to goal motion. The robot switches between these modes to reach the target location. Experiments with multiple moving obstacles were performed in real and simulated environment. The approach was compared with velocity obstacle approach and the proposed method was faster to reach the target location.

2.3 Comparison

- There are different criteria for comparing approaches in the field of motion planning in dynamic environment.

- Apart from avoiding collision, the algorithms generally have at least one additional goal (generally reaching target in minimal time).

- This target position is generally provided to the robot as problem statement, though the target may not be in line of sight. In this case, the robot must plan a path from its current position to the target location. For this task, it will need a map of the environment.

- The approaches discussed above have different assumptions regarding this parameter of the experiment. While [8],[9],[10],[11],[22],[31],[33],[35] provide global connection to the goal, most of the approaches do not. They can, however, be extended by combining with a global path planner that provides checkpoints which in line of sight of the robot.

- Some approaches are only applicable to certain vehicle model types while most of them are applicable to every vehicle type.

- There are mainly 3 types of robot models, namely holonomic (can instantaneously move in any direction), unicycle (can move instantaneously in the direction it is facing and can turn instantaneously with certain angular acceleration) and bicycle (can instantaneously move in the direction it is facing and can turn within certain angle depending on the linear velocity)[4].

- Naturally, an approach which can be applied to all types of robot model is desirable. Though, in indoor environments, bicycle type robots might not be very useful because of their high turning radius. This makes them less maneuverable in narrow and cluttered spaces.

- Roadmap and ICS based approaches are not dependent on the model of the robot as they plan in state time space of the robot. This automatically takes care of kinodynamic constraints during planning phase.

Approach	Method	Vehicle type	Obstacle restriction	Obstacle shape	Experiment
DWA	[7]	holonomic	-	All	real
	[8]	holonomic	-	All	real
	[9]	holonomic	-	-	sim
	[10]	holonomic	cv, vd	All	sim
	[11]	holonomic	cv, cd	All	real
VO	[12]	holonomic+	cv, cd	circular	sim
	[13]	unicycle+	cv, cd	All	real
	[14]	holonomic+	cv, vd	circular	sim
	[1]	bicycle	cv, vd	All	sim
	[15]	unicycle	vv, vd	circular	sim
	[16]	holonomic	vv, vd	circular	sim
	[17]	holonomic	vv, vd	circular	sim
VS	[18][19]	unicycle+	cv, cd	All	sim
ICS	[22]	Bicycle+	vv, vd	All	sim
	[23]	holonomic+	cv, vd	circular	sim
	[24]	Bicycle+	cv, vd	circular	sim
	[25]	holonomic+	cv, vd	circular	sim
	[26]	holonomic+	cv, vd	circular	sim
	[28]	Bicycle+	vv, vd	circular	sim
	[29]	Bicycle+	vv, vd	All	sim
	[30]	Unicycle+	vv, vd	All	sim, real
Roadmap	[31]	All	vv, vd	All	sim, real
	[32]	All	-	circular	sim
	[33]	All	cv, vd	circular	sim
	[34]	holonomic	-	All	real
	[35]	holonomic, unicycle	cv, cd	All	sim, real
	[36]	unicycle	cv, cd	All	sim, real

Comparison of some of the state of the art approaches for motion planning in dynamic environment
+: can be applied theoretically to any vehicle type
cv: constant velocity, vv: varying velocity
cd: constant direction, vd: varying direction
sim: experiments performed in simulation
real: experiments performed on an actual robot
(Note: The experiments might have relaxed some assumptions taken for theoretical approach. The *Obstacle restriction* and *Obstacle shape* information is based on the information provided from experimental setup. If a literature provides multiple experiments with different relaxed assumption, then we consider the assumption which is most resembles the theory.)

16

- As stated before, for a robot motion planner must consider it own dynamic[21]. Most of the methods discussed above consider kinodynamic constraints.

- When the experiments are performed in a simulated environment, the information about the position and velocity vector of the obstacles as well as the position and heading of the robot are easily available. But, in real environment, these information is erroneous/uncertain in nature. The approaches which take this uncertainty into account are clearly more desirable.

- This is one of the only reason why ICS based approaches are not declared as the obvious choice. Even though they have the capability to reason over infinite time, they need perfect information about the environment to guarantee safety. On the opposite end of the spectrum are reactive approaches, which only reason about next iteration. What they lack in planning long term decision, they try to make it up on the frontier of lack of information.

- The approaches which are applicable in the environment where the obstacles are moving at variable velocity and in variable direction are more desirable because it resembles more closely to the real world. In real environment, a human might slow down to talk to someone or accelerate to overtake other people. All the state time space based approaches can theoretically deal with this constraint.

- Some approaches assume only circular obstacles for the ease of calculation or for the experiments, but most of the approaches can be extended to consider non circular moving obstacles by considering more than one circle for an obstacle[1]. Theoretically, all state time space based approaches can deal with obstacles of any shape.

- Clearly, state time space based approaches are better choice if the information about the environment is fairly accurate as they can deal with any robot model travelling in environment with obstacles of any shape travelling in variable direction at variable velocity. The possible problem might be the computation time required for them to plan a path in state time space after gathering the information.

3

Methodology

How you are planning to test/compare/evaluate your research. Criteria used.

3.1 Setup

3.2 Experimental Design

4

Solution

Your main contributions go here

4.1 Proposed algorithm

4.2 Implementation details

5

Evaluation

Implementation and measurements.

6

Results

6.1 Use case 1

Describe results and analyse them

6.2 Use case 2

6.3 Use case 3

7

Conclusions

7.1 Contributions

7.2 Lessons learned

7.3 Future work

A

Design Details

Your first appendix

B

Parameters

Your second chapter appendix

References

[1] F. Large, C. Laugier, and Z. Shiller, "Navigation among moving obstacles using the nlvo: Principles and applications to intelligent vehicles," *Autonomous Robots*, vol. 19, pp. 159–171, Sep 2005.

[2] S. Keshmiri and S. Payandeh, "An overview of mobile robotic agents motion planning in dynamic environments," in *Proceedings of the 14th IASTED International Conference, Robotics and Applications (RA20), MA, Boston*, pp. 152–159, 2009.

[3] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171 – 185, 2018.

[4] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, p. 463497, 2015.

[5] K. Fujimura, *Motion planning in dynamic environments*. Springer Science & Business Media, 1991.

[6] T. Tsubouchi, "Motion planning for mobile robots in a time-varying environment: A survey," *Journal of Robotics and Mechatronics*, vol. 8, no. 1, pp. 15–24, 1996.

[7] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[8] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, pp. 341–346, IEEE, 1999.

[9] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, 2005.

[10] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1986–1991, IEEE, 2007.

[11] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C. b. Moon, and J. B. Song, "Safe navigation of a mobile robot considering visibility of environment," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 3941–3950, Oct 2009.

[12] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[13] E. Prassler, J. Scholz, and P. Fiorini, "A robotics wheelchair for crowded public environment," *IEEE Robotics & Automation Magazine*, vol. 8, no. 1, pp. 38–45, 2001.

[14] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, pp. 3716–3721, IEEE, 2001.

[15] F. Belkhouche, "Reactive path planning in a dynamic environment," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 902–911, 2009.

[16] J. Van Den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1928–1935, IEEE, 2008.

References

[17] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[18] E. Owen and L. Montano, "Motion planning in dynamic environments using the velocity space," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2833–2838, Aug 2005.

[19] E. Owen and L. Montano, "A robocentric motion planner for dynamic environments using the velocity space," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4368–4374, Oct 2006.

[20] T. Fraichard and H. Asama, "Inevitable collision statesa step towards safer robots?," *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.

[21] T. Fraichard, "A short paper about motion safety," in *IEEE int. conf. on robotics and automation*, 2007.

[22] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2210–2215, IEEE, 2005.

[23] L. Martinez-Gomez and T. Fraichard, "Collision avoidance in dynamic environments: an ics-based solution and its comparative evaluation," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 100–105, IEEE, 2009.

[24] L. Martinez-Gomez and T. Fraichard, "An efficient and generic 2d inevitable collision state-checker," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 234–241, IEEE, 2008.

[25] O. Gal, Z. Shiller, and E. Rimon, "Efficient and safe on-line motion planning in dynamic environments," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 88–93, IEEE, 2009.

[26] Z. Shiller, O. Gal, T. Fraichard, *et al.*, "The nonlinear velocity obstacle revisited: The optimal time horizon," in *Workshop on guaranteeing safe navigation in dynamic environments. In IEEE international conference on robotics and automation*, 2010.

[27] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1492–1498, IEEE, 2010.

[28] D. Althoff, J. J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 285–302, 2012.

[29] S. Bouraine, T. Fraichard, and H. Salhi, "Provably safe navigation for mobile robots with limited field-of-views in dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 267–283, 2012.

[30] J. Hernandez-Aceituno, L. Acosta, and J. D. Pineiro, "Application of time dependent probabilistic collision state checkers in highly dynamic environments," *PLoS one*, vol. 10, no. 3, p. e0119930, 2015.

[31] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[32] J. P. Van Den Berg and M. H. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885–897, 2005.

[33] J. Van Den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 2366–2371, IEEE, 2006.

[34] J. Minguez and L. Montano, "Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 45–59, 2004.

[35] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous robots*, vol. 13, no. 3, pp. 207–222, 2002.

References

[36] A. V. Savkin and C. Wang, "A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles," *Robotica*, vol. 31, no. 6, pp. 993–1001, 2013.