Adaptive motion planning and path following for permanent resident biofouling prevention robot operating in fish farms

Martin Føre[1], Sverre Fjæra[1], Sveinung Johan Ohrem[2], Eleni Kelasidi[2,†],
Nina Bloecher[3], Herman Biørn Amundsen[1,2],
[1]*Dept. of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway*
[2]*Aquaculture Operations and Robotics, SINTEF Ocean AS, Trondheim, Norway*
[3]*Production biology and closed aquaculture systems, SINTEF Ocean AS, Trondheim, Norway*
† Corresponding author: eleni.kelasidi@sintef.no

***Abstract*—This paper presents a study where the potential of achieving autonomous net cleaning operations in commercial net pens is demonstrated through numerical simulations. Specifically, we investigated solutions for coping with the challenges that occur when obstacles appear in the path of an autonomous cleaning robot. Simulations were conducted using a net-crawling robot specifically designed for net cleaning purposes. The robot model was equipped with a modified version of the Elastic Band Method (EBM) enabling it to conduct adaptive motion planning. After proposed modifications and adaptations the method was first parameterised and verified through simulations where the robot had to avoid both static and dynamic obstacles along a simplistic path. The full functionality of the method was then explored in a simulation case study styled after commercial net cleaning operations: The robot was assigned an initial trajectory designed to ensure full coverage of a commercial scale net pens. Stationary and moving obstacles (e.g. fish, cables or other structural components, other vehicles) were then simulated by introducing regions of avoidance interfering with the trajectory initially planned, challenging the adaptive motion planning capabilities of the robot.**

The outcomes from the verification studies demonstrated that the use of the EBM enabled the robot to avoid obstacles appearing in or near its path without deviating significantly from the initial path. This was confirmed in the simulation case study results, implying that the original mission goals could be maintained (i.e. achieve maximum coverage during net cleaning). These results thus demonstrate that the Elastic Band Method is a good candidate for implementing adaptive motion planning for underwater vehicles tasked with complex operations in flexible and dynamic environments.

***Index Terms*—Underwater robotics, aquaculture, biofouling, grooming**

I. Introduction

Biofouling, the unwanted growth of organisms (e.g. hydroids, mussels, algae) at artificial substrates, is a major challenge for the marine salmon farming industry [1]. Fouling of pen nets is particularly problematic, as it may lead to various challenges such as reduced flow through the net (limiting water exchange in the net pen), and increased net weight. These factors may affect production efficiency, fish welfare and increase the risk of wear on the net, which in turn may result in unwanted incidents such as fish escapes. Maintaining an efficient biofouling mitigation regime is therefore a key element in the management of modern fish farms.

Biofouling mitigation accounts for 5-10% of the production cost in aquaculture [2], and most commonly relies on a combination of copper net coatings and the use of periodic manual in-situ pressure cleaning. This is conducted using rigs equipped with rotating discs expelling water jets that remove the organisms from the net. While the rigs were initially diver or crane operated, they are now mostly attached to Remotely Operate Vehicles (ROVs) or have inherent propulsion units to be steered along the net. While pressure washing efficiently removes biofouling from nets, it has side effects that may negatively impact the sustainability and safety of an aquaculture production operation.

Firstly, while pressure cleaning in itself does not cause damage to the net, it does lead to abrasion of the net coating, increasing the release of toxic copper particles into the environment [3]. Furthermore, the heavy equipment may lead to holes when used incorrectly [4]. Secondly, the general trend of increasing farm size and moving farms further offshore may increase HSE challenges related to operation of heavy equipment, especially as weather windows suitable for net cleaning operations are expected to decrease in rougher offshore conditions. Finally, the commonly conducted bi-weekly cleaning operation causes the removal of biofouling organisms from the net and their subsequent release into the net pen volume. This is a health risk to the fish as the cleaning waste particles have the potential to harm the fish in direct contact as well as through the transfer of pathogens associated with the biofouling [5].

These challenges and their potential ramifications for the fish production and HSE at a site have induced an increasing industrial interest in exploring if autonomous systems could enable more efficient, safe and cost-effective fouling management methods for commercial fish farms. A particularly attractive approach could be solutions where fouling levels are kept persistently low through continuous grooming of the

Research Council of Norway, grant no. 296392; "NetClean 24/7 - Tetherless robot for biofouling prevention and inspection in salmon farming"

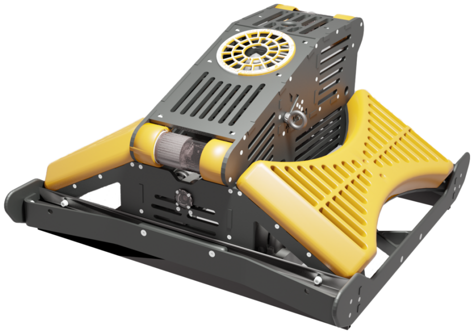


Fig. 1: The Remora net cleaning robot. Image courtesy of Mithal AS [8].

net by autonomous devices, thus preventing occlusion of the net as well as the release of potentially harmful cleaning waste [5]. In the shipping industry, similar approaches have been successfully trialled to protect ship hulls from fouling [6], [7]. Based on this background, Mithal AS has developed a new robot designed for continuous net grooming, called the Remora (shown in Fig. 1). The Remora is set up with two belts of hooks that allow it to attach to and crawl along the net. This renders the use of thrusters for maintaining a fixed distance to the net during inspection and cleaning obsolete. An autonomous solution based on such a concept is believed to have three concrete advantages over existing cleaning solutions and practices: 1) increasing fish health and welfare by securing ideal water flow conditions through continuous prevention of net occlusion, as well as avoidance of the release of potentially harmful net cleaning waste particles; 2) As grooming may be conducted on non-biocidal net coatings, it is likely to facilitate a reduction of the use of biocides in salmon farming; and 3) reducing personnel needs and hence HSE risks during cleaning operations. These benefits may become particularly important in the future due to ongoing trends in moving operations to more exposed locations with more challenging environmental conditions [9].

An initial study identifying the requirements for autonomous net grooming using underwater robots pointed out that a robust and functional control framework for the robot is essential [10]. This framework needs to contain the necessary components, not only to conduct continuous cleaning operations, but also to cope with unforeseen challenges that may arise during net cleaning operations. It is crucial that the system is able to handle such challenges without requiring human assistance for the solution to be capable of truly autonomous operations. An important element that needs to be addressed in this context is the system's ability to avoid collisions with obstacles. Relevant obstacles one might encounter when conducting a net cleaning operation include both static (e.g. ropes, other net pen components) and dynamic (e.g. fish, other vehicles, structural movements) elements that in some way hinder the free movement of the robot along the path originally planned to achieve the mission goals. Collisions with obstacles could thus result in an unsuccessful cleaning operation, and at worst cause damage to the robot, the fish, and/or the net.

An elegant method for achieving obstacle avoidance in robotics is to design motion planning methods that are able to dynamically re-plan the initial path. Motion planning can be done at different levels of complexity, the first and most basic method being offline global motion planning, where an entire mission is planned such that all mission objectives are achieved. Although this approach is very intuitive from a mission fulfilment perspective, and can be useful when changes are less likely to arise during the mission, it is less suitable for tasks in more dynamically changing environments. Examples of algorithms for global motion planning include the the A*-family, the RRT-family and Voronoi methods [11]–[13].

The ability to cope with unforeseen challenges can be improved by using local path planners [14]. These methods do not aspire to solve the entire mission planning problem, but are designed to plan a collision free path between two points that are relatively close to each other [15]. Examples of local path planners include specific local methods such as Ferman's spiral [16] and Dubins path [14], [17], but also methods that can be used for both local and global planning, such as the RRT-Family [13]. When combined with global path planners, these methods can thus be used to modify the path such that the vehicle avoids the obstacles. This has previously been done e.g. by using a Probabilistic Roadmap (PRM) as a local path planner modifying the global path obtained through cellular decomposition [18].

The main challenge of handling obstacle avoidance using local motion planning is that the local planner, when compensating for an obstacle, can become stuck in a local minimum from which it cannot find a feasible path to the end goal. Reactive motion planning approaches may thus serve as better options, as they typically ensure local feasibility and are sufficiently fast to handle the uncertainty of obstacles identified at short time horizons [19]. Like local methods, reactive methods generally need to be combined with a global path planner to acquire the initial trajectory that will ensure mission completion. Examples of reactive motion planning methods include the lazy PRM approach [20], Deformable Virtual Zone (DMV) [21], Braitenberg vehicles [22] and solutions based on bio-inspired neural networks [23].

Another example of reactive motion planning of particular interest is the Elastic Band Method (EBM), which initially was developed for robotic motion planning, but later used for automotive problems within lane changing and vehicle following [24], [25]. EBM optimises a global plan locally by minimising the path's length while taking static and moving obstacles into account. The path is optimised incrementally, meaning that the algorithm will continuously improve the path while the robot moves, resulting in a better output for longer paths.

This paper presents a simulation study on adaptations and applications of the Elastic Band Method to an underwater net cleaning robot for static and dynamic obstacle avoidance during net cleaning missions. After necessary modifications and adaptations of the EBM proposed in [25], the adaptive motion planning algorithm presented in this paper has been implemented and integrated in a simulation framework for underwater robotic applications in complex and dynamic environments such as aquaculture net pens. The proposed integrated control framework consists of the model of the net-crawling robot, the model of the net pen, the adaptive motion planning method, and the guidance and control system. Stationary and moving obstacles (e.g. fish, cables or other structural components, other vehicles) were simulated by introducing regions of avoidance interfering with the initially planned trajectory, challenging the adaptive motion planning capabilities of the robot. The full functionality of the control framework and the modified EBM algorithm were explored through extensive simulation case studies styled after commercial net cleaning operations. The simulations were carried out using a mathematical model of the net-crawling robot (e.g. Remora), along with a line-of-sight (LOS) guidance law and feedback linearising controllers. The obtained results demonstrated that the EBM is well-suited for autonomous net cleaning operations in dynamically changing environments such as aquaculture net pens.

Section II describes in detail the EBM and the necessary modifications for applications to net cleaning robots operating in a net pen. Section III briefly introduces the robot model, the guidance law and the controllers, as detailed descriptions of these are beyond the scope of this paper. The results from the simulation case studies are then presented and discussed. Section IV concludes the paper.

II. Modified Elastic Band Method

This section gives a brief summary of the EBM, originally proposed in [25], and presents the necessary modifications introduced in order to apply this method for robotic operations in aquaculture net pens.

A. Original algorithm

In basic terms, the EBM conceptualises the path as a band of partly overlapping bubbles of varying size. The volume within the bubbles is assumed to be unobstructed and thus eligible for the robot to move within. The size of the bubbles comprising a path can depend on the vehicles physical size and the maximum vehicle speed. Furthermore, the vehicles desired speed along the path can be determined based on the bubble size, e.g., when an obstacle appears in or near the path, or the planned path is otherwise adversely affected by external events and conditions, the EBM will compensate for this by reducing the sizes of the bubbles close to the disturbance, thereby also reducing the desired movement speed. A reduction in bubble size along the path may lead to a loss of the connection between consecutive bubbles in the sense that they no longer have sufficient overlap. To cope with this, the EBM will interject new bubbles in these areas, thus ensuring that the path is still coherent. The result is a path representation with varying width, resembling an elastic band that is stretched around one or several obstacles, which is the very background for the name of the method. Although very large environment changes can lead to the algorithm failing to deform to a collision-free path even if one exists, its flexibility increases the likelihood of achieving a feasible path. Chances of success can also be improved by implementing several elastic bands and choosing the best one of these based on scientific criteria [25].

The EBM algorithm can be divided into four distinct phases, each of which handles a key element of the method implementation. In *Phase 1*, the aim is to set up an initial path consisting of a sequence of partly overlapping bubbles, $\mathbf{B}$, each bubble having a position and radius, i.e., $\mathbf{B} = \begin{bmatrix}\mathbf{b}^{\mathrm{T}} & R^b\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^4$, where $\mathbf{b} = \begin{bmatrix}x, y, z\end{bmatrix}^{\mathrm{T}}$ contains the bubbles positions, and $R^b > 0$ is the bubble radius. Optimally, the initial path, $\mathbf{\Gamma}_{init}$ should be constructed as the shortest path from start to final destination considering no obstacles or other hindrances. This will lead to a sequence of bubbles that are largely evenly spaced along the path, and that are of similar size (i.e. R^b being close to equal across bubbles). To prevent the method from suggesting too fast movement (i.e. too large bubbles) or too slow speeds/non-converging paths (i.e. too small bubbles), the bubble size needs to be restricted. As such, the bubble size is chosen as

$$R^b_{min} \leq R^b \leq R^b_{max} \,, \tag{1}$$

where $R^b_{min} = 2D^v$, with $D^v > 0$ as the vehicles diameter and $R^b_{max} = |V^v_{max}|\Delta T$, where V^v_{max} is the vehicles maximum speed and $\Delta T > 0$ is the time interval for local path planning.

Phase 2 in the EBM is dedicated to deriving how the path should be deformed to avoid static and/or dynamic obstacles. This is done by finding the balance between two virtual forces emulating how a bubble (index i) is contracted onto the initial path by bubbles $i-1$ and $i+1$ ($\mathbf{F}^i_{int}$) and repulsed from the initial path due to external obstacles ($\mathbf{F}^i_{ext}$), respectively. $\mathbf{F}^i_{int}$ can be found as a geometric expression based on the relative positions of the bubbles:

$$\begin{aligned}\mathbf{F}^i_{int} = k_{int}\bigg(&\frac{\mathbf{b}_{i+1}-\mathbf{b}_i}{||\mathbf{b}_{i+1}-\mathbf{b}_i||}(||\mathbf{b}_{i+1}-\mathbf{b}_i|| - R^b_{min}) + \\ &\frac{\mathbf{b}_{i-1}-\mathbf{b}_i}{||\mathbf{b}_{i-1}-\mathbf{b}_i||}(||\mathbf{b}_{i-1}-\mathbf{b}_i|| - R^b_{min})\bigg)\end{aligned} \tag{2}$$

where k_{int} is a scalar gain used to adjust the contraction impact, and $\mathbf{b}_{i-1}$, $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$ are the positions of bubbles $i-1$, i and $i+1$, respectively. In sum (2) shows that two consecutive bubbles will induce a mutual contraction force between themselves that increases with increased intermittent distance, and that is limited downwards by the minimal bubble radius, R^b_{min}.

Since a scenario may contain several obstacles and some of these may even be moving, finding the virtual external repulsion force acting upon bubble i requires that all obstacles ($\mathbf{O}_j = [x_j, y_j, z_j, R^O_j], \forall j$) are considered. This is done by first finding the obstacles close enough to have an impact (i.e. where $||\mathbf{b}_i - \mathbf{O}_j|| < R_i^b + R_j^O$), and then computing the repulsion force for each of these obstacles:

$$\mathbf{f}_{ext}^j(\mathbf{b}_i) = k_{ext} \cdot e^{-D_a} \left(\frac{\mathbf{b}_i - \mathbf{O}_j}{||\mathbf{b}_i - \mathbf{O}_j||} \right) \tag{3}$$

where k_{ext} is a scalar value used to adjust the impact of the repulsion. The fading function e^{-D_a} is used to ensure that the repulsive effect of an obstacle decreases with distance, i.e.,

$$D_a = \max \left\{ ||\mathbf{b}_i - \mathbf{O}_i|| - D_{safe}, 0 \right\} \ . \tag{4}$$

Finding the total repulsive force ($\mathbf{F}_{ext}^i$) affecting bubble i is then a matter of summing up the contributions of all relevant obstacles:

$$\mathbf{F}_{ext}^i = \sum_{j=1}^{N} \mathbf{f}_{ext}^j(\mathbf{b}_i) \tag{5}$$

The internal and external forces are then summed to yield a total virtual force acting upon bubble i:

$$\mathbf{F}_{total}^i = \alpha \mathbf{F}_{int}^i + \beta \mathbf{F}_{ext}^i \tag{6}$$

where α and β are scalar weighting factors that are used to scale between the contributions of internal and external forces in the path deformation.

The resulting virtual force is then used to find an updated position for bubble i by multiplication with the step size for bubble modification (γ) and addition of the result to the present position of bubble i:

$$\mathbf{P}_{new}^i = \mathbf{P}_{old}^i + \gamma \mathbf{F}_{total}^i \tag{7}$$

The method iterates through these steps for each bubble i until the new position is clear from the obstacle or the deformation exceeds a predefined tolerance level, whereupon the size of the bubble is set equal to the clearance to the nearest obstacle as long as this distance conforms with (1). This process is conducted for all bubbles until the method results in an updated new path, $\mathbf{\Gamma}_{EB}$, with updated bubble positions and sizes that ensure avoidance of the obstacles.

In *Phase 3*, the aim is to ensure that the new path, $\mathbf{\Gamma}_{EB}$, is feasible and efficient in being used for navigation purposes, and mainly focuses on reorganising the bubbles comprising the path. The first step is to check if the deformation conducted in Phase 2 has led to some bubbles becoming redundant. A bubble is redundant if it is entirely within the radius of another bubble, i.e.,

$$|R_k - R_{k-1}| \geq ||\mathbf{b}_k - \mathbf{b}_{k-1}|| \tag{8}$$

or if it is within the overlapping area of two other bubbles, i.e.,

$$R_{k+1} + R_{k-1} > ||\mathbf{b}_k - \mathbf{b}_{k-1}|| + ||\mathbf{b}_{k+1} - \mathbf{b}_k|| \tag{9}$$

Bubbles identified as redundant by the algorithm are removed from $\mathbf{\Gamma}_{EB}$, thereby reducing the total number of bubbles.

The second task in Phase 3 is to ensure the connectivity of the path, i.e. to ensure that there are no gaps between consecutive bubbles in $\mathbf{\Gamma}_{EB}$. This is done by iterating through all bubbles and checking if the distance between each consecutive pair of bubbles is covered by their respective radii, i.e.:

$$R_i + R_{i-1} - d_{ol} < ||\mathbf{b}_i - \mathbf{b}_{i-1}|| \tag{10}$$

where d_{ol} is a parameter specifying the desired overlap between two consecutive bubbles. For cases where (10) holds, the method will insert a new bubble with suitable parameters to fill in the gap.

While the first three phases of the EBM are likely to result in the generation of a collision free path that also fulfils the main mission criteria, there is no guarantee that this path is in fact feasible for a vehicle to follow. *Phase 4* of the EBM is therefore aimed at transforming the trajectory to a format that is less likely to contain unrealistic jumps and kinks in position, heading or speed. The first step in achieving this is to smooth the path using e.g., the cubic B-spline method [26]. This will result in effectual filtering of unrealistic and unnecessary waypoints. However, this comes at a cost of increased inaccuracy as the waypoints originally derived by the elastic band deformation may then be exchanged with points that adhere less to the planned mission and more to ensuring obstacle avoidance. Smoothing with such methods should therefore be done carefully to avoid compromising the mission objectives and obstacle avoidance.

The final step in Phase 4 is to adjust the desired velocity of the vehicle based on the properties of the elastic band. This can be done by evaluating the equation:

$$\mathbf{V}_i = \frac{\mathbf{b}_{i+1} - \mathbf{b}_i}{T_c} \tag{11}$$

where T_c is a predefined time interval (typically the simulation time step) [26]. The speed of the vehicle when following the path $\mathbf{\Gamma}_{EB}$ will thus be determined by how close consecutive bubbles are to each other, leading to slower movement when bubbles are close (typically close to an obstacle or at tight corners) than when bubbles are far from each other (typically straight lines with no obstacles). It is also possible to scale the speed directly based on the bubble size, an approach that would give a relatively similar effect [24].

B. Modifications and adaptations of EBM

The following modifications to the EBM presented in [26] have been applied:

1) Updated redundancy check: In Phase 3 of the original EBM a situation may occur where a bubble is evaluated to be both redundant and ”missing” at the same time. This can happen if the desired overlap, d_{ol} is greater than the overlap between $\mathbf{b}_{k+1}$ and $\mathbf{b}_{k-1}$. A solution to this is to add the desired overlap to (9), i.e,.

$$R_{k+1} + R_{k-1} > ||\mathbf{b}_k - \mathbf{b}_{k-1}|| + ||\mathbf{b}_{k+1} - \mathbf{b}_k|| + d_{ol} \tag{12}$$

2) Decide new bubble radius: During Phase 2 of the EBM presented in [26] a bubble’s radius is decided based on the smallest clearance distance to all the obstacles. We propose that this process is repeated in Phase 3 as well, since the path may have been significantly altered when a bubble is removed or inserted during the initial steps of Phase 3. This modification is implemented to ensure collision free operations since we are conducting operations in a highly dynamic environment with static and moving obstacles.

3) Avoiding marginal improvements: In order to prevent the EBM from spending time trying to marginally improve the path, a condition is added to the algorithm. That is, if

$$||\mathbf{\Gamma}_{EB_i} - \mathbf{P}^i_{new}|| < s_c \tag{13}$$

where $\mathbf{\Gamma}_{EB_i}$ is the position of bubble i in the modified path, $\mathbf{P}^i_{new}$ is the proposed new position of bubble i and $s_c \geq 0$ is a constant determined by the user that determines whether the new proposed position should be accepted or rejected. By avoiding marginal improvements, we put priority on securing real-time implementation of the algorithm for operations in complex underwater environments.

4) Bubble size to determine desired velocity: In [26] the relative bubble position is used to determine the velocity of the robot, see (11). For our application the velocity is calculated by multiplying (11) by a factor, i.e.,

$$\mathbf{V}_i = \kappa \frac{\mathbf{b}_{i+1} - \mathbf{b}_i}{T_c} \ , \tag{14}$$

where $\kappa > 0$, in order determine a realistic desired velocity for the net-crawling robot. This is done since it is essential to respect the physical constraints of the robotic system related to the maximum operational velocity of the robot.

5) Using a guidance law: In the work presented in [24] and [26] the EBM produces a trajectory for the robot to follow. In this paper, we instead propose using a LOS guidance law to generate the desired course angle for the robot. This is done by considering the generated bubbles as waypoints along the path. This obviates the need for path smoothing with e.g. cubic B-spline as the look-ahead distance parameter of the LOS guidance law can be used to ensure smooth transitions to new waypoints. Furthermore, a reference model that smooths out the changes in desired heading and speed can be applied to avoid jumps and kinks in the reference signals.

6) Adaptation to cylindrical coordinates: While the mathematical model of the net-crawling robot, Remora, was programmed to navigate using cylindrical coordinates, the EBM method is designed for using Cartesian coordinate systems. This meant that the EBM method had to be expanded with capabilities to operate on cylinder coordinates to be applicable to simulations using the model of a net-crawling robot proposed in [27]. This was solved by adapting the coordinate conversion in (17).

III. Integrated Control Framework and Simulation Results

The integrated control framework proposed in this paper consisting of the model of the net-crawling robot, the model of the net pen, the adaptive motion planning method, and the guidance and control system was implemented in SINTEF's software platform FhSim (Figure 2). FhSim offers numerical models of several different sub-systems relevant for simulating aquaculture operations, including net pens, ropes/cables, underwater vehicles and fish [28]. Implementation into FhSim also enabled access to a built-in 3D-visualisation engine based on OGRE, allowing for more advanced visualisation of the simulations. In subsections below, the components of the integrated control framework will be introduced followed by the simulations results obtained in this paper.

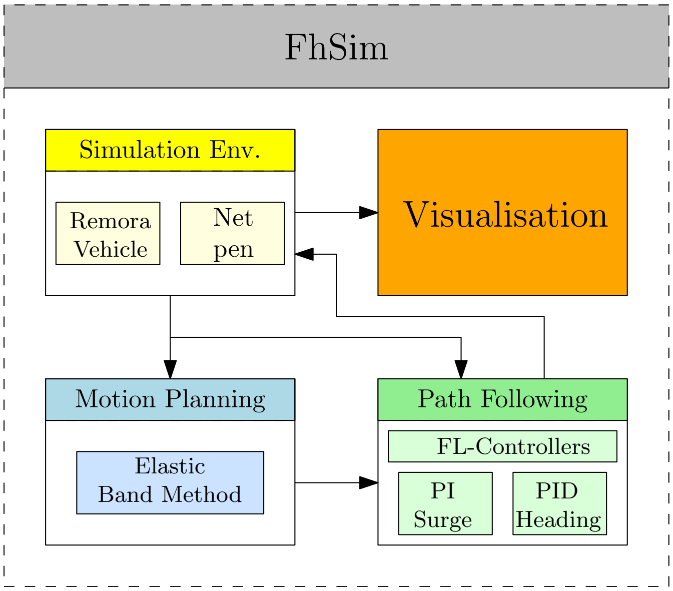


Fig. 2: Illustration of the integrated control framework.

A. Robot model

Simulations were conducted using a recently developed mathematical model of the net-crawling robot (e.g. Remora) [27]. Unlike a free-swimming ROV that operates in six degrees of freedom (DOFs), the Remora's design, with hooked belts interlinking with the net pen, enables it to crawl along the net, essentially meaning that it operates in a 3 DOF cylindrical coordinate system. The states defining the robots' movement are thus the azimuth angle (a_v) describing the position along the circumference of the pen, the robots' depth (D) and the heading angle(ψ). The equations of motion are defined in [27] as

$$\dot{\bar{\boldsymbol{\eta}}} = \boldsymbol{J}_c(\psi)\bar{\boldsymbol{\nu}} \tag{15}$$

$$\dot{\bar{\boldsymbol{\nu}}}_r = \boldsymbol{M}_c^{-1}\left(-\boldsymbol{D}_{l,c}\bar{\boldsymbol{\nu}}_r - \boldsymbol{D}_{n,c}(\bar{\boldsymbol{\nu}}_r)\bar{\boldsymbol{\nu}}_r + \boldsymbol{\tau}_c\right) \tag{16}$$

where $\bar{\boldsymbol{\eta}} = \begin{bmatrix} a_v & D & \psi \end{bmatrix}^{\mathrm{T}}$ contains the robots' position and heading angle, and $\bar{\boldsymbol{\nu}}_r = \begin{bmatrix} u_r & r \end{bmatrix}^{\mathrm{T}}$ contains the robots' relative surge speed, u_r, and yaw rate r. The matrices $\boldsymbol{M}_c$, $\boldsymbol{D}_{l,c}$ and $\boldsymbol{D}_{n,c}$ contains parameters for the robots' inertia, linear damping and nonlinear damping, respectively, while $\boldsymbol{\tau}_c$ contains the input force and moment. As the robots' position is defined in a cylindrical coordinate system, a conversion to the Cartesian North-East-Down system can be realised through

$$\begin{aligned} N &= r_c(t)\cos(a_v) \\ E &= r_c(t)\sin(a_v) \\ D &= D\,, \end{aligned} \tag{17}$$

where $r_c(t)$ is the radius of the net pen at the current position of the robot (and thus defined by the azimuth, a_v, and time as the net pen may deform due to external forces leading to a varying radius). See [27] for more information.

B. Guidance and control system

The reference heading was realised using a Line of Sight (LOS) guidance law [29, Ch. 10.3.2], which sought to minimise the cross-track error between the net-crawling robot and a virtual straight line between the position of the previous bubble and the next bubble in the elastic band. Furthermore, the desired speed of the robot was determined by how close consecutive bubbles were to each other. This is further elaborated later in this section.

To ensure that the surge speed and heading angle of the robot followed the reference values, a feedback linearisation controller was used [27]. This has the advantage of enabling the perfect cancellation of the nonlinear terms and that the origin of the closed-loop system became a Uniformly Globally Exponentially Stable (UGES) equilibrium point. As a consequency, it was possible to focus more closely on the particulars of the EBM method. However, feedback linearisation control can be challenging in physical experiments as the required assumptions, e.g., perfect parameter knowledge, are often not met in these situations. This may render the closed-loop system unstable. As such, model independent control concepts such as PI or PID controllers can be used when performing physical experiments.

C. Simulation case studies

1) Parameterisation and functional verification: The initial simulation case studies were designed to verify the basic functionality of the EBM, and establish ranges for some of the crucial parameters used by the method. This was done using an initial path that started at 5 m depth. The path then moved to the surface where it followed a half-circle with a radius of 25 m, resembling tracing the perimeter of a net pen. The path then moved to 15 m depth before commencing a similar half-circle motion, returning to the starting angular position, then moving directly upwards until the starting point was reached, see Fig. 4. Two static obstacles were inserted on the net plane, along the path, one at the surface and one at 15 m depth, to induce action from the adaptive motion planning method. The success criteria in these simulations were that the EBM should provide a path that 1) is sufficiently smooth for the robot to follow, 2) avoids all regions of avoidance (obstacles), 3) followed the initial path as well as possible while respecting 1) and 2).

TABLE I: Parameter configurations for the EBM tested in the parameterisation study.

Config#	k_{int}	k_{ext}	α	β
1	0.5	0.5	0.5	0.5
2	0.05	0.5	0.5	0.5
3	0.05	0.5	0.1	0.5
4	0.05	5	0.1	0.5
5	0.05	5	0.1	0.01
Final	0.1	3	0.25	0.75

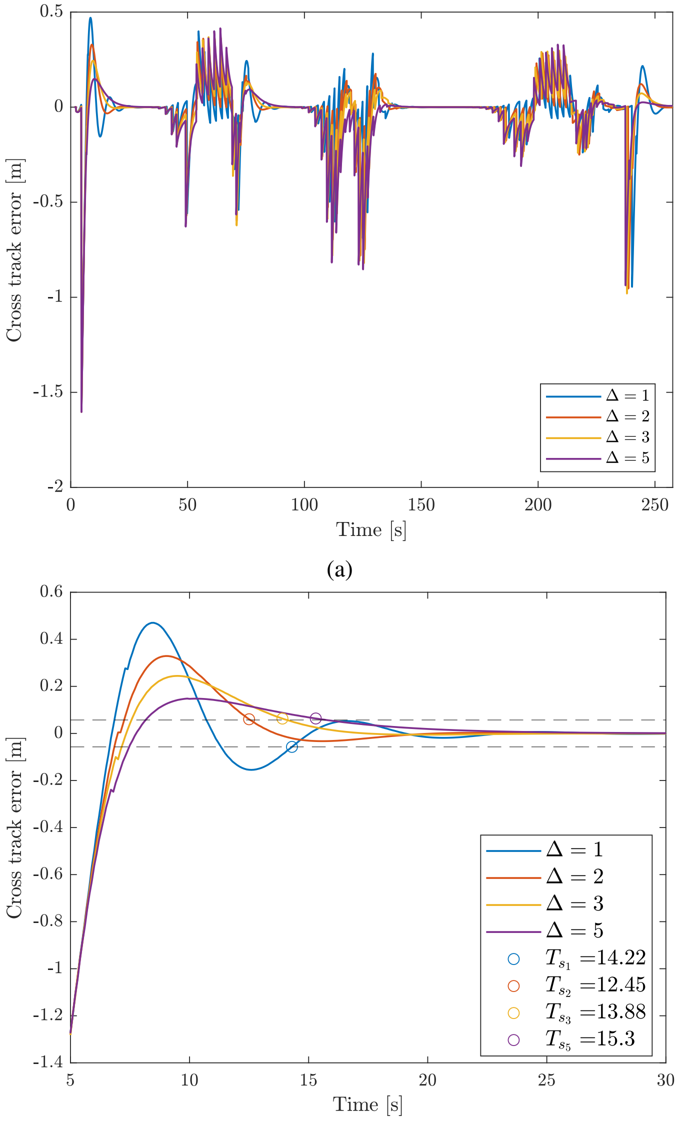
The simulations were performed using five different sets of values for the parameters k_{int}, k_{ext}, α and β (Table I), to identify the sensitivity of the EBM to changes in these particular values. In all simulations the bubble modification step size was set to $\gamma = 0.01$, the desired overlap to $d_{ol} = 0.25$ and the marginal improvement constant to $s_c = 0.1$. The feedback linearising surge speed controller had parameters $k_{p,u} = k_{i,u} = 4$ for the proportional and integral parts, respectively, while the feedback linearising heading controller had parameters $k_{p,\psi} = 5$, $k_{i,\psi} = 1$ and $k_{d,\psi} = 5$ for the proportional, integral and derivative parts, respectively. Both controllers were tuned using trial and error. The lookahead distance in the LOS guidance law was set to 3 m as this was found to give a satisfactory transient with an acceptable trade-off between settling time and overshoot (Figure 3).

The outcomes of the simulations with configurations 1-5 in Table I were then used to establish a final set of parameters which is provided in the last line of the table. This parameter set was used in the remainder of the simulations conducted in this study.

The simulation results clearly illustrated the effect of varying the parameters: Reducing k_{int} by a factor of 10 (configuration 1 to 2) led to the robot following the initial path more closely (Figure 4), while reducing α by a factor of 5 (configuration 2 to 3) resulted in a less smooth path. Moreover, increasing k_{ext} by a factor of 10 (configuration 3 to 4) led to the robot avoiding the region of avoidance with a slightly wider margin, while decreasing β by a factor of 50 led to a large reduction in the impact of the obstacles on the path, essentially making the robot unable to avoid the obstacle. The parameters labelled as *Final* were used in the remainder of the experiments, as this set of values led to the robot successfully avoiding the obstacles while not significantly deviating from the initial path.

The variations between the cases also illustrated that the parameter sensitivity of the method was as expected, with k_{int} and α scaling the propensity of the method to stick close to Γ_{init} and k_{ext} and β scaling the deformation caused by obstacle avoidance. While the interplay between these factors is what yields the final result, the sensitivity level appears to vary between them. For instance, while changing α with a factor of 5 elicited a response in the final Γ_{EB}, β had to be subjected to a 10 times larger change to have a similar effect. This may have ramifications for how the method should be tuned for a specific purpose in that larger modifications in behaviour are implemented by adjusting α, while β is only adjusted to obtain smaller changes.

Another simulation using the same path as in the param-



(a)

(b)

Fig. 3: a) Cross track error with different values of the lookahead distance Δ during the simulation presented in Fig. 4, with the final parameter set. b) Zoomed view of the cross track error with the settling time T_s visualized.

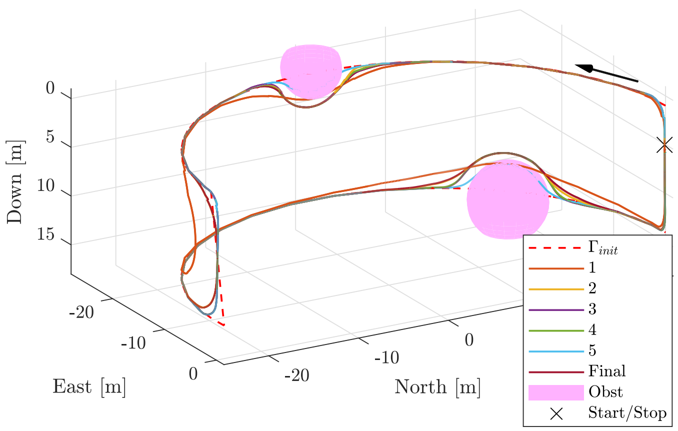


Fig. 4: Trajectories obtained when using different parameter configurations for the Elastic Band Method. Γ_{init} represents the initial path.

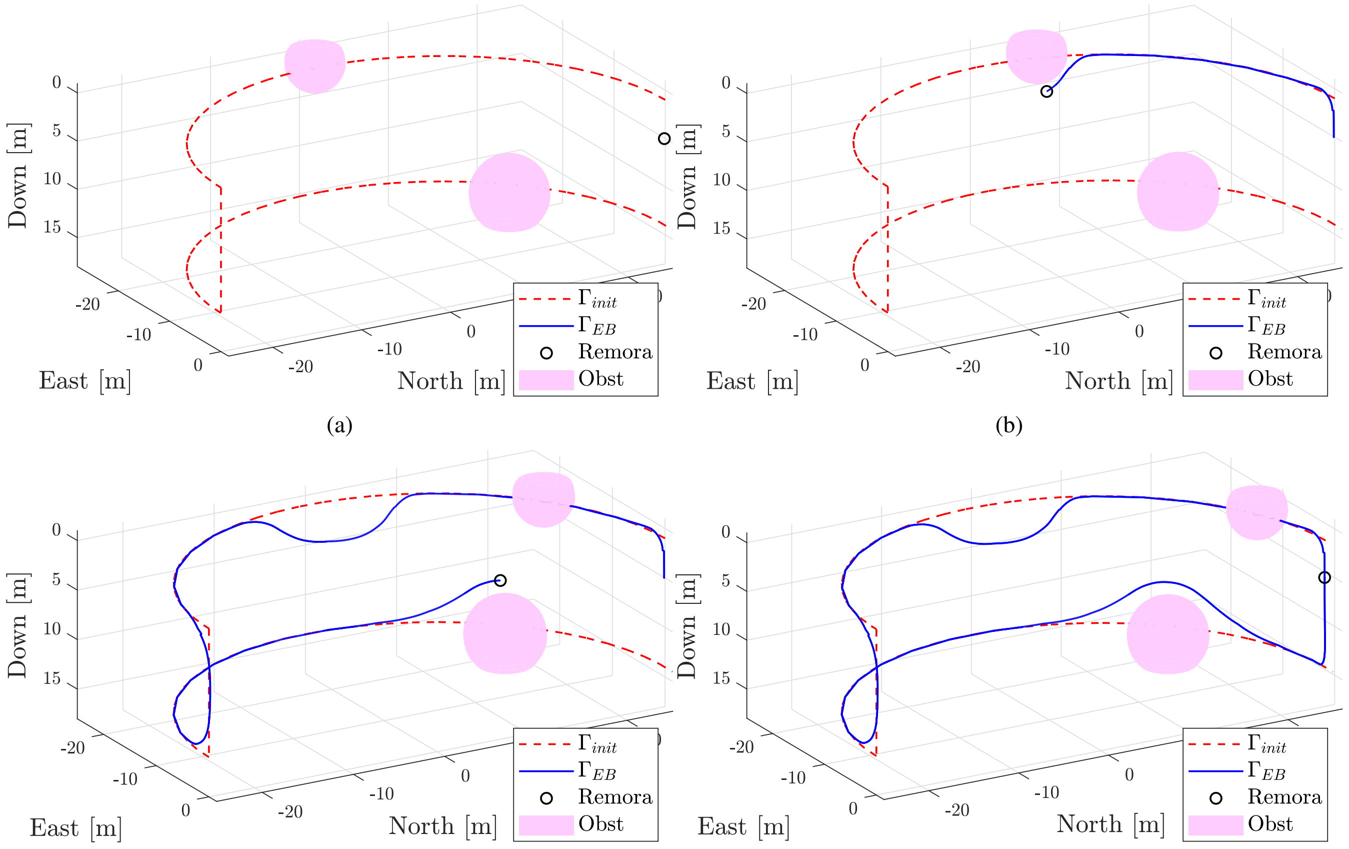
eterisation study was set up using the final parameters given in Table I. This simulation also contained two obstacles, but with the obstacle at the surface being dynamic (e.g. it was moving along the initial path of the robot but in the opposite direction). The purpose of this was to explore if a dynamic obstacle would compromise the method's abilities in adaptive motion planning.

The robot steered clear of the moving obstacle at the surface in the second simulation, indicating that the EBM was also able to handle dynamic obstacles (Figure 5). The main difference between the static and dynamic obstacle avoidance was that the deviation from the initial path (Γ_{init}) due to the dynamic obstacle was slightly longer than for the static obstacle.

When using the final choice of parameters, the robot demonstrated a clear ability to avoid both obstacles while deviating minimally from the initial path and hence the mission goals. This demonstrates that the Elastic Band Method is a viable tool for adaptive motion planning for underwater robots faced by unpredictable obstacles occurring in their path. The method appears robust in terms of parameter variations, and finding a set of parameters that will work for a particular application should not be an insurmountable challenge, even for cases where there are several unknown factors.

The introduction of a dynamic obstacle did not compromise the ability of the EBM to both avoid the obstacle and return to the prescribed path after this. This illustrates that this method can be used for adaptive motion planning also when the obstacles and conditions are dynamic. The increased deviation around the surface obstacle in the dynamic case occurred because the path modification induced by the EBM is cumulative. This means that it at startup modified the path based on the initial position of the obstacle. At the next update time step for the EBM, the already modified path would be further deformed to avoid the obstacle in its new position. This would be repeated at the next EBM update time step, and so on, until the robot is past the obstacle, a feature that could potentially render the use of EBM for adaptive planning challenging in situations where the robot needs to account for one or more dynamic obstacles.

Since the EBM cumulatively updates the path based on the instantaneous positions of all known obstacles, a moving obstacle can cause quite large path deformations even though the vehicle is not in direct risk of colliding with it. This can potentially lead to larger path deformations than strictly necessary, which may in turn prevent the vehicle from following the initial path that was designed based on the mission goals. Since this could lead to mission failure (e.g. incomplete net cleaning), the EBM should be equipped with measures to reduce or correct the deviations between initial and actual



(a)

(b)

(c)

(d)

Fig. 5: Trajectory of the robot when encountering a dynamic obstacle at the surface and a static obstacle at the bottom. a) Starting position, b) robot deviates to avoid dynamic obstacle, c) robot deviates to avoid static obstacle, d) robot returns to starting point.

paths, especially for longer trials. One way of doing this could be to let the vehicle retain the initial path through the entire campaign, and then set the EBM up to periodically check the deviation between the current path and the initial path, and dynamically correct larger deviations.

2) Simulation case study: The simulation case study was designed to resemble an actual cleaning operation in a fish farm to better relate the results to industrial practices. Simulations were conducted using a model of a commercial size net pen (25 m radius and 15 m vertical net wall depth). A virtual docking station was placed on the net at 5 m depth. The robot was programmed to follow an initial path (Γ_{init}) that spiralled down along the net from the surface to the bottom of the vertical net section. The path was designed such that it covered the vertical net section, and thus simulated a full cleaning cycle. Early in the simulation, a dynamic obstacle was programmed to move along the water surface, thus interfering with the initial stages of the motion plan for the robot. The robot was thus forced to modify its path to avoid collision with the obstacle. Later in the simulation, the robot was programmed to report a low battery level (less than 20%). This should prompt the robot to return to the docking station to charge. After charging, the robot should resume the operation from the point where it decided to return for charging. Finally, the robot encountered a static obstacle near the bottom of the net pen, before the sequence concluded when a full coverage of the net was achieved. Together, these elements should contribute to demonstrating the efficacy of the method for dynamic motion planning.

The EBM modified trajectory of the robot (Γ_{EB}) demonstrates that the robot was able to avoid both the dynamic and the static obstacles, without significantly deviating from the initial path (Γ_{init}). Moreover, the robot returned to the virtual docking station when the battery level was low, and after recharging resumed the planned path from the point where it left to dock.

The robot avoided colliding with all obstacles in the cleaning simulation case and otherwise followed the prescribed initial path closely. Furthermore, it was able to go to the docking station to recharge, and then resume the path from the point where it left to dock. Since these are all elements that may impact an actual cleaning operation in aquaculture, this implies that the Elastic Band Method is an attractive approach for adaptive motion planning for aquaculture robotics. In

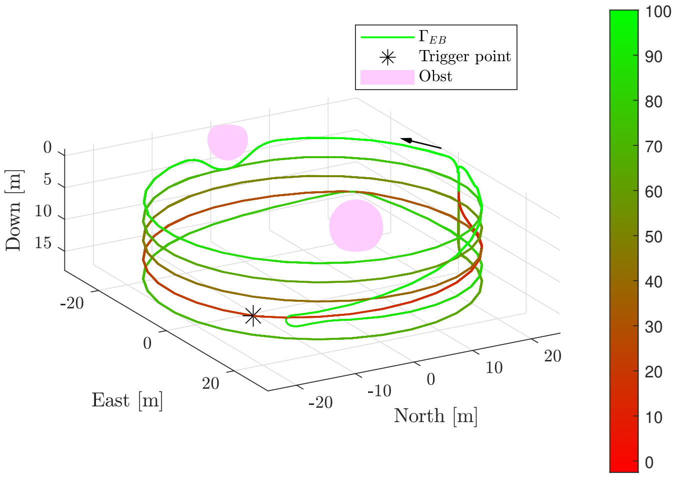


Fig. 6: Resulting trajectory (Γ_{EB}) for a simulated net pen cleaning operation with an initial trajectory (Γ_{init}) that covers the net using a spiral pattern. The colour of the trajectory describes the simulated battery level (green = high, red = low), and the asterisk marks the point where the robot decides to return to the docking for charging. The black arrow denotes the movement direction of the robot.

industrial terms the simulation results would translate to the robot being able to cover almost all the net pen surface, barring only the segments nearest to the obstacles.

To investigate the real-time capabilities of the EBM, the runtime of the algorithm was compared for different configurations, and with different number of points along the path. This is shown in Figure 7. The average runtime of the algorithm, for all cases, was approximately 0.272 milliseconds which can be argued to be low enough for applications in an aquaculture setting, considering the movement speed of the robot and the obstacles, which can be assumed to be low (≤ 1 m/s).

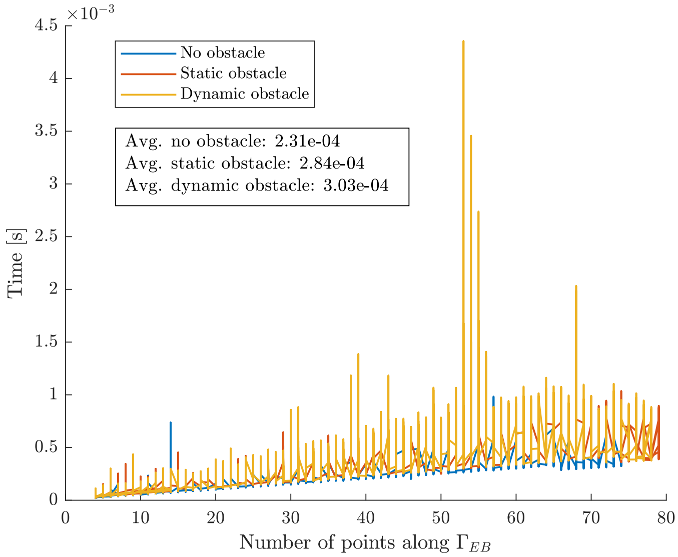


Fig. 7: Time complexity with 3 different configurations, evaluated against the number of points along the path, Γ_{EB}.

For all the investigated simulation case studies, the robot performance was evaluated based on a set of criteria including the ability to re-plan the path in real-time, the error between desired and actual path, and various controller performance metrics.

IV. Conclusions and Future Work

This paper presented a study where the potential of achieving autonomous net cleaning operations in commercial aquaculture net pens is demonstrated through numerical simulations. The EBM has been implemented and integrated in a simulation framework, with the necessary modifications and adaptations for use in an underwater robotic application in complex and dynamic environments. In particular, the EBM method has been integrated in a general control framework consisting of a model of the net-crawling robot, a model of the net pen, and a guidance and control system, to demonstrate applications of the Elastic Band Method to an underwater net cleaning robot for static and dynamic obstacle avoidance during net cleaning missions. The full functionality of the method was then explored through extensive simulation case studies styled after commercial net cleaning operations. The obtained results demonstrate that the adapted EBM method is well-suited for autonomous net cleaning operations in dynamically changing environments such as aquaculture net pens.

As future work the authors propose verifying the EBM for underwater net cleaning robots in a physical experiment. However, for such experiments to be successful, one would first need to acquire or develop methods for underwater positioning of the robot, and dynamic detection of obstacles. Such methods could be based on acoustic or optical means.

Acknowledgement

The authors would like to thank the partners in the NetClean24/7-project (Mithal AS, Aanderaa AS and Nordlaks AS) for their support in the activities leading up to this publication.

References

[1] J. Bannister, M. Sievers, F. Bush, and N. Bloecher, "Biofouling in marine aquaculture: a review of recent research and developments," *Biofouling*, vol. 35, no. 6, pp. 631–648, 2019.

[2] I. Fitridge, T. Dempster, J. Guenther, and R. De Nys, "The impact and control of biofouling in marine aquaculture: a review," *Biofouling*, vol. 28, no. 7, pp. 649–669, 2012.

[3] N. Bloecher, K. Frank, M. Bondø, D. Ribicic, P. C. Endresen, B. Su, and O. Floerl, "Testing of novel net cleaning technologies for finfish aquaculture," *Biofouling*, vol. 35, no. 7, pp. 805–817, 2019.

[4] H. Moe Føre and R. Gaarder, "Documentation of mechanical properties and dimensions of net material in aquaculture," *SINTEF Report. SINTEF Ocean, Trondheim*, p. 95, 2018.

[5] N. Bloecher and O. Floerl, "Towards cost-effective biofouling management in salmon aquaculture: a strategic outlook," *Reviews in Aquaculture*, 2020.

[6] M. Tribou and G. Swain, "The effects of grooming on a copper ablative coating: a six year study," *Biofouling*, vol. 33, no. 6, pp. 494–504, 2017.

[7] K. Z. Hunsucker, E. Ralston, H. Gardner, and G. Swain, "Specialized grooming as a mechanical method to prevent marine invasive species recruitment and transport on ship hulls," in *Impacts of Invasive Species on Coastal Environments*. Springer, 2019, pp. 247–265.

[8] "Mithal AS," https://www.mithal.no/, accessed: 2021-07-08.

[9] H. V. Bjelland, M. Føre, P. Lader, D. Kristiansen, I. M. Holmen, A. Fredheim, E. I. Grøtli, D. E. Fathi, F. Oppedal, I. B. Utne *et al.*, "Exposed aquaculture in norway," in *OCEANS 2015-MTS/IEEE Washington*. IEEE, 2015, pp. 1–10.

[10] S. J. Ohrem, E. Kelasidi, and N. Bloecher, "Analysis of a novel autonomous underwater robot for biofouling prevention and inspection in fish farms," in *2020 28th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2020, pp. 1002–1008.

[11] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn, "Path planning and obstacle avoidance for autonomous mobile robots: A review," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2006, pp. 537–544.

[12] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3d path planning algorithms," *Journal of Control Science and Engineering*, vol. 2016, 2016.

[13] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[14] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[15] Y. K. Hwang and N. Ahuja, "Gross motion planning—a survey," *ACM Computing Surveys (CSUR)*, vol. 24, no. 3, pp. 219–291, 1992.

[16] A. M. Lekkas, "Guidance and path-planning systems for autonomous vehicles," 2014.

[17] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[18] L. Zhang, Y. J. Kim, and D. Manocha, "A hybrid approach for complete motion planning," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 7–14.

[19] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1, pp. 65–100, 2010.

[20] R. Bohlin and L. E. Kavraki, "Path planning using lazy prm," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 521–528.

[21] A. Sánchez, R. Zapata, R. Cuautle, and M. Osorio, "Reactive motion planning for mobile robots," *Mobile Robots Motion Planning, New Challenges, I-Tech Education and Publishing*, pp. 469–486, 2008.

[22] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*. Springer, 2017, vol. 118.

[23] M. Yan, D. Zhu, and S. X. Yang, "A novel 3-d bio-inspired neural network model for the path planning of an auv in underwater environments," *Intelligent Automation & Soft Computing*, vol. 19, no. 4, pp. 555–566, 2013.

[24] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 802–807.

[25] T. Brandt and T. Sattel, "Path planning for automotive collision avoidance based on elastic bands," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 210–215, 2005.

[26] C.-T. Lee and C.-C. Tsai, "3d collision-free trajectory generation using elastic band technique for an autonomous helicopter," in *FIRA RoboWorld Congress*. Springer, 2011, pp. 34–41.

[27] S. J. Ohrem, H. Amundsen, and E. Kelasidi, "Control-oriented modeling of an underwater biofouling prevention and inspection robot," in *ICAR 2021*, Submitted for publication.

[28] K.-J. Reite, M. Føre, K. G. Aarsæther, J. Jensen, P. Rundtop, L. T. Kyllingstad, P. C. Endresen, D. Kristiansen, V. Johansen, and A. Fredheim, "Fhsim—time domain simulation of marine systems," in *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers Digital Collection, 2014.

[29] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.