Hybrid Collision Avoidance with Moving Obstacles

Yi Chai * **Vahid Hassani** **

* *Department of Marine Technology,*
Norwegian Univ. of Science and Technology,
Trondheim, Norway.

** *Department of Mechanical, Electronics and Chemical Engineering,*
Oslo Metropolitan University,
Oslo, Norway (e-mail: vahid.hassani@oslomet.no).

Abstract: This paper proposes a hybrid collision avoidance (COLAV) approach based on the integration of a global path planning algorithm and a reactive collision avoidance technique. This combination provides a robust path planning tool that can avoid collision with moving obstacles. Bézier curves are exploited as the basis for global path planning, while dynamic window (DW) algorithm is employed to search for optimal velocity pairs which ensure collision-free trajectory. In particular, the interface between the deliberate and reactive method is developed, enabling the vehicle to simultaneously track the generated global path towards the goal and avoid local collision. The performance and robustness of the proposed hybrid COLAV method is evaluated through numerical simulations.

Keywords: Bézier curve, path planning, dynamic window, collision avoidance

1. INTRODUCTION

A considerable amount of work has been done in the field of autonomous vehicles and collision avoidance (COLAV) over the past few decades. Autonomous path planning and collision avoidance are essential for Autonomous Surface Vehicles (ASV), navigating in unknown or partially known environment with static and moving obstacles in the vicinity of the vehicle. The hybrid COLAV architecture proposed in this article, decomposes the task into global path planning and local collision avoidance.

Reactive COLAV methods are widely used due to the low demand for computing capabilities. Velocity Obstacles method is one of those reactive COLAV approaches, intended for motion planning to avoid static and moving obstacles in the velocity space, Fiorini and Shiller (1998). Ge and Cui (2002) proposed a new potential field method for motion planning of mobile robots in a dynamic environment with moving target and obstacles. Additionally, dynamic window algorithm is one of the existing reactive COLAV approach, originally designed for robot with first order nonholonomic constraints, Fox et al. (1997). A modified DW algorithm presented in Eriksen et al. (2016), is adapted and tested for autonomous underwater vehicles (AUV) with second-order nonholonomic constraints.

Nevertheless, dynamic window algorithm suffers from many drawbacks, and the most significant one is high sensitivity to the local minima. Seder and Petrovic (2007)

proposes an improved dynamic window algorithm incorporated with a focused D* search algorithm, such that the vehicle is less likely to be trapped in a local minima. Furthermore, Serigstad et al. (2018) introduces a hybrid dynamic window approach, functions as an interface to any deliberate COLAV method which generates time parameterized trajectories, enabling vehicles to avoid local minima.

Motivated by the above considerations, in this paper, a hybrid COLAV architecture is presented, based on the combination of global pre-defined path generated by Bézier curves and dynamic window algorithm. Furthermore, interface between these two methods is developed, steering the vehicle to track the global path while avoiding both static and moving obstacles. Besides, detecting and recognizing obstacles, especially for moving obstacles, is generally difficult. Light Detection and Ranging Device (LIDAR) or other range-based sensors will be deployed on real vehicle to perceive relative position and map local terrain, facilitating further implementation of Simultaneous Localization and Mapping (SLAM).

The global path planning is carried out using a new generation of path planning that incorporates in its formulation the dynamics of the vehicles and extra data made available by on board sensors about obstacles and other vehicles in vicinity, Hassani and Lande (2018). Bézier Curves are used as the basis for generating a rich set of paths that determines spatial and temporal profile of the vehicles.

Using differential flatness property of the vehicle, we are able to reconstruct all the states of the vehicles during the maneuver. The calculated states are then used to assign a cost function to each path that reflects the dynamic capabilities of the vehicle on that path. Hence, the global path generated by Bézier curves takes the dynamics of vehicle into account in their formulations; see Hassani and Lande (2018).

The rest of the article is organized as follows. Section 2 summarizes the results in Hassani and Lande (2018) and presents a brief introduction to the global path generator used in this article. In section 3, a short description of the Dynamic Window algorithm is presented. Section 4 describes the key idea behind the proposed hybrid COLAV technique. The performance and robustness of the proposed hybrid COLAV algorithm is evaluated through several simulation scenarios in Section 5. Conclusions and suggestions for future research are summarized in Section 6.

2. GLOBAL PATH GENERATOR FOR FIXED OBSTACLES

This section summarized the results of Hassani and Lande (2018) in which, a class of Bézier curves is used to provide a rich class of potential paths. Using the flatness property of ASV, all the states and inputs of the ship along the path is computed from which a cost value can be assigned to each candidate path. Finally, an optimization problem is formulated that would give birth to a global path generator that would generate a path from point A to point B in presence of fixed obstacles. the calculated path satisfies dynamic limitations of the ASV such as required curvature, continuity, smoothness.

2.1 Bézier curve

The mathematical basis for the Bézier curve are the Bernstein polynomials, named after the Russian mathematician Sergei Natanovich Bernstein, Farin (2014). In 1912 the Bernstein polynomials were first introduced and published as a means to constructively prove the Weierstrass theorem. In other words, as the ability of polynomials to approximate any continuous function, to any desired accuracy over a given interval. The slow convergence rate and the technological challenges in the construction of the polynomials at the time of publication, led to the Bernstein polynomial basis being seldom used for several decades to come. Around the 1960s, independently, two French automobile engineers of different companies, started searching for ways of representing complex shapes, such as automobile bodies using digital computers. The motivation for finding a new way to represent free-form shapes at the time, was due to the expensive process of sculpting such shapes, which was done using clay. The first engineer concerned with this matter was Paul de Faget de Casteljau working for Citroën, who did his research in 1959. His findings lead to what is known as de Casteljau algorithm, a numerically stable method to evaluate Bézier curves.

De Casteljau work were only recorded in Citroën internal documents, and remained unknown to the rest of the world for a long time. His findings are however today, a great tool for handling Bézier curves, Farin (2014). The person who lends his name to the Bézier curves, and is principally responsible for making the curves so well known, is the engineer Pierre Étienne Bézier. Bézier worked at Renault, and published his ideas extensively during the 1960s and 1970s. Both Bézier and de Casteljau original formulations did not explicitly invoke the Bernstein basis, however the key features are unmistakably linked to it and today the Bernstein basis is a key part in the formulation, Farouki (2012).

A Bézier curve is defined by a set of control points $\boldsymbol{P}_i$ $(i = 0 \ldots n)$ for which n denotes the degree of the curve. The number of control points for a curve of degree n is $n + 1$, and the first and last control points will always be the end points of the curve. The intermediate points does not necessarily lay on the curve itself. The Bézier curve can be express on a general form as

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} B_i^n(t) \boldsymbol{P}_i \quad t \in [0, 1], \tag{1}$$

where t defines a normalized time variable and $B_i^n(t)$ denotes the blending functions of the Bézier curve, which are Bernstein polynomials defined as

$$B_i^n = \binom{n}{i}(1 - t)^{n-i} t^i, \quad i = 0, 1, 2 \ldots, n. \tag{2}$$

2.2 Differential flatness

A dynamic model of ASV is presented in Hassani and Lande (2018); furthermore, it is shown that the proposed model exhibits a differential flatness property; see Van Nieuwstadt and Murray (1998). A system is said to be differentially flat if one can find a set of outputs, equal in number to the number of inputs, such that one can express all states and inputs as functions of these outputs and their derivatives. This can be formulated mathematically for a nonlinear system, as follows. Consider a nonlinear system

$$\dot{x} = f(x, u) \quad x \in \mathbb{R}^n, \, u \in \mathbb{R}^m \tag{3}$$
$$y = h(x) \qquad y \in \mathbb{R}^m, \tag{4}$$

where x denotes the state vector, u denotes the control input vector and y denotes the tracking output vector.

Such a system is said to be differentially flat if there exist a vector $z \in \mathbb{R}^m$, known as the flat output, of the form

$$z = \zeta(x, u, \dot{u}, ..., u^{(r)}), \tag{5}$$

such that

$$x = \phi(y, \dot{y}, ..., y^{(q)}) \tag{6}$$
$$u = \alpha(y, \dot{y}, ..., y^{(q)}), \tag{7}$$

where ζ, ϕ and α are smooth functions.

3. DYNAMIC WINDOW ALGORITHM

Dynamic window method is a local reactive avoidance technique, searching for inputs implemented in the space of velocities. The main advantage of this approach is that it directly incorporates the dynamics of the vehicle, since the velocity space consists of translational velocity and rotational rate, which turn into surge speed u and yaw rate r for ASV, specifically. By adopting velocity space, the pruning of the search space enormously simplify the computational effort. Furthermore, the trajectory of the ASV can be approximated by a sequence of straight lines and circular arcs, and each arc is uniquely determined by the velocity tuple (u, r) with the radius $R = u/r$. For each velocity pair within the velocity space, the dynamic window algorithm is designed to predict the trajectory that velocity pair (u, r) might generate for the next n time intervals. Then, we only consider the first time interval and assume that velocity vector remains unchanged within the remaining n-1 time intervals. This assumption is based on the observation that search is automatically repeated after each time interval, while velocity will remain constant if there are no new commands.

3.1 Search Space

With the constraints imposed on the velocity space, the resulting search space is the intersection of three restricted velocity sets, namely, the set of possible velocities V_s, admissible velocities V_a and dynamic window V_d. The set of possible velocities is limited by the extreme value of the surge speed u and yaw rate r, which is defined as

$$V_s = \{(u, r) | u \in [0, u_{max}] \wedge r \in [-r_{max}, r_{max}]\}. \quad (8)$$

Due to the kinematic and dynamic constraints, the search space is reduced to a certain span around the current velocity, which only consists of reachable velocities within the next time interval. Thus, the dynamic window can be described as

$$V_d = \{(u, r) | u_c \in [u - \dot{u}_b \cdot \Delta t, u_c + \dot{u}_a \cdot \Delta t] \\ \wedge \quad \omega \in [r_c - \dot{r}_b \cdot \Delta t, r_c + \dot{r}_a \cdot \Delta t]\}, \quad (9)$$

where accelerations u_a and r_a are maximal translational and rotational accelerations, while u_b and r_b are maximal breakage decelerations. Terms u_c, r_c are current surge speed and yawrate.

The existence of obstacles in the vicinity imposes restrictions on the velocity pairs. The velocity is considered admissible if the vehicle is able to move to the next point before it hits the next obstacle on the predicted trajectory. As a consequence, the search space is reduced to a set of velocities that allow the vehicle to move without colliding with any obstacle, which can be defined as

$$V_a = \{(u, r) | u \leq \sqrt{2 \cdot dist(u, r) \cdot \dot{u}_b} \\ \wedge \quad r \leq \sqrt{2 \cdot dist(u, r) \cdot \dot{r}_b}\}, \quad (10)$$

where $dist(u, r)$ represents the distance to the closest obstacle on the corresponding trajectory.

3.2 Objective Function

Among those velocity pairs within the resulting search space V_r, velocity vector (u, r) is chosen to maximize a certain objective function, which consists of some criteria, like target heading, clearance and velocity.

$$G(u, r) = \alpha \cdot goal(u, r) + \beta \cdot dist(u, r) + \gamma \cdot vel(u, r) \\ s.t.(u, r) \in V_r, \quad (11)$$

where the terms goal(u, r), dist(u, r) and vel(u, r) are weighted by the factors α, β and γ. The terms involved in the objective function can be denoted as,

$$goal(u, r) = arccos(\frac{\overrightarrow{OA} \cdot \overrightarrow{OB}}{|\overrightarrow{OA}| \cdot |\overrightarrow{OB}|}), \quad (12)$$

$$dist(u, r) = \frac{1}{r_{min}}, \quad (13)$$

$$vel(u, r) = u_{max} - u_c. \quad (14)$$

Trajectory of the vehicle can be calculated with the velocity pairs (u, r), which implies the position is given at each time step. The term goal(u, r) is used to measure the progress towards the target, mathematically denoted as the angle between the vector pointing to goal and vector connecting start point and current position. r_{min} is referred to the distance from current position to the nearest obstacle, and the distance function dist(u, r) will reach a maximum value when obstacle occurs in the vicinity. The velocity term vel(u, r) is the difference value between maximal surge speed and the current one, which means vel(u, r) is exclusively dependent on surge speed u.

4. ADAPTIONS FOR HYBRID COLAV

As a reactive COLAV approach, dynamic window algorithm is restricted in many ways. The main drawback is that the vehicle may suffer from the risk of getting stuck in local minima and being unable to reach the goal, even though an exact path leading to the goal exists. Hence, it becomes necessary to employ a global path generated by Bézier curves, as a guidance for dynamic window algorithm. Based on the proposed deliberate and reactive COLAV methods, it's essential to develop the interface between global path planning and local collision avoidance algorithm.

4.1 Pure Pursuit Path Tracking Algorithm

To incorporate global pre-defined path generated by Bézier curves with dynamic window algorithm, a path tracking algorithm is obliged to be adopted. Pure pursuit path tracking algorithm (Coulter, 1992) has been widely used as a steering controller for autonomous vehicles. Yamasaki et al. (2009) proposes a robust path-following for UAV using pure pursuit guidance algorithm. Rankin et al. (1998) presents a review and evaluation of PID, pure pursuit, and weighted steering controller for an autonomous land vehicle.

The major objective of this method is to calculates curvatures enabling the vehicle to chase a moving target point that is some distance ahead of it on the pre-planned path. The chord length of the arc represents the look-ahead distance joining current position and goal point, and it's used when search for the next target point. The state of vehicle, including position and heading need to be updated after each search, and can be presented as

$$x_{i+1} = x_i + v cos\theta_i \Delta t, \tag{15}$$

$$y_{i+1} = y_i + v sin\theta_i \Delta t, \tag{16}$$

$$\theta_{i+1} = \theta_i + \omega \Delta t. \tag{17}$$

4.2 Interface between deliberate and reactive COLAV

Desired trajectory has been derived by employing pure pursuit path tracking algorithm, which can be used as a guidance for dynamic window. Hence, the interface between the deliberate and reactive method needs to be developed, enabling the vehicle to simultaneously track the generated global path towards the goal and avoid local collision. Based on the objective function presented in section 3, a new term corresponding to path alignment should be incorporated, denoted as $align(p_p, p_t)$, distance between point on pre-defined trajectory and current position determined by velocity pair (u, r) at each time step.

$$G(u, r) = \alpha \cdot goal(u, r) + \beta \cdot dist(u, r) + \\ \gamma \cdot vel(u, r) - \delta \cdot align(p_p, p_t) \tag{18}$$

These weight factors α, β, γ and δ determine how the hybrid COLAV favors trajectory keeping, collision avoidance or aligning with the global path. Each weight constant could be tuned to highlight the importance of each criteria.

In addition, since deliberate COLAV based on Bézier curves only ensures collision-free path with the presence of static obstacles, it is important to note that the gain in terms of obstacle clearance function β should be tuned bigger compared to other weighting factors, such that the vehicle is able to avoid when a moving obstacle emerges in the vicinity. As a consequence, the practical trajectory may deviate from the global path to a certain extent, presented in the following simulation section.

5. SIMULATION RESULTS

In this section, some simulation scenarios are presented to show the performance of hybrid COLAV method, including the ability of following global pre-planned path and collision avoidance. In the following scenarios, the hybrid algorithm manages to generate a trajectory from start point (0, 0) to goal point (1000, 1000) under different conditions of obstacles.

First Scenario:
As shown in Fig. 1, trajectory of vehicle aligns well with the global pre-defined path when merely considering static obstacles. The trajectory differs slightly when approaching close static obstacles, that indicates the prominent ability of tracking planned path. As shown in the simulations, when vehicle gets very close to a static obstacle, it allows

deviation from the global path to keep a fair distance from the obstacle while turning around the obstacle in the vicinity of position x = 400m. As the constraints of obstacle avoidance imposed on the deliberate method is less strict, yielding the path fairly close to the obstacle, which is considered as an unacceptable risky behaviour for reactive DW algorithm.
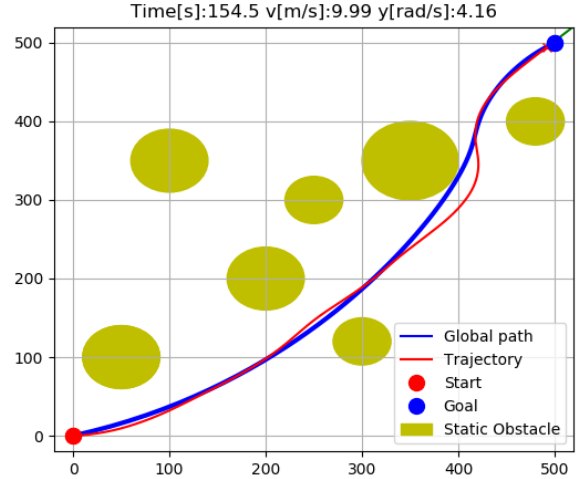

Fig. 1. DW trajectory with static obstacles

Second Scenario:
In the second scenario, moving obstacles with constant speed and heading are involved, with parameters shown in Table 1.

Table 1. Parameters of the moving obstacles

Parameter	Moving Obs 1	Moving Obs 2
Initial position	[200, 400]	[400, 100]
Heading angle	$-45°$	$120°$
Moving speed	3 m/s	3 m/s

The global path generated based on Bézier curve in conjunction with optimization formulation is inadequate to handle moving obstacles due to the less responsiveness to unexpected situation. As shown in Fig. 2, ASV ends up with colliding with the straight-line moving obstacle if it merely follows the global path. Fig. 3 validates that hybrid method incorporated with reactive DW algorithm is more powerful in the case of avoiding collision with moving obstacles, and gives us a clear explication that the vehicle is able to follow the global path when there is no threat, while significantly deviating from the global path in the middle section to stay clear of the moving obstacle and the vehicle then catches up with the path after entering safe region.
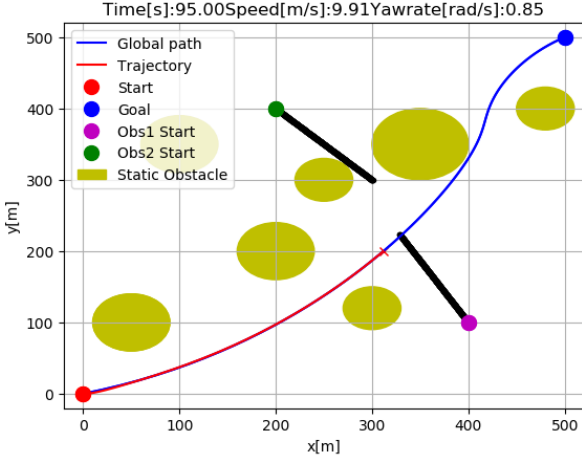
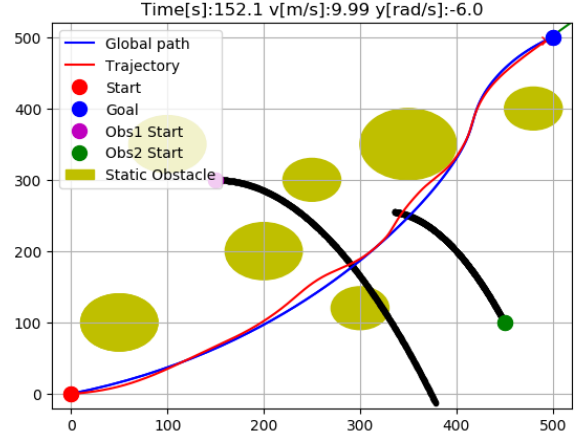Fig. 2. Global trajectory with straight-line moving obstacles

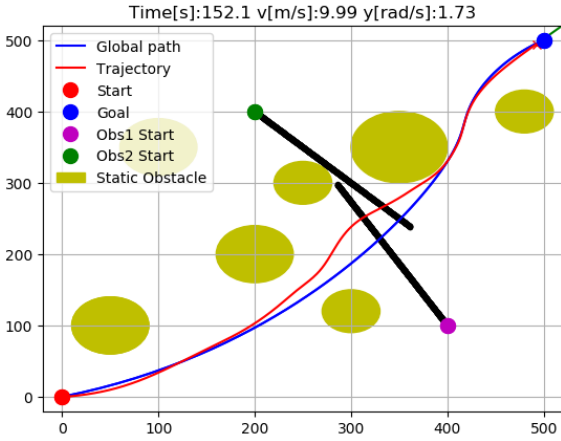Fig. 3. DW trajectory with straight-line moving obstacles

Third Scenario:
In this scenario, moving obstacles with varying heading and velocity leading to circular-arc trajectories, are taken into consideration, as shown in Table 2. As depicted in Fig. 4, the vehicle deviates from the global path to avoid the first moving obstacle emerging in the vicinity by changing the yaw rate, and it starts to catches up with the global path after entering the safe region. After tracking the path for a short distance, the occurrence of the second obstacle steers the vehicle off the track again. Further, the vehicle changes its heading to re-follow the path as soon as it gets rid of the obstacle.

Table 2. Parameters of the moving obstacles

Parameter	Moving Obs 1	Moving Obs 2
Initial position	[150, 300]	[450, 100]
Motion	y=-0.006x^2 + 1.8x +165	-0.01x^2 + 6.5x - 800
Moving speed	3 m/s	-1.5 m/s

Fig. 4. DW trajectory with circular-arc moving obstacles

Fourth Scenario:
In this scenario, an unknown dynamic obstacle with random trajectory is employed to evaluate the robustness of this hybrid method. The unpredictable and rapidly-varying motion trends have made the collision avoidance task more challenging, demanding for more responsive performance. As depicted in Fig. 5, the hybrid algorithm is still able to generate a collision-free trajectory almost coincided with the desired global path. When the vehicle is approaching the second moving obstacle, it is surrounded by static and moving obstacles on both sides. Compromise has been made to guarantee the safety by giving up keeping a fair distance to the static obstacle. As a consequence, the ASV takes the potential risk of running into the static obstacle to achieve collision avoidance with moving obstacle.
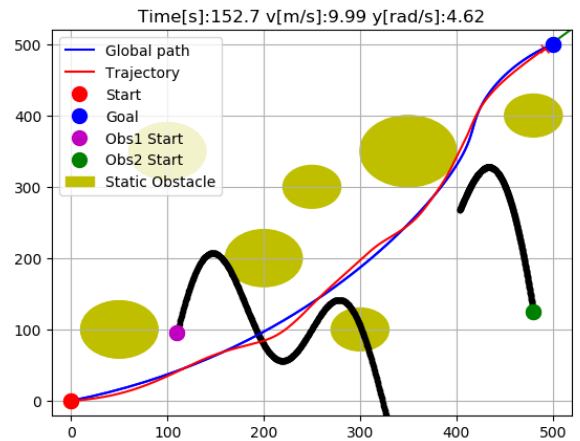

Fig. 5. DW trajectory with random moving obstacles

Fifth Scenario:
To evaluate the robustness of this COLAV algorithm, a Gaussian noise is added to the measured position and velocity of the obstacle. In other words, in this Scenario the COLAV algorithm only has access to noisy measurements

of position and speed of the moving obstacle. Due to the robustness of the hybrid COLAV algorithm, the vehicle is still capable of generating a feasible and collision-free trajectory until the noise is increased to an unaccepted value. And the tolerance limit to noisy measurement is tested by increasing the value of standard deviation σ. Fig.6 shows the result of trajectory including Gaussian noise with standard deviation $\sigma = 10$. The vehicle is still able to follow the global path and avoid random obstacle involving Gaussian noise with zero mean value and standard deviation $\sigma = 5, 10$, while it fails to proceed when the standard deviation is set to 15.
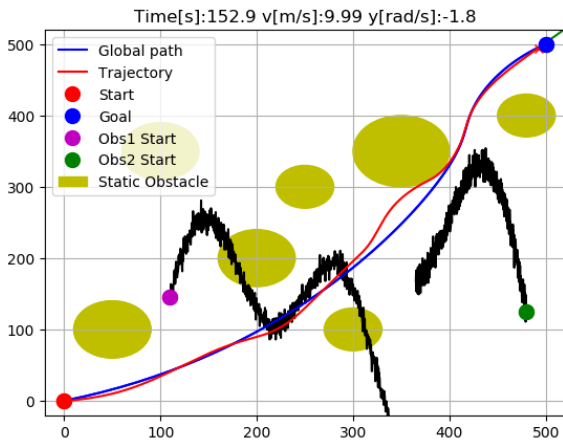

Fig. 6. DW trajectory with random moving obstacles including Gaussian noise

6. CONCLUSION

A hybrid COLAV method based on Bézier curves and dynamic window algorithm is introduced. Pure pursuit guidance is exploited to track the global path and extensively contribute to developing the interface between deliberate and reactive COLAV method. Furthermore, the feasibility and robustness of the algorithm is analysed regarding different scenarios through numerical simulations. The future work will include conforming with the International Regulations for Preventing Collisions At Sea (ColReg).

REFERENCES

Coulter, R.C. (1992). Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST.

Eriksen, B.O.H., Breivik, M., Pettersen, K.Y., and Wiig, M.S. (2016). A modified dynamic window algorithm for horizontal collision avoidance for auvs. In *2016 IEEE Conference on Control Applications (CCA)*, 499–506. IEEE.

Farin, G. (2014). *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier.

Farouki, R.T. (2012). The bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29(6), 379–419.

Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7), 760–772.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 23–33.

Ge, S.S. and Cui, Y.J. (2002). Dynamic motion planning for mobile robots using potential field method. *Autonomous robots*, 13(3), 207–222.

Hassani, V. and Lande, S.V. (2018). Path planning for marine vehicles using bezier curves. In *Proceedings of the 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*, volume 51, 305–310. Elsevier.

Rankin, A.L., Crane, C.D., and Armstrong, D.G. (1998). Evaluating a pid, pure pursuit, and weighted steering controller for an autonomous land vehicle. In *Mobile Robots XII*, volume 3210, 1–13. International Society for Optics and Photonics.

Seder, M. and Petrovic, I. (2007). Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 1986–1991. IEEE.

Serigstad, E., Eriksen, B.O.H., and Breivik, M. (2018). Hybrid collision avoidance for autonomous surface vehicles. *IFAC-PapersOnLine*, 51(29), 1–7.

Van Nieuwstadt, M.J. and Murray, R.M. (1998). Real-time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8(11), 995–1020.

Yamasaki, T., Takano, H., and Baba, Y. (2009). Robust path-following for uav using pure pursuit guidance. In *Aerial Vehicles*. IntechOpen.