Modeling, Identification and Control, Vol. 43, No. 3, 2022, pp. 101–109, ISSN 1890–1328

Leveraging Past Experience for Path planning of Marine Vessel: A Docking Example

Peihua Han Guoyuan Li Houxiang Zhang

Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Aalesund, Norway. E-mail: {*from,peihua.han,guoyuan.li,hozh*}*@ntnu.no*

Abstract

Path planning before maneuvering is crucial for the safe and efficient operations of marine vessels. The past successful human maneuvering experience can be leveraged to enable the safe and efficient path planning of vessels. In this paper, the previous successful maneuvering operations from ship operators are leveraged to find the optimal path. A deep conditional generative model is used to learn the distribution from those experiences. The model is then combined with the sampling-based RRT* planning algorithm to guide the search process. In this way, the theoretical guarantee of RRT* is preserved while the sampling process is more efficient. The docking operation is used as an example to validate the method. Experimental results show that the presented method not only improves the success rate and convergence speed to the optimal cost but also generalizes well to starting points beyond maneuvering experience.

Keywords: path planning; RRT*; conditional variational autoencoder; learning from past experience

1 Introduction

Intelligent maritime transportation systems have received great interest among the maritime industry in the past decades, which involve the development of remotely operated and autonomous ships. These ships have the potential to increase maritime transportation efficiency and reduce human-based errors while lowering fuel consumption and extending the operational window (Jalonen et al., 2016). With the improvement of data accessibility, advanced control theory, and computing power, ship intelligence has become the main topic to achieve ship autonomy, and thus gradually move from manually operated ships to fully autonomous ships.

Although much focus and effort have been put into ship autonomy in recent years (Han et al., 2022), ship manoeuvring is still largely a manual task performed by ship operators. Planning an optimal trajectory is of key importance for safe and efficient maneuvering. Sampling-based planning algorithms have emerged as a promising framework for solving planning problems with constrained kinematics and complex environments. These algorithms do not require discretization of the state space and avoid explicit construction of obstacles by using a collision checking module (Karaman and Frazzoli, 2011). Rapidly-exploring random tree (RRT) (LaValle and Kuffner Jr, 2001) is one of the most popular sampling-based planning algorithms. It is a tree-based planner that does not require an exact connection of the two states, so it is easier to deal with differential constraints. To address the lack of optimality of path found by RRT, RRT* (Karaman and Frazzoli, 2010) is proposed by introducing incremental rewiring of the graph. The RRT* has been studied extensively for collision avoidance of marine vessels in open waters (Zaccone and Martelli, 2020; Enevoldsen et al., 2021). Usually, RRT* draws random state samples from a uniform distribution, which is sufficient to guarantee the probabilistic completeness and asymptotic optimality (Karaman and Frazzoli, 2011). However, marine vessel often works in an open ocean

doi:10.4173/mic.2022.3.2

with a predetermined seaway combined with narrow passages. It can be inefficient to uniformly sample the entire state space. In docking operations, for instance, a vessel needs to travel from open water with few obstacles to a docking point, which is usually located in a narrow passage. In such a case, the marine vessel works in a small subset of the state space and therefore it is reasonable to bias the samples towards promising regions.

One way to bias the sample distribution is to use past experience. The successful maneuvering operations performed by human operators can be leveraged. Besides, vessels with similar deadweight and draught might share similar maneuvering plans. This provides possibly essential data in a given port to bias the sample. In this paper, a deep conditional generative model is utilized to learn the sampling distribution from previous human experience. By utilizing learning-based models, explicitly defining rules for sampling promising regions is avoided, and therefore it is easier to apply in different areas. In this paper, the docking operation is used as an example. Previous human docking maneuver data from a research vessel is used to train this model. The model is then implemented as the sampling module in the RRT* algorithm. Experimental results show that the success rate and convergence speed to optimal cost is improved with the learned sampler. In addition, the method generalizes well beyond the starting point of maneuvering experience. Key contributions of this paper include the construction of a learned sampler in RRT* for docking of the marine vessels, as well as verification of the planning performance.

The remainder of this paper is organized as follows: Section 2 presents previous research in this domain. Section 3 describes an RRT* containing a sampler learned from human experience. The experiments and experimental results are discussed in Section 4. Section 5 concludes the paper.

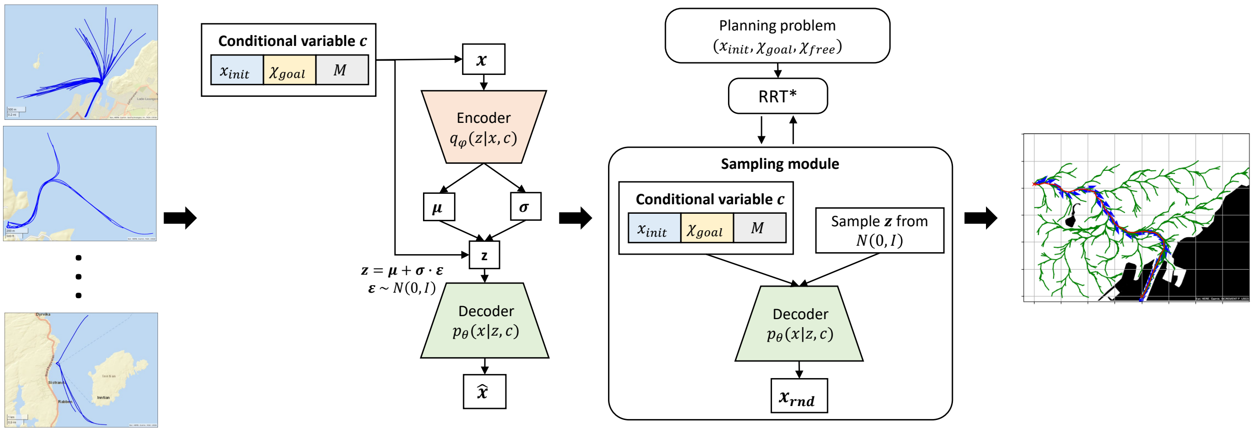
2 RELATED WORK

2.1 Learn from past planning experience

Several methods have been proposed to conduct more efficient planning for future problems by leveraging past planning experience. Tamar et al. (2016) proposed value iteration networks that are capable of finding near-optimal trajectories in 2-D and 3-D mazes by learning an approximate planning computation. Qureshi et al. (2019) proposed a motion planning network that uses a neural network to learn the state transition from past experience. It is used online iteratively to plan new paths. The above works learn a planner end-to-end. Other researches attempt to bias the sampler in sampling-based planning methods via learning-based methods. Kuo et al. (2018) proposed to use sequential models that are trained with successful plans to guide the steering function in RRT*. Huh and Lee (2018) treated the sampler in the RRT as a stochastic policy to be learned via Q-learning. Ichter et al. (2018) used a conditional variational autoencoder to encode the collected experience to the sampler in the sampling-based planning algorithms. Kim et al. (2018) used a generative adversarial network to represent an action sampler and introduced importance sampling for using samples from a non-target distribution to make learning more efficient. Qureshi and Yip (2018) proposed to learn the past experience via a stochastic feed-forward neural network and used it as the sampler in RRT*. Wang et al. (2020) used a convolutional neural network to predict the probability distribution of the optimal path on the map and guide the sampling process of RRT*. In this paper, we leverage learning-based methods to bias samplers in sampling-based planning methods by learning from past human experiences. The main reason is that this kind of method still maintains the theoretical guarantee of the sampling-based methods.

2.2 Automatic vessel docking

In recent years, there has been an increased interest in automatic vessel docking. Most of the works come from optimal control theory. Martinsen et al. (2019) framed the docking of the marine vessels as an optimal control problem with the addition of spatial constraints and solved it with nonlinear model predictive control. Martinsen et al. (2021) further developed a two-stage method for trajectory planning, which uses graph search on a precomputed mesh for path generation and convex optimization for path refinement. Similarly, Bergman et al. (2020) first used a lattice-based motion planner to compute a sub-optimal feasible path, and then improvement to the path is performed by receding horizon optimization. Miyauchi et al. (2022) represented the spatial obstacles such as berths, buoys, anchoring vessels as polygons and optimized the docking trajectory with evolution strategy. The above methods are optimization-based methods in spite of whether a sub-optimal initial guess is used, which are often computationally intensive and can be difficult to deal with non-convex obstacles. Learning-based methods, such as imitation learning Shuai et al. (2019) and reinforcement learning Anderlini et al. (2019), have also been investigated for ship docking. However, these methods are known to be data-intensive and might fail in out-of-distribution scenarios.



(a) Human Docking Manoeuvring Data (b) Offline Training of CVAE (c) Online Execution of RRT* with Learned Sampler (d) Planning Path

Figure 1: Schematic illustration of the method that leverages human experience in RRT*.

3 RRT* with learned sampler

The details of the approach to learn a bias sampler from past human experience and integrate it with RRT* for docking are presented here. Note that a planning problem is defined by $(x_{init}, \chi_{goal}, \chi_{free})$, where x_{init} is initial state, χ_{goal} is goal state region, and χ_{free} is obstacle-free state space. The task is to find a feasible/optimal path from x_{init} to χ_{goal}.

3.1 Learned sampler with human experience

The goal is to make use of human experience in the sampling procedure of RRT*, where we use a conditional generative model to learn the distribution of past experience. Common used deep conditional generative model includes conditional variational autoencoder (CVAE) (Sohn et al., 2015) and conditional generative adversarial network (CGAN) (Mirza and Osindero, 2014). We choose to use CVAE as in Ichter et al. (2018) because CGAN is harder to train and may suffer from mode collapse.

Fig. 1 shows the schematic illustration of the method. Human docking maneuvering data is collected, which includes the initial state x_{init}, goal state region χ_{goal}, map M, and state of the trajectory $x_t(t = 1, 2, ..., T)$. The x_{init}, χ_{goal}, M are taken as conditional variables to train a CVAE to reconstruct x_t. The decoder of the trained CVAE is then used as the sampling module for RRT*. The RRT* with the learned sampler can generate a path when a new planning problem is introduced.

3.2 Conditional variational autoencoder

The conditional variational autoencoder (Sohn et al., 2015) is a deep generative model. It is an extension of variational autoencoders (VAE) (Kingma and Welling, 2013) that provides control over the VAE data generation process. The CVAE consists of an encoder $q_\phi(z|x,c)$ and a decoder $p_\theta(x|z,c)$. The encoder $q_\phi(z|x,c)$ transforms the sampled points x and conditions c into the latent variables z, where the conditions c in our case can be initial state x_{init}, goal region χ_{goal} and map of obstacles M. The decoder $p_\theta(x|z,c)$ reconstructs the sampled points x from latent variables z and conditions c. The latent variables z are stochastic variables that can be denoted as $p(z|c)$. Note that the encoder and decoder are modeled in the structure of the neural network which is parametrized by ϕ and θ, respectively. The CVAE optimizes the parameters, ϕ and θ, by maximizing the following function:

$$E_{q_\phi(z|x,c)}[\log p_\theta(x|z,c)] - D_{KL}\big(q_\phi(z|x,c)||p_\theta(z|c)\big) \\ \leq \log p(x|c) \tag{1}$$

where D_{KL} is the Kullback-Leibler (KL) divergence. Minimizing the above loss function equals to minimizing the Evidence Lower Bound (ELBO) of the log likelihood $\log p(x|c)$, which is essentially the conditional probability we would like to learn for the sampler. The KL divergence constrains the latent bottleneck and limits the representation capacity of latent variables. To better balance the trade-off between reconstruction quality and efficient representation, a β-VAE (Higgins et al., 2016) is used here. Note that the latent variables $p(z|c)$ are usually modelled as an isotropic unit-variance Gaussian. The output x can be assumed

Algorithm 1 Online execution of RRT* with learned sampler

Input: Motion planning $(\chi_{free}, x_{init}, \chi_{goal})$

1: $T \leftarrow$ InitializeTree()
2: $T \leftarrow$ InsertNode(x_{init})
3: $c \leftarrow$ ConstructCondition(x_{init}, χ_{free}, M)
4: **for** $i = 1$ **to** $i = n_{iter}$ **do**
5:   **if** UniformSample(0, 1) ≤ 0.5 **then**
6:     $x_{rand} \leftarrow$ CVAESampler(c, χ_{free})
7:   **else**
8:     $x_{rand} \leftarrow$ RandomSampler(χ_{free})
9:   **end if**
10:   $x_{nearest} \leftarrow$ Nearest(T, x_{rand})
11:   $(x_{new}, u_{new}) \leftarrow Steer(x_{nearest}, x_{rand})$
12:   **if** CollisionFree(x_{new}) **then**
13:     $\chi_{near} = Near(x_{new}, T)$
14:     $x_{min} \leftarrow ChooseParent(\chi_{near}, x_{new})$
15:     $T \leftarrow$ InsertNode(x_{min})
16:     $T \leftarrow ReWire(T, \chi_{near}, x_{min})$
17:   **end if**
18: **end for**
19: **return** T

to follow a normal distribution as well. Maximizing Eq. (1) is then equivalent to minimizing the following loss function:

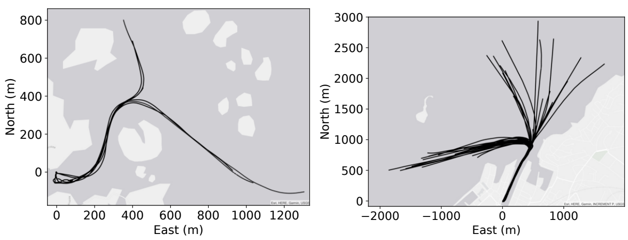
$$loss = \|x - \hat{x}\|^2 + \beta D_{KL}\big(q_\phi(z|x, c)||\mathcal{N}(0, I)\big) \qquad (2)$$

where $\hat{x}$ is the reconstructed sampled points. β is a hyperparameter. Once trained, the decoder can be used to generate samples from $p(x|c)$ by sampling from the isotropic unit-variance Gaussian in the latent space.

3.3 Online execution of RRT* with learned sampler

In the online phase, the trained neural-informed sampler is utilized to generate samples for RRT*. The pseudo code for online execution of the proposed neural-informed RRT* is outlined in Algorithm 1. Note that the use of neural-informed sampler is highlighted in the gray box.

The online phase starts by initializing a planning problem $(\chi_{free}, x_{init}, \chi_{goal})$ and gradually grow a tree T. Conditional variable c can be constructed based on x_{init}, χ_{goal}, M (Line 3). The algorithm runs for n_{iter} iterations (Line 4). At each iteration, there is a probability of 50% to sample from the learned CVAE sampler (Line 6). Specifically, a sample can be generated by sampling from the latent space at $N(0, I)$, conditioning on c. In this way, the tree T tends to grow on existing experience. Additionally, a uniform sampler is used with a probability of 50% in each iteration (Line 8) to ensure completeness guarantees of RRT*. The sample generated from the above procedure is then be used in the RRT* method (Line 10-17). Details of the RRT* algorithm can be found in Karaman and Frazzoli (2010); Karaman et al. (2011).



(a) (b)

Figure 2: Northeast path taken by the RV Gunnerus for all docking operations: (a) Mausund, and (b) Trondheim.

RRT* is known to ensure probabilistic completeness and asymptotic optimality (Karaman and Frazzoli, 2011), that is, as the number of iterations approaches to infinity, the probability of finding a feasible path/minimum cost path approaches to one. These theoretical guarantees still hold for the proposed method due to the inclusion of an auxiliary uniform sampler (Ichter et al., 2018).

3.4 Vessel assumption

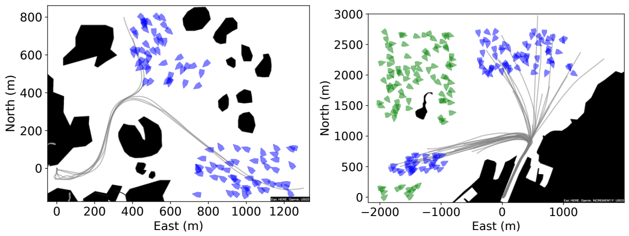
Although the presented method can be used to perform motion planning, this study is simplified to path planning for vessel docking. The maneuvering capability is then described by a minimum turning radius and a maximum extending distance. The state of the vessel is therefore defined as its location and heading $(north, east, heading)$.

4 EXPERIMENT

4.1 Vessel docking data

The vessel docking data is collected from a data acquisition system onboard the NTNU's research vessel (RV) Gunnerus[1]. A one-year time period was selected starting from August 2016 and ending in June 2017. The procedure to isolate successful dockings during these periods is detailed in Skulstad et al. (2020). In this study, only the docking operations in the Mansund harbor and Trondheim harbor are used. There are a total of 45 docking operations in Trondheim harbor while

[1]RV Gunnerus, https://www.ntnu.edu/oceans/gunnerus.



(a) (b)

Figure 3: The sampled starting points for evaluation: (a) Mausund, and (b) Trondheim. Note that blue is within human experience, while green is beyond human experience.

there are only 5 in Mansund harbor. The time of the docking operations is limited to approximately 20 minutes. The original dataset is collected at a sampling rate of 1 Hz and it is down-sampled to 0.25 Hz in this study.

In order to generalize the position coordinates across docking locations, the position of the ship is processed. The docking position is first determined from the 45 trajectories and is used as the origin (0, 0) for each docking operation. A conversion from the position given as latitude and longitude in the earth-centered, earth-fixed (ECEF) frame to the local northeastdown (NED) frame in meters was performed. Fig. 2a and Fig. 2b show the path taken by the vessel toward the docking location at coordinates (0, 0) m in the Mausund and Trondheim harbor, respectively. It is worth noting that the obstacles in the open waters are sparse and the docking point is located in a narrow passage in Fig. 2b.

4.2 Experimental settings

To evaluate the performance of the method, starting points for docking operations are randomly sampled. At the Mausund port, 100 starting points are sampled as shown in Fig. 3a. At the Trondheim port, 100 starting points within the range of human experience and 100 starting points outside the range of human experience are randomly sampled, as presented in Fig. 3b. The two types of starting points are distinguished by green and blue. Note that we only distinguish the starting point in the Trondheim harbor because it is easy to distinguish. The docking point is located at coordinates (0, 0) m.

4.2.1 Baseline methods

We implement two baselines to demonstrate the performance of the proposed method:

- **Manually-defined bias (MDB).** From the human maneuvering data, we invited experts to select several states that are considered critical for docking operations. Then the probability of sampling these states is increased in the original RRT*.

- **Deeply-informed neural sampling (DINS).** We follow the procedure in Qureshi and Yip (2018). A feed-forward neural network is trained to learn the state transition from the human experience. The trained model is then used as a sampler for the first 500 iterations in RRT*. For the rest of the iterations, a uniform sampler is used. The stochastic of the trained neural network is achieved through the use of Dropout (Srivastava et al., 2014).

4.2.2 Evaluation metrics

To evaluate the performance of the proposed method, a set of fixed-number samples of varying numbers are given to the samplers. Two metrics are used: a) success rate to find a feasible path and b) the cost of the solution path.

4.2.3 Implementation details

The CVAE sampler is implemented in Pytorch (Paszke et al., 2019). The CVAE consists of an encoder and a decoder with 4 dense layers for each. The size of the latent variables is set to 8. The network is trained with Adam optimizer at a learning rate of 0.01. The hyperparameter β in Eq. (2) is set to 0.01. The RRT* planning algorithm is implemented in Python. Note that we use a goal biased version of RRT*, which samples the goal state with a probability of 5%.

4.3 Experimental results

4.3.1 Mausund harbor

Fig. 4 presents the examples of the planning path using RRT* with CVAE sampler in the Mausund harbor. It is shown that even though the trajectory data is limited, the CVAE still captures the promising regions to the docking location. By following the CVAE sampling region, a smooth path can be found despite the different states of the starting point.

Fig. 5 shows the convergence of success rate and cost for an average of 100 starting points at the Mausund port. The MDB has the worst success rate and normalized cost. The CVAE performed slightly better than the DINS in terms of success rate, but almost the same in terms of normalized cost. Even with limited data,

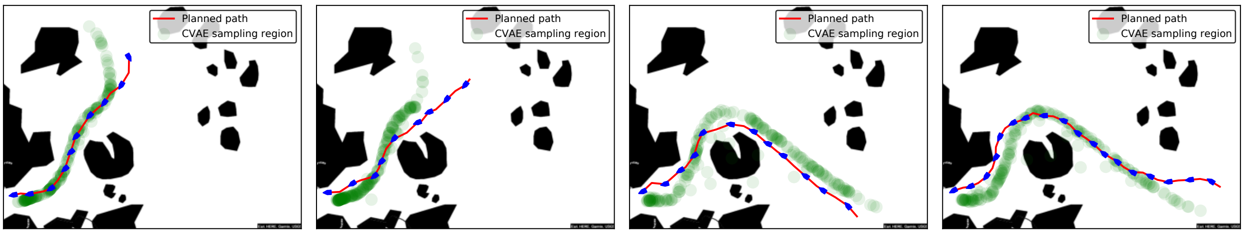


Figure 4: Examples of the path using RRT* with a learned CVAE sampler in the Mausund harbor.

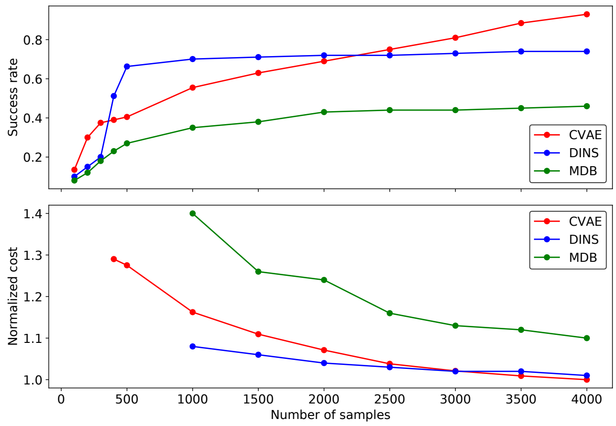


Figure 5: Comparison of the convergence rate in the Mansund harbor.

learned samplers can provide better success rates and optimal costs than human-defined biases.

4.3.2 Trondheim harbor

Fig. 6 and Fig. 7 show the examples of the planning path using RRT* with CVAE sampler when the starting point is within and beyond maneuvering experience in the Trondheim harbor, respectively. In Fig. 6, it is shown the proposed method can easily localize the path in the CVAE sampling region, resulting in a smooth low-cost path to the docking location. In Fig. 7, even though the CVAE sampling region might not be collision-free, the method can still find a path towards the docking location. Since we only have limited maneuvering data that excludes the area behind the island, CVAE generates samples that traverse the island. In such cases, it is observed that the path gets around the obstacle and then follows the CVAE sampling region. This might be owed to the RRT* and the auxiliary uniform sampler. Therefore, the proposed method can be generalized to starting points beyond human maneuvering experience.

The convergence of success rate and cost is presented in Fig. 8 and Fig. 9. In both figures, the RRT* with a learned sampler, no matter CVAE or DINS, outperforms that with a manual-biased sampler in terms of success rate and normalized cost. This may be attributed to that the learned sampler bias the samples towards and inside the narrow passage. In Fig. 8, it is observed that CVAE and DINS perform similarly, while CVAE has a better success rate and DINS has a better normalized cost. However, it is shown in Fig. 9 that when the starting points are beyond maneuvering experience, the DINS degrades while the CAVE still maintains similar performance to starting points within maneuvering experience. The reason might be that in our case, DINS only biases the samples in the first 500 iterations, and it has to rely on the uniform sampler if no successful path is found in the first 500 iterations. In conclusion, CVAE performs well not only on the starting points within human experience but also on starting points outside human experience.

5 CONCLUSIONS

In this paper, a deep conditional generative model is utilized to learn the sampling distribution of docking operations from the human maneuvering experience. The model is used as the sampler to bias the sampling procedure in the RRT* algorithm, resulting in a fast converge to a feasible and optimal path. Human maneuvering data of a vessel in one year are extracted from two different ports to validate the approach. Experimental results show that the conditional variational autoencoder is able to learn from past experience and generate samples in the promising region for docking operations. The convergence speed to success rate and optimal cost of RRT* is improved by using this learned sampler. The method can also be generalized to unseen starting points for docking even though the sampling region from the learned sampler is not collision-free. Since an important aspect of docking is to reach the position at zero velocity, and the proposed method can be used for motion planning, future work includes motion planning for docking operations considering detailed ship dynamics models.

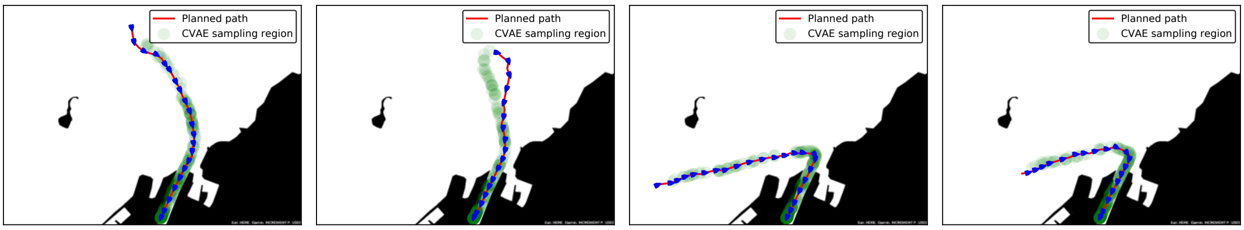


Figure 6: Examples of the path using RRT* with a learned CVAE sampler when the starting point is within human experience in the Trondheim harbor.

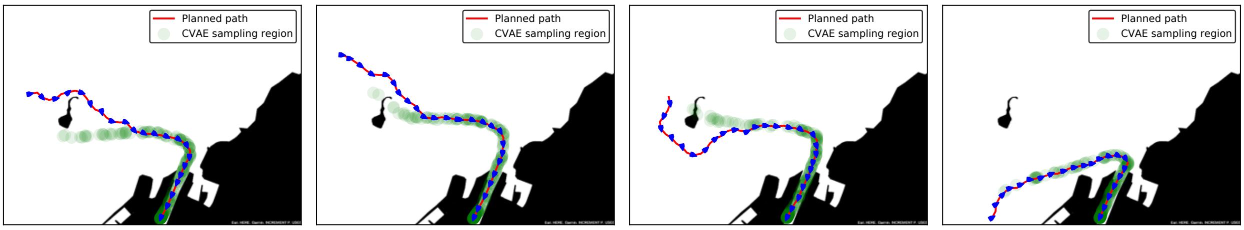


Figure 7: Examples of the path using RRT* with a learned CVAE sampler when the starting point is beyond human experience in the Trondheim harbour.

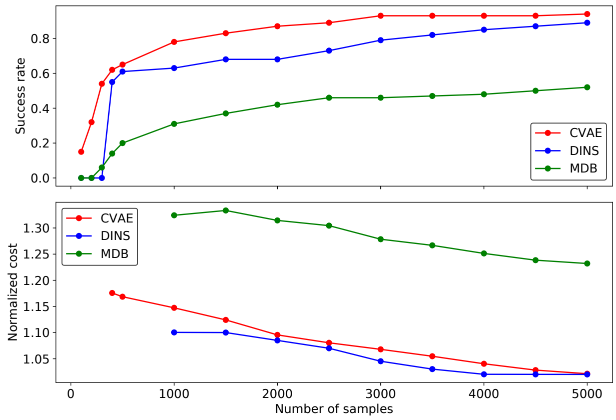


Figure 8: Comparison of the convergence rate when the starting point is within human experience in the Trondheim harbour.

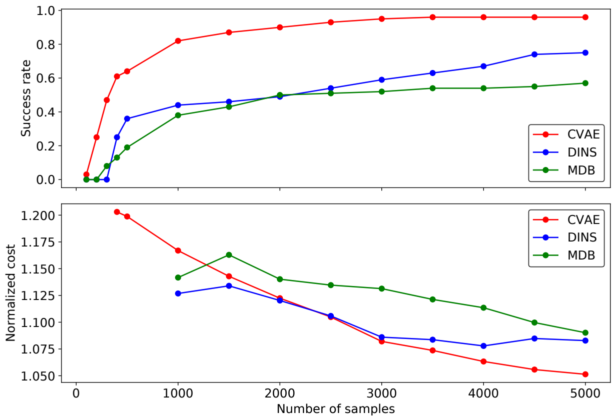


Figure 9: Comparison of the convergence rate when the starting point is beyond human experience in the Trondheim harbour.

Acknowledgments

This work is supported by a grant from the Research Council of Norway through the Knowledge-Building Project for industry "Digital Twins For Vessel Life Cycle Service" (Project nr. 280703).

References

Anderlini, E., Parker, G. G., and Thomas, G. Docking control of an autonomous underwater vehicle using reinforcement learning. *Applied Sciences*, 2019. 9(17):3456. doi:10.3390/app9173456.

Bergman, K., Ljungqvist, O., Linder, J., and Axehill, D. An optimization-based motion planner for autonomous maneuvering of marine vessels in complex environments. In *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, pages 5283–5290, 2020. doi:10.1109/CDC42340.2020.9303746.

Enevoldsen, T. T., Reinartz, C., and Galeazzi, R. Colregs-informed rrt* for collision avoidance of marine crafts. In *2021 IEEE Inter-*

national Conference on Robotics and Automation (ICRA). IEEE, pages 8083–8089, 2021. doi:10.1109/ICRA48506.2021.9560909.

Han, P., Li, G., Cheng, X., Skjong, S., and Zhang, H. An uncertainty-aware hybrid approach for sea state estimation using ship motion responses. *IEEE Transactions on Industrial Informatics*, 2022. 18(2):891–900. doi:10.1109/TII.2021.3073462.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *2017 International Conference on Learning Representations (ICLR)*. 2016.

Huh, J. and Lee, D. D. Efficient sampling with q-learning to guide rapidly exploring random trees. *IEEE Robotics and Automation Letters*, 2018. 3(4):3868–3875. doi:10.1109/IROS.2018.8593947.

Ichter, B., Harrison, J., and Pavone, M. Learning sampling distributions for robot motion planning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pages 7087–7094, 2018. doi:10.1109/ICRA.2018.8460730.

Jalonen, R., Tuominen, R., and Wahlström, M. Remote and autonomous shipsthe next steps: Safety and security in autonomous shippingchallenges for research and development. *Rolls-Royce, Buckingham Gate, London: The Advanced Autonomous Waterborne Applications (AAWA)*, 2016. pages 56–73.

Karaman, S. and Frazzoli, E. Incremental sampling-based algorithms for optimal motion planning. *Robotics Science and Systems VI*, 2010. 104(2).

Karaman, S. and Frazzoli, E. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 2011. 30(7):846–894. doi:10.1177/0278364911406761.

Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., and Teller, S. Anytime motion planning using the rrt. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, pages 1478–1483, 2011. doi:10.1109/ICRA.2011.5980479.

Kim, B., Kaelbling, L., and Lozano-Pérez, T. Guiding search in continuous state-action spaces by learning an action sampler from off-target search experience. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. 2018. doi:10.1609/aaai.v32i1.12106.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kuo, Y.-L., Barbu, A., and Katz, B. Deep sequential models for sampling-based planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pages 6490–6497, 2018.

LaValle, S. M. and Kuffner Jr, J. J. Randomized kinodynamic planning. *The international journal of robotics research*, 2001. 20(5):378–400. doi:10.1177/02783640122067453.

Martinsen, A. B., Lekkas, A. M., and Gros, S. Autonomous docking using direct optimal control. *IFAC-PapersOnLine*, 2019. 52(21):97–102. doi:10.1016/j.ifacol.2019.12.290.

Martinsen, A. B., Lekkas, A. M., and Gros, S. Optimal model-based trajectory planning with static polygonal constraints. *IEEE Transactions on Control Systems Technology*, 2021. doi:10.1109/TCST.2021.3094617.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Miyauchi, Y., Sawada, R., Akimoto, Y., Umeda, N., and Maki, A. Optimization on planning of trajectory and control of autonomous berthing and unberthing for the realistic port geometry. *Ocean Engineering*, 2022. 245:110390. doi:10.1016/j.oceaneng.2021.110390.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 2019. 32.

Qureshi, A. H., Simeonov, A., Bency, M. J., and Yip, M. C. Motion planning networks. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pages 2118–2124, 2019. doi:10.1109/ICRA.2019.8793889.

Qureshi, A. H. and Yip, M. C. Deeply informed neural sampling for robot motion planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pages 6582–6588, 2018. doi:10.1109/IROS.2018.8593772.

Shuai, Y., Li, G., Cheng, X., Skulstad, R., Xu, J., Liu, H., and Zhang, H. An efficient neural-network based approach to automatic ship docking. *Ocean Engineering*, 2019. 191:106514. doi:10.1016/j.oceaneng.2019.106514.

Skulstad, R., Li, G., Fossen, T. I., Vik, B., and Zhang, H. A hybrid approach to motion prediction for ship dockingintegration of a neural network model into the ship dynamic model. *IEEE Transactions on Instrumentation and Measurement*, 2020. 70:1–11. doi:10.1109/TIM.2020.3018568.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 2015. 28.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 2014. 15(1):1929–1958.

Tamar, A., Wu, Y., Thomas, G., Levine, S., and Abbeel, P. Value iteration networks. *Advances in neural information processing systems*, 2016. 29.

Wang, J., Chi, W., Li, C., Wang, C., and Meng, M. Q.-H. Neural rrt*: Learning-based optimal path planning. *IEEE Transactions on Automation Science and Engineering*, 2020. 17(4):1748–1758. doi:10.1109/TASE.2020.2976560.

Zaccone, R. and Martelli, M. A collision avoidance algorithm for ship guidance applications. *Journal of Marine Engineering & Technology*, 2020. 19(sup1):62–75. doi:10.1080/20464177.2019.1685836.