AN AUTONOMOUS SURFACE VEHICLES PATH TRACKING AND COLLISION AVOIDANCE USING DEEP REINFORCEMENT LEARNING

R.Sreedhar, Associate professor, Sridevi Women's Engineering College, Hyderabad, E-mail id: rachasreedharswec@gmail.com
Arra Hima Sai Varsha, B.Tech, Dept of Information Technology, Sridevi Women's Engineering College, Hyderabad, E-mail id: himavarsha191@gmail.com
Surabhi Maithily, B.Tech, Dept of Information Technology, Sridevi Women's Engineering College, Hyderabad, E-mail id: maithilyrao.s@gmail.com
Kotagiri Supriya, B.Tech, Dept of Information Technology, Sridevi Women's Engineering College, Hyderabad, E-mail id: Supriyakotagiri@gmail.com
V.Rishika, B.Tech, Dept of Information Technology Sridevi Women's Engineering College, Hyderabad, E-mail id: rishikavangari@gmail.com

ABSTRACT: In this work, we explore the feasibility of applying proximal policy optimization, a state-of-the-art deep reinforcement learning algorithm for continuous control tasks, on the dual-objective problem of controlling an under actuated autonomous surface vehicle to follow an a priori known path while avoiding collisions with non-moving obstacles along the way. The AI agent is trained and evaluated in a challenging, stochastically generated simulation environment based on the Open AI gym Python toolkit. Notably, the agent is provided with real-time insight into its own reward function, allowing it to dynamically adapt its guidance strategy. Depending on its strategy, which ranges from radical path-adherence to radical obstacle avoidance, the trained agent achieves an episodic success rate close to 100%.
Keywords: *Machine learning algorithm, Reinforcement Learning.*

1. INTRODUCTION

Autonomy offers surface vehicles the opportunity to improve the efficiency of transportation while still cutting down on greenhouse emissions. However, for safe and reliable autonomous surface vehicles (ASV), effective path planning is a pre-requisite which should cater to the two important tasks of path following and collision avoidance (COLAV). In the literature, a distinction is typically made between reactive and deliberate COLAV methods. In short, reactive approaches, most notably artificial potential field methods, dynamic window methods, velocity obstacle methods and optimal control-based methods, base their guidance decisions on sensor readings from the local environment, whereas deliberate methods, among them popular graph-search algorithms such as and Voronoi graphs as well as randomized approaches such as rapidly-exploring random tree and probabilistic roadmap, exploit a priori known characteristics of the global environment in order to construct an optimal path in advance, which is to be followed using a low-level steering controller.

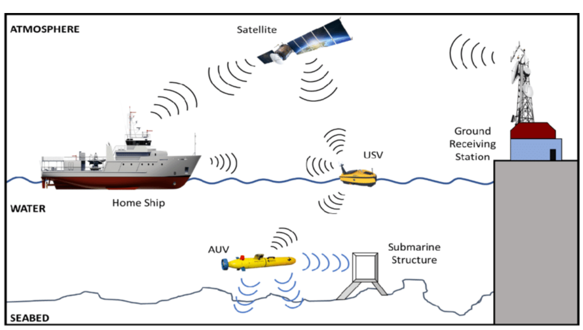

Fig.1 Autonomous surface vehicles architecture

By utilizing more data than just the current perception of the local neighborhood surrounding the agent, deliberate methods are generally more likely to converge to the intended goal, and less likely to suggest guidance strategies leading to dead ends, which is frequently observed with reactive methods due to local minima. However, in the case where the environment is not perfectly known, as a result of either incomplete or uncertain mapping data or due to the environment having dynamic features, purely deliberate methods often fall short. To prevent this, such methods are often executed repeatably on a regular basis to adapt to discrepancies between recent sensor observations and the a priori belief state of the environment.

However, as this class of methods are computationally expensive by virtue of processing global environment data, this is sometimes rendered infeasible for real-world applications with limited processing power, especially as the problem of optimal path planning amid multiple obstacles is provably NP-hard. Thus, a common approach is to utilize a reactive algorithm, which is activated whenever the presence of a nearby obstacle is detected, as a fallback option for the global, deliberate path planner. Such hybrid architectures are intended to combine the strengths of reactive and deliberate approaches and have gained traction in recent years. The approach presented in this work is somewhat related to this; the existence of some a priori known nominal path is presumed, but following it strictly will invariantly lead to collisions with obstacles.

2. LITERATURE REVIEW

The vector field histogram - fast obstacle avoidance for mobile robots, Robotics and Automation

A real-time obstacle avoidance method for mobile robots which has been developed and implemented is described. This method, named the vector field histogram (VFH), permits the detection of unknown obstacles and avoids collisions while simultaneously steering the mobile robot toward the target. The VFH method uses a two-dimensional Cartesian histogram grid as a world model. This world model is updated continuously with range data sampled by onboard range sensors. The VFH method subsequently uses a two-stage data-reduction process to compute the desired control commands for the vehicle. Experimental results from a mobile robot traversing densely cluttered obstacle courses in smooth and continuous motion and at an average speed of 0.6-0.7 m/s are shown. A comparison of the VFN method to earlier methods is given.

Motion planning and collision avoidance using navigation vector fields

This paper presents a novel feedback method on the motion planning for unicycle robots in environments with static obstacles, along with an extension to the distributed planning and coordination in multi-robot systems. The method employs a family of 2-dimensional analytic vector fields, whose integral curves exhibit various patterns depending on the value of a parameter lambda. More specifically, for an a priori known value of lambda, the vector field has a unique singular point of dipole type and can be used to steer the unicycle to a goal configuration. Furthermore, for the unique value of lambda that the vector field has a continuum of singular points, the integral curves are used to define flows around obstacles. An almost global feedback motion plan can then be constructed by suitably blending attractive and repulsive vector fields in a static obstacle environment. The method does not suffer from the appearance of sinks (stable nodes) away from goal point. Compared to other similar methods which are free of local minima, the proposed approach does not require any parameter tuning to render the desired convergence properties. The paper also addresses the extension of the method to the distributed coordination and control of multiple robots, where each robot needs to navigate to a goal configuration while avoiding collisions with the remaining robots, and while using local information only. More specifically, based on the results which apply to the single-robot case, a motion coordination protocol is presented which guarantees the safety of the multi-robot system and the almost global convergence of the robots to their goal configurations. The efficacy of the proposed methodology is demonstrated via simulation results in static and dynamic environments.

High-speed navigation using the global dynamic window approach

Many applications in mobile robotics require the safe execution of a collision-free motion to a goal position. Planning approaches are well suited for achieving a goal position in known static environments, while real-time obstacle avoidance methods allow reactive motion behavior in dynamic and unknown environments. This paper proposes the global dynamic window approach as a generalization of the dynamic window approach. It combines methods from motion planning and real-time obstacle avoidance to result in a framework that allows robust execution of high-velocity, goal-directed reactive motion for a mobile robot in unknown and dynamic environments. The global dynamic window approach is applicable to nonholonomic and holonomic mobile robots.

A modified dynamic window algorithm for horizontal collision avoidance for auvs

Much research has been done on the subject of collision avoidance (COLAV). However, few results are presented that consider vehicles with second-order nonholonomic constraints, such as autonomous underwater vehicles (AUVs). This paper considers the dynamic window (DW) algorithm for reactive horizontal COLAV for AUVs, and uses the HUGIN 1000 AUV in a case study. The DW algorithm is originally developed for vehicles with first-order nonholonomic constraints and is hence not directly applicable for AUVs without resulting in degraded performance. This paper suggests further developments of the DW algorithm to make it better suited for use with AUVs. In particular, a new method for predicting AUV trajectories using a linear approximation which accounts for second-order nonholonomic constraints is developed. The new prediction method, together with a modified search space, reduces the mean square prediction error to about one percent of the original algorithm. The performance and robustness of the modified DW algorithm is evaluated through simulations using a nonlinear model of the HUGIN 1000 AUV.

Proactive collision avoidance for asvs using a dynamic reciprocal velocity obstacles method

We propose a collision avoidance method that incorporates the interactive behavior of agents and is proactive in dealing with the uncertainty of the future behavior of obstacles. The proposed method considers interactions that will be experienced by an autonomous surface vessel (ASV) in an environment governed by the international regulations for preventing collisions at sea (COLREGs). Our approach aims at encouraging dynamic obstacles to cooperate according to COLREGs. Therefore, we propose a strategy for assessing the cooperative behavior of obstacles, and the result of the assessment is used to adapt collision avoidance decisions within the Reciprocal Velocity Obstacles (RVO) framework. Moreover, we propose a predictive approach to solving known limitations of the RVO framework, and we present computationally feasible extensions that enable the use of complex dynamic models and objectives suitable for ASVs. We demonstrate the performance and potentials of our method through a simulation study, and the results show that the proposed method leads to proactive and more predictable ASV behavior compared with both Velocity Obstacles (VO) and RVO, especially when obstacles cooperate by following COLREGs.

Obstacle avoidance for low-speed autonomous vehicles with barrier function

This paper presents an obstacle avoidance algorithm for low speed autonomous vehicles (AV), with guaranteed safety. A supervisory control algorithm is constructed based on a barrier function method, which works in a plug-and-play fashion with any lower-level navigation algorithm. When the risk of collision is low, the barrier function is not active; when the risk is high, based on the distance to an “avoidable set”, the barrier function controller will intervene, using a mixed integer program to ensure safety with minimal control effort. This method is applied to solve the navigation and pedestrian avoidance problem of a low speed AV. Its performance is compared with two benchmark algorithms: a potential field method and the Hamilton-Jacobi method.

A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games:

We describe and implement an algorithm for computing the set of reachable states of a continuous dynamic game. The algorithm is based on a proof that the reachable set is the zero sublevel set of the viscosity solution of a particular time-dependent Hamilton-Jacobi-Isaacs partial differential equation. While alternative techniques for computing the reachable set have been proposed, the differential game formulation allows treatment of nonlinear systems with inputs and uncertain parameters. Because the time-dependent equation's solution is continuous and defined throughout the state space, methods from the level set literature can be used to generate more accurate approximations than are possible for formulations with potentially discontinuous solutions. A numerical implementation of our formulation is described and has been released on the web. Its correctness is verified through a two vehicle, three- dimensional collision avoidance example for which an analytic solution is available.

3. IMPLEMENTATION

Existing System

Collision avoidance, traditionally considered a high-level planning problem, can be effectively distributed between different levels of control, allowing real-time robot operations in a complex environment. We have applied this obstacle avoidance scheme to robot arm using a new approach to the general problem of real-time manipulator control. We reformulated the manipulator control problem as direct control of manipulator motion in operational space-the space in which the task is originally described-rather than as control of the task's corresponding joint space motion obtained only after geometric and kinematic transformation. This method has been implemented in the COSMOS system for a PUMA 560 robot. Using visual sensing, real-time collision avoidance demonstrations on moving obstacles have been performed.

Limitations:

➢ As this class of methods are computationally expensive by virtue of processing global environment data, this is sometimes rendered infeasible for real-world applications with limited processing power.

Proposed Method

We propose three approaches for transforming the sensor readings into a reduced observation space from which a satisfactory policy mapping should be easier to achieve. applying proximal policy optimization, a state-of the-art deep reinforcement learning algorithm for continuous control tasks, on the dual-objective problem of controlling an under actuated autonomous surface vehicle to follow an a priori known path while avoiding collisions with non-moving obstacles along the way. The AI agent is trained and evaluated in a challenging, stochastically generated simulation environment based on the OpenAI gym Python toolkit.

Advantages:

➢ This presents a wholehearted acceptance of these algorithms for safety critical applications

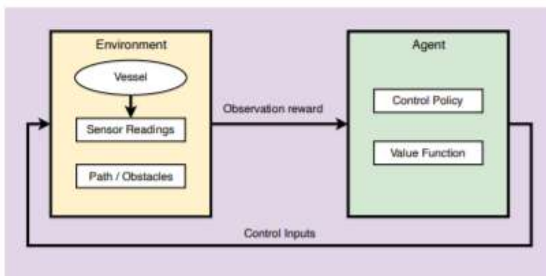

Fig.2: System architecture

RL is an area of machine learning (ML) of particular interest for control applications, such as the guidance of surface vessels under consideration here. Fundamentally, this ML paradigm is concerned with estimating the optimal behavior for an agent in an unknown, and potentially partly unobservable environment, relying on trial-and-error-like approaches in order to iteratively approximate the behavior policy that maximizes the agent's expected long-time reward in the environment. The field of RL has seen rapid development over the last few years, leading to many impressive achievements, such as playing chess and various other games at a level that is not only superhuman, but also overshadows previous AI approaches by a wide margin.

The focus of this paper is to explore how RL, given the recent advances in the field, can be applied to the guidance and control of ASV. Specifically, we look at the dual objectives of achieving the ability to follow a path constructed from a priori known way-points, while avoiding collision with obstacles along the way. In an end-to-end fashion, control signals for a simulated vessel are generated by a RL agent which, based on the readings from a rangefinder sensor suite which is attached to the vessel as well as rewards received from the environment, learns how to intelligently control the vessel in challenging obstacle avoidance scenarios.

For simplicity, we limit the scope of this work to non-moving obstacles of circular shapes. As RL methods are, model-free approaches, by their very nature, a positive result can bring significant value to the robotics and autonomous system field, where implementing a guidance system typically requires knowledge of the vessel dynamics, in the form of non-linear first-principle models with parameters that can only be determined experimentally at great cost.

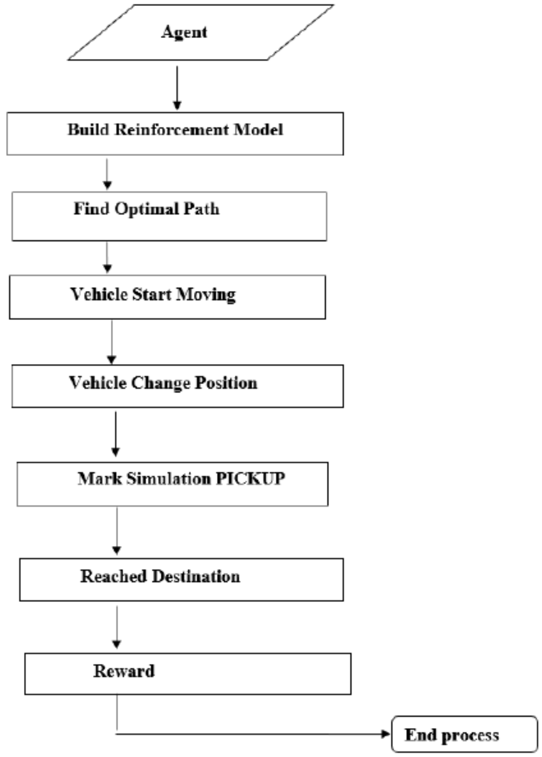

Fig.3: Dataflow diagram

4. ALGORITHMS

Reinforcement learning algorithm:

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Example: The problem is as follows: We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.

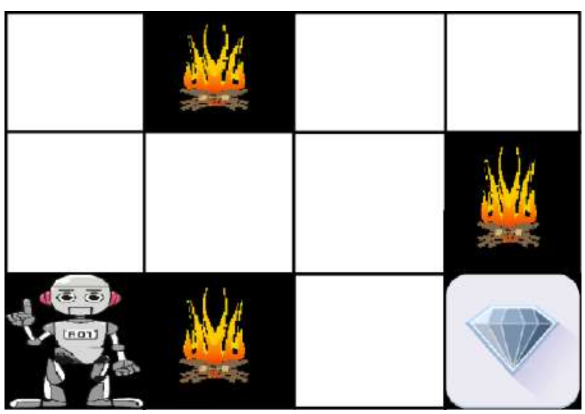

Fig.4: Reinforcement learning algorithm

The above image shows the robot, diamond, and fire. The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.

Main points in Reinforcement learning – Input:

The input should be an initial state from which the model will start

Output: There are many possible outputs as there are a variety of solutions to a particular problem

Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.

The model keeps continues to learn.

The best solution is decided based on the maximum reward.

Types of Reinforcement:

There are two types of Reinforcement:

Positive – Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior. Advantages of reinforcement learning are:

- Maximizes Performance
- Sustain Change for a long period of time
- Too much Reinforcement can lead to an overload of states which can diminish the results

Negative:

Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided. Advantages of reinforcement learning:

- Increases Behavior
- Provide defiance to a minimum standard of performance
- It Only provides enough to meet up the minimum behavior

Various Practical applications of Reinforcement Learning –

❖ RL can be used in robotics for industrial automation.

❖ RL can be used in machine learning and data processing

❖ RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.

Proximal Policy Optimization:

The PPO algorithms strikes a balance between ease of implementation and data efficiency, and is likely to perform well in a wide range of continuous environments without extensive hyperparameter tuning. Sensitivity to hyperparameter choices is a frequently encountered problem for policy gradient methods and given the computation time required to train and test agents in a collision avoidance environment, this could be a detrimental bottleneck in our research.

for iteration = 1, 2, ... do

for actor = 1, 2, ...N do

For T time-steps, execute policy πθ.

Compute advantage estimates Aˆ1, . . . AˆT

for epoch = 1, 2, ...NE do

Obtain mini batch of NMB samples from the NAT simulated time-steps. Perform SGD update from minibatch (XMB, YMB). θ ← θ 0

5. EXPERIMENTAL RESULTS

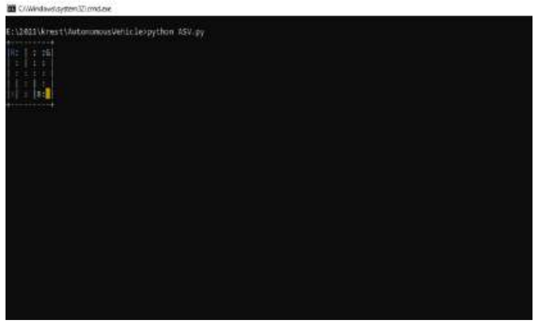

Fig.5: Output screen1

In above screen vehicle simulation displayed and then reinforcement learning will start training to get optimal path by avoiding obstacle to reach destination

Fig.6: Output screen2

In above screen reinforcement algorithm start iterating or running episode to predict or to take action to find optimal path.

Fig.7: Output screen3

In above screen you can see yellow vehicle start moving towards Y or B

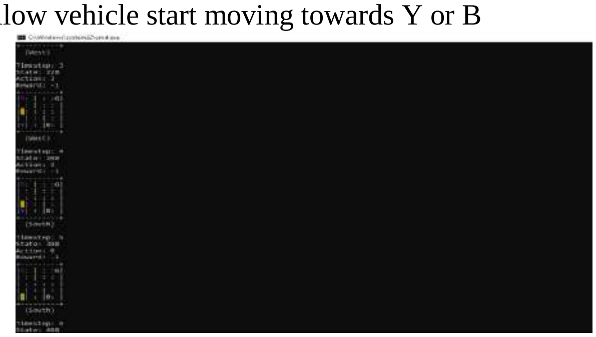

Fig.8: Output screen4

In above screen we can see vehicle reached to source Y or B location as both are in same row and see below screen where vehicle changed to green colour after reaching to source Y

Fig.9: Output screen5

In above screen in first diagram vehicle changed to green and mark simulation as PICKUP and then start moving towards destination R

Fig.10: Output screen6

In above screen once vehicle reached to destination then vehicle changed backed to yellow colour and after taking correct decision at destination algorithm got rewards as 20. In below graph we can see rewards.

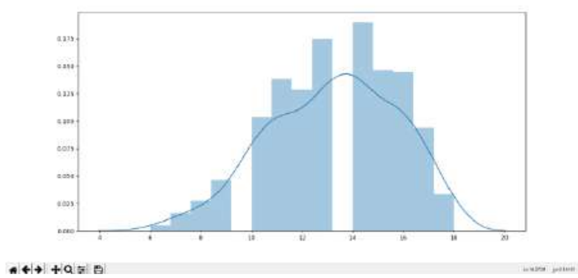

Fig.11: Probability-Reward Graph

In above graph x-axis represents rewards and y-axis represents probability of correct decision and the lower the probability the better is the action or decision. In above graph at 20th reward the decision was accurate.

6. CONCLUSION

In this work, we have demonstrated that RL is a viable approach to the challenging dual-objective problem of controlling a vessel to follow a path given by a priori known way-points while avoiding obstacles along the way without relying on a map. More specifically, we have shown that the state-of-the-art PPO algorithm converges to a policy that yields intelligent guidance behavior under the presence of non-moving obstacles surrounding and blocking the desired path.

REFERENCES

[1] R. W. Beard and T. W. McLain, Small Unmanned Aircraft: Theory and Practice. Princeton, NJ, USA: Princeton Univ. Press, 2012.

[2] O. Khatib, ‘‘Real-time obstacle avoidance for manipulators and mobile robots,’’ Int. J. Robot. Res., vol. 5, no. 1, pp. 90–98, Jul. 2016, doi: 10.1177/027836498600500106.

[3] J. Borenstein and Y. Koren, ''The vector field histogram-fast obstacle avoidance for mobile robots,'' IEEE Trans. Robot. Autom., vol. 7, no. 3, pp. 278–288, Jun. 1991.
[4] D. Panagou, ''Motion planning and collision avoidance using navigation vector fields,'' in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2014, pp. 2513–2518.
[5] D. Fox, W. Burgard, and S. Thrun, ''The dynamic window approach to collision avoidance,'' IEEE Robot. Autom. Mag., vol. 4, no. 1, pp. 23–33, Mar. 1997.
[6] O. Brock and O. Khatib, ''High-speed navigation using the global dynamic window approach,'' in Proc. IEEE Int. Conf. Robot. Autom., vol. 1, May 1999, pp. 341–346.
[7] B.-O.-H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig, ''A modified dynamic window algorithm for horizontal collision avoidance for AUVs,'' in Proc. IEEE Conf. Control Appl. (CCA), Sep. 2016, pp. 499–506.
[8] P. Fiorini and Z. Shiller, ''Motion planning in dynamic environments using velocity obstacles,'' Int. J. Robot. Res., vol. 17, no. 7, pp. 760–772, Jul. 1998. [Online]. Available: http://dblp.unitrier.de/db/journals/ijrr/ijrr17.html#FioriniS98
[9] D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, ''Proactive collision avoidance for ASVs using a dynamic reciprocal velocity obstacles method,'' in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 2402–2409.
[10] Y. Chen, H. Peng, and J. Grizzle, ''Obstacle avoidance for low-speed autonomous vehicles with barrier function,'' IEEE Trans. Control Syst. Technol., vol. 26, no. 1, pp. 194–206, Jan. 2018.
[11] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, ''A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games,'' IEEE Trans. Autom. Control, vol. 50, no. 7, pp. 947–957, Jul. 2005.
[12] B. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, ''The branching-course model predictive control algorithm for maritime collision avoidance,'' J. Field Robot., vol. 36, no. 7, pp. 1222–1249, Aug. 2019.
[13] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, ''MPCbased collision avoidance strategy for existing marine vessel guidance systems,'' in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 7618–7623.
[14] G. Bitar, M. Breivik, and A. M. Lekkas, ''Energy-optimized path planning for autonomous ferries,'' IFAC-PapersOnLine, vol. 51, no. 29, pp. 389–394, 2018. [Online]. Available: http://www.sciencedirect. com/science/article/pii/S2405896318321451
[15] P. Hart, N. Nilsson, and B. Raphael, ''A formal basis for the heuristic determination of minimum cost paths,'' IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, Jul. 1968.