

Towards Autonomous Shipping: Benefits and Challenges in the Field of Information Technology and Telecommunication

I. Yanchin & O. Petrov

State Marine Technical University, Saint-Petersburg, Russia

ABSTRACT: This paper is dedicated to an overview of components of an onboard control system of an autonomous ship. This system controls and operates the ship. Therefore, this system needs to be able to analyze the ship's state, predict its future development and analyze the consequences of its own decisions. The paper focuses on software aspects of the onboard control system, not the hardware. The paper provides an overview of technologies that can be used to implement the components of such a system responsible for planning new routes, handling the ship during the voyage, ensuring its seaworthiness and safety during the voyage, monitoring an autonomous ship from an onshore control centre, ensuring the robustness of the onboard control system, and collective operations of multiple autonomous ships. The paper describes benefits the maritime industry would gain from deploying some of the technologies developed for autonomous ships on ordinary, human-controlled ships. The paper also describes some challenges, especially in the field of automatic decision and reasoning, arising from the emergence of autonomous and smart ships. The main contribution of the paper is that it summarizes existing research in different areas of autonomous ship technology.

1 INTRODUCTION

Autonomous shipping is no doubt a future of the maritime industry [21]. According to Rolls-Royce, a fully automatic ship will go to its first voyage by 2035 [21] which means the need for related research now. However, autonomous shipping poses a great challenge for information technology and research in the field of onboard control systems. In contrast with autonomous cars, autonomous ships will operation in much larger environments travelling by much longer distances. The following types of ships with regard to autonomy are defined in research: ordinary ships, where all decisions are made by the crew and all operations are performed by the crew, smart ships, where the crew still makes a decision, but the operations are performed automatically and autonomous ships where both decisions and

operations are automatic [12]. Smart ships are not guaranteed to have the crew on board. Such ships can be controlled from a remote control centre [6, 37]. However, it seems doubtful that smart ships with fully remote control can be implemented because such ships would require stable network connection with the control centre along the entire route of the voyage, which may be difficult to secure in the case of trans-ocean voyages [37]. Therefore, smart ships are expected to have some degree of autonomy to make them able to operate amid the absence of connection with the control centre. It is expected that an autonomous ship can also be controlled remotely, therefore there can be stages of fully autonomous operation and remotely controlled stages within a single voyage [35]. Remote control stages may be required to navigate the ship in the most dangerous or difficult circumstances, e.g. at nearshore areas and

at ports [37]. In this paper we focus on fully-autonomous ship operations, thus regarding autonomous ships and smart ships that operate in the autonomous regime.

This paper is dedicated to the challenges in the field of information technology and telecommunication with regard to smart and autonomous ships. It is expected that the success in these fields will significantly affect the maritime industry as being the most important milestone in the journey towards autonomous shipping [35].

Within this paper, we assume that an autonomous ship is controlled and monitored by a single onboard control system. This system runs on the onboard computers that are connected into the onboard computational network. The paper describes the design neither of the onboard control system nor the computational network, but refers to them when describing their components responsible for tackling the challenges arising from the emergence of autonomous and smart ships.

This paper is organized as follows. Section 2 provides an overview of techniques that can be used to plan safe and optimal routes for autonomous ships. Section 3 is dedicated to automatic ship handling, ensuring its seaworthiness and safety while following the route, keeping the ship on the route, and collective operations of multiple autonomous ships. Section 4 describes the remote monitoring of a ship. Section 5 is dedicated to the robustness of the onboard control system that controls the ship. Section 6 concludes the paper.

2 AUTONOMOUS NAVIGATION

Route planning is an important feature of the ship's onboard control system. The crew uses this feature to plot efficient routes taking into account weather conditions, legal regulations, fuel efficiency and other requirements [17]. If the system fails to plot an effective route or the resulting route is not suitable for some reason, the crew still can intervene and introduce corrections. In case of an autonomous ship, there is no crew to correct the inefficient route. Planning an efficient route for an autonomous ship is a more complicated task than in the case of an ordinary ship [6]. The onboard control system needs to be able to plan as efficient routes as possible without any human intervention.

Within this paper, we define an efficient route as a route that is the shortest route possible in terms of travel time that does not put the ship in danger in any way. This definition can be extended to assume additional requirements. For example, the route planning system can take fuel consumption into account, in this case, the system would try to achieve a compromise between travel speed (and thus travel time) and fuel consumption possibly by using slow steaming.

In case of an autonomous ship, there can be two approaches for planning a route. In the first case, the route is planned by the onboard control system of the ship, in the second case, the route is planned by a shore control centre and then transmitted to the ship

[36, 41]. In the latter case, the route could be evaluated by an expert before being transmitted to the ship [41]. In both cases, the input data for the route planning procedure is the destination point and the additional constraints for the route (e.g. fuel consumption, arrival time and so on).

Weather conditions affect the ship's safety and its ability to follow the route. Taking the weather into account not only helps to avoid negative effects on the ship [35,41] but also reduces the economical costs of a voyage by reducing the travel time and fuel consumption [16, 43]. Therefore, the ability to take weather conditions into account is essential for the route planning system [17, 23, 38].

One of the most common algorithms used for pathfinding is A*. This algorithm is able to find the shortest path between two vertices of a graph or grid. It is possible to tune the algorithm to take into account requirements specific for a particular task, e.g. weather conditions [17], ice [45] or in order to reduce the computation time [28]. However, A* is known to be highly dependant on the heuristic function used to determine which vertices to investigate and for its high memory consumption in case of the large search spaces (grid with many vertices) [33]. One can reduce the complexity by reducing the number of points the grid has, however, this reduces the freedom of the algorithm to choose movement direction thus reducing the optimality of the route.

Planning a route for a ship is a complex task that requires attention to many factors affecting routes safety and optimality. The navigators plan routes using their experience and background [35]. Building this experience into a formal route planning algorithm like A* or Dijkstra's is a difficult task. We argue that the maritime industry could benefit from employing algorithms and techniques that belong to the fields of Artificial Intelligence, Machine Learning and Deep Learning. We can divide these promising techniques into two groups: those that use a kind of agent that is trained from its own or external experience and those that are heuristic, fuzzy, or competitive approaches to find the solution for the task in question [30].

Two of the most well-known techniques of the first family are Reinforcement Learning and its variant, Deep Reinforcement Learning. Reinforcement Learning uses a table π , called policy, that helps understand why the agent took particular action being in a particular state, thus making the decision process interpretable. This table is designed during the preparation stage of the training process and must contain all of the possible agent states and all of the possible actions the agent can take. Deep Reinforcement Learning does not require any predefined states and table π , instead, it uses a deep neural network that takes state representation as input. Deep Reinforcement Learning has demonstrated good results in training an agent to play Go [39], chess [40] or video games [44], making it possible to beat a champion human player. However, in practice, even an expert often fails to design a good reward function for the agent, thus making it possible to maximize the reward without actually completing the task [22]. Moreover, currently, there is not enough research in the field of Reinforcement Learning about solving problems other than games [22]. Another

difficult here is that the training process assumes that the agent needs to observe all possible states and take all possible actions for any of them to test what happens, thus making the training process computationally difficult.

We argue that there is still not enough research to employ Reinforcement Learning to planning a route for a ship. Another obstacle on this path is that the agent needs an environment that it interacts with while solving the task [22]. As we have already pointed out, the agent needs to observe all possible situations, including those that lead to maritime accidents, moreover, it needs to observe these situation several times with slightly different properties. Therefore, using natural environments as training environments and real ships as agents to train the route planning system has unacceptably high costs, not to mention the related dangers. Another approach is to simulate the natural environment. In this case, all of the processes that take place in Nature and affect the ship are modelled and this modelled environment is used to train the agent. However, this approach requires the model to represent the real processes as precisely as possible so that the agent can use its knowledge gained from the simulation in the real-life tasks. This poses a difficult task.

We, therefore, argue that we should not use agents that are trained from their own experience for planning ship routes due to high costs of real experiments and extremely difficult development of simulators.

We, therefore, think that the route planning algorithms should be based on approaches that can adapt to a changing environment and can handle complex success criteria but do not require preliminary training. Genetic algorithms are one of such approaches [20]. The fitting function can be expressed as a weighted sum of values describing different route's properties thus making the algorithm to generate a good route through maximizing these value. Genetic algorithms can be effectively parallelized using the island model, taking advantage of distributed onboard computation systems [7, 18]. Since in every moment a genetic algorithm operates on a set of possible solutions, it is possible to get a set of Paretooptimal routes as a result and choose the final one using another examination technique [25]. Another advantage of genetic algorithms is that it is possible to restrict the total computation time allotted for planning the route and still get an acceptable sub-optimal solution [18].

As we have already said, navigators use their experience to judge on the route's quality. Experience is a very subjective and personal concept so it could not be formalized to be used by an automatic route planning systems. We propose to use a set of characteristic coefficients that can be used to measure route's properties thus making it possible to compare routes and select the best one. The fitting function of a genetic algorithm can be represented as a weighted sum of this coefficient values. These coefficients express how close the route is to the ideal route. Therefore, professional navigators could be interviewed about what is the optimal and safe route

and what is the ideal route, and then these qualities can be expressed through formalized coefficients.

Here we propose several basic coefficients that can be used as a starting point. This set can be extended with additional coefficients designed for a particular ship. Moreover, different routes for the same ship could be planned while taking into account different coefficients that were selected according to the cargo, technical conditions or other factors.

The safety coefficient can be defined as

$$S(r) = 1 - P_o^{(r)} \quad (1)$$

where $P_o^{(r)}$ is the probability of a negative encounter (e.g. collision, rough weather, severe pitching) to occur while the ship follows the route r . The shorter route is a better route, therefore the distance coefficient can be defined as

$$D(r) = \frac{|(s,d)|}{l_r} \quad (2)$$

where $|{(s,d)}|$ is the length of the line segment between start and destination points and l_r is the length of the route being examined. The closer the route to the straight line, the greater the value of D is. Since it is not possible to find a shorter way between two points than the straight line, this coefficient can be used to measure the optimality of the route with regard to its length. Another coefficient related to optimality is the time coefficient

$$T(r) = \frac{t_{|(s,d)|}}{t_r} \quad (3)$$

where $t_{|(s,d)|}$ is the time needed for the ship to travel along the $|{(s,d)}|$ straight line with the maximum technically possible speed and t_r is the time needed to travel following the route r . This coefficient helps ensure that the route is not only short in terms of distance but that it will not take long to follow it. According to [?], the number of manoeuvres the ship needs to perform during the voyage also affects the route's optimality and the most important manoeuvre is the change of direction. The change of direction usually happens at a waypoint that joins two edges of the route each of which defines new direction. Therefore, we can define the simplicity coefficient as

$$C(r) = \frac{2}{p_r}$$

where p_r is the number of waypoints in the route. Every route has at least two points: start and destination, and the simplicity coefficient encourages the route to have as little points as possible.

Route planning algorithm needs to be robust and able to handle unexpected situations. It also needs to be predictable and its output needs to be interpretable. That is why designing such an algorithm requires extensive testing and verification [35]. Formal algorithms like A* or Dijkstra's are proven to be able to find the best solution but they

pose strong requirements for the environment and the input data [11]. Non-formal algorithms like genetic algorithms may pose less strict requirements but there can be no formal proof of correctness. Therefore, testing and verification of such algorithms require simulation of the common and extreme situations in order to validate their behaviour. There are several techniques that can be used to verify and validate the correctness of models and algorithms while using simulation modelling, the Balchi's scheme is one of them [1, 2].

3 AUTONOMOUS SHIP HANDLING

After the ship has acquired the route, it needs to follow it. This means the onboard system needs to be able to handle the ship autonomously. This is expected to be the most difficult component of the onboard control system since it needs to take into account different aspects of ship handling, including, but not limited to, keeping the ship on the route and ensuring that the ships seaworthiness is satisfactory [35].

3.1 Keeping the ship on the route

Keeping the ship on the route is the first responsibility of the onboard control system when underway. The system analyzes the ship's motion and checks that the actual route does not deviate from the prescribed one more than allowed. In the case of discrepancy, the control system orders the propulsion system of the ship to perform actions to resolve this discrepancy [24].

Since it is expected that the prescribed route allows some sort of freedom (i.e. it does not prescribe the exact waypoints), another important aspect is planning the exact manoeuvres prescribed by the route [37]. For example, if a route prescribes the ship to go under a bridge, the onboard control system must issue orders to make the ship go under the bridge's arc by a safe distance from the columns and ensure that the arc is high enough. Although bridges, canal locks, and other infrastructure facilities can be detected during the planning stage, the actual route affects the moment of time when the ship reaches the facility and ship's exact position relative to it. That is why the ship needs to be able to plan the manoeuvres itself before performing them thus requiring the onboard control system to analyze the current situation and issue orders with regard both to the manoeuvre and the environment. Therefore, this component of the control system needs to be universal enough to be able to deal with different manoeuvres and surrounding situations.

It is expected that the onboard control system can be based on AI approaches [35] thus being able to handle situations that differ from those observed during the design stage.

The onboard control system can benefit from tree-based situation classification approach [31]. In case of such an approach, there is a set of classifiers organized as a tree. Each classifier takes the representation of the situation that includes, but is not

limited to, images from the cameras that capture images or video of the surroundings, radars that observe the surroundings, gyroscopes and accelerometers that measure the ship's state and other sensors. The root classifier that corresponds to the root vertex of the tree performs the initial classification of the situation, whether it corresponds to an open area, a bridge, locks, and so on. This initial classifier makes the top-level decision detecting which type the situation belongs to and does not detect its internal properties. Once the generic type of the situation is determined, the situation representation is passed to the next level of the tree. Classifiers of this level are specialized to observe situations of a specific type and are able to detect different subtypes of them. Figure 1 illustrates the tree-based classification approach.

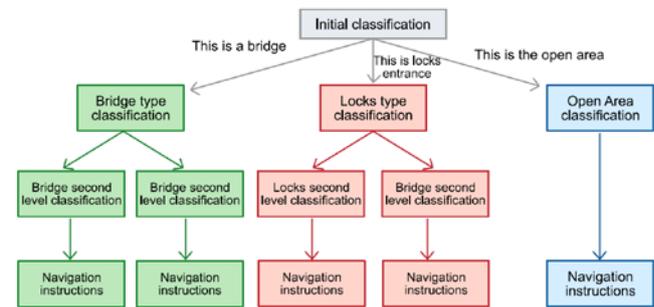


Figure 1. Sample tree-based situation classifier. Each level refines the output of the previous one until the final decision is made.

Therefore, each level of the classification tree refines the decision made by the previous level thus enabling the onboard control system to capture all important properties of the situation [31]. The refinement process finishes when the tree traversal process reaches a leaf that contains navigation instructions suitable for the detected situation [31]. The use of a classification tree helps to keep each classifier specialized on a particular task thus making it easier to design it, train it, maintain its operation, interpret the output or insert new tree branches (thus making the tree aware of new situations) leaving other branches intact. In addition, the classifiers can be independently replaced or updated according to the requirements.

The classifiers could be based on Deep Neural Networks or use simpler machine learning models suitable for the task being solved by the classifier. Convolution Neural Networks have demonstrated good results when processing image and video data or other spatially distributed values [15]. If the inputs for the network contain both image and non-image data, it is possible to train a two-branch neural network (figure 2), where one branch handles image data and another one non-image data and then they join their outputs for further processing. Twobranch neural networks have demonstrated good results when processing mixed data of different nature [46].

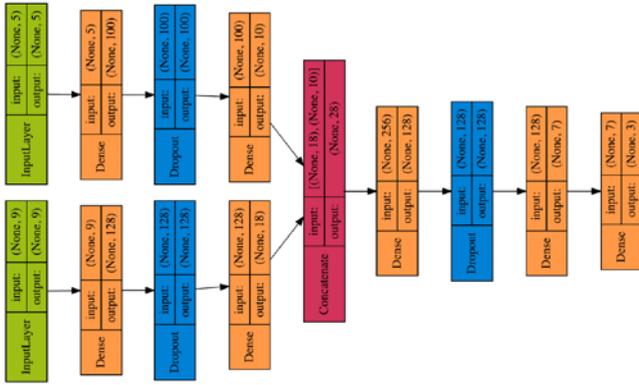


Figure 2. An example of neural network with two branches.

3.2 Ensuring the route correctness

As we have already said, the route's quality is expressed through a set of characteristic coefficients that describe the route's properties. These coefficients can also be used to ensure that the actual route does not deviate from the prescribed one in a way that may lead, in the long run, to an accident.

Consider the prescribed route and the actual route that is being plotted during the voyage. In order to check that the actual route is not going to cause an accident we can compute its characteristic coefficients during the voyage taking into account the actual locations of the waypoints and their properties, and then compare these coefficients to those computed for the prescribed route. This lets to determine which properties of the actual route are worse than expected. Another approach is to plot the sum of the characteristic coefficients and thus decide whether the trend is positive or stable (the route quality either increases or stays the same) or negative (the route quality decreases) [30]. Figure 3 shows an example of a route quality plot.

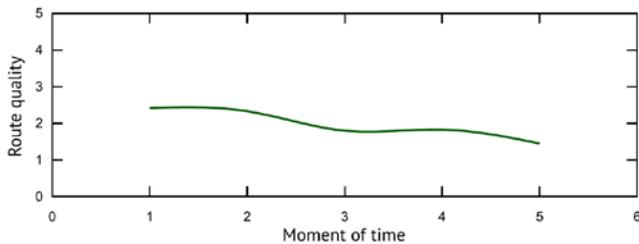


Figure 3: Route quality decreases over time, this indicates negative trend.

According to the plot 3, the route quality decreases with time, which means that if the ship does not re-plan the route, an accident may happen [30].

3.3 Ship collective intelligence

Autonomous and smart ships are expected to be equipped with an onboard control system that makes all of the decision related to ship handling. As inputs, this system takes data from the sensors installed onboard. In some situations, however, an autonomous ship needs to take into account the actions of the surrounding ships in order to maintain safety and comply with COLREGs [13, 35]. Consider

two possible cases: in the first case there are only ships that operate fully automatically, and the second case when at least one of the ships is operated by humans. In both cases, the ships need to pass safely through a particular area.

In the first case, when there are only smart ships, these ships can communicate in order to infer a decision [36]. The main difficulty here is to make all of the ships to come to the same decision, i.e. to make all of the ships agree with it. All ships are considered equal, i.e. there are no primary ships that make decisions and secondary ships that obey. In this case, they can use a distributed consensus algorithms like PAXOS [26] or Raft [34] that are widely used in clustered and distributed computing. These algorithms help a set of distributed computing nodes get the same result for a particular task, thus coming to a consensus. These algorithms can be used by autonomous ships to come to a consensus about their actions in the case of an encounter. Both PAXOS and Raft guarantee that all distributed nodes end up with the same state of the finite automaton that forms the core of the algorithms. This means that all of the ships in question are able to agree on how to behave in order to safely pass through the area.

In the second case, the autonomous ships and ordinary ship also need to communicate in some way in order to agree on their actions [36]. However, in this case, the ordinary ship is operated by humans that need to be able to communicate with machines operating other ships. This issue can be resolved by installing onto ordinary ships a special communication unit that is able to communicate with smart ships (standardization required). The crew uses this unit to explain their plans on the safe passage through the area in question so that the autonomous ships are aware of the human's decisions and thus can infer their own. The smart ships can still use PAXOS and Raft to come up with a decision taking into account human's opinion. Another approach is to make this onboard unit a part of the consensus framework. The ordinary ship also takes part in the decision process and is considered as a member of a distributed consensus network. However, in this case, the crew of the ordinary ship needs to obey the collective decision.

3.4 Ensuring the ship's seaworthiness

One of the most important tasks the crew needs to care about is the ship's seaworthiness. Here we concentrate on seakeeping performance criteria, i.e. the ability of a ship to effectively and safely deal with the environmental conditions, i.e. pitch and roll, buoyancy and stability. In the case of an ordinary ship, the crew is responsible for maintaining these criteria, but in the case of an autonomous or smart ship, this needs to be done automatically. Since there is no crew on board, no one can perform actions to prevent an accident that is about to happen. The onboard control system should preliminarily detect such situations and take actions not just to prevent them, but to avoid the related risks [30]. There is a known controllability issue: under some circumstances, the ship may not be under a full control [35], therefore the onboard control system

needs to take into account that the issued orders may be performed only partially, or have no effect. Another difficulty is the uncertainty in the ship's state data: the data that is thought to be actual and correct may be out of date or incomplete [42].

Waves affect the seakeeping characteristics inducing rolling and pitching, which, in turn, affect the ship's stability. Waves are highly stochastic, non-stationary process and therefore predicting its parameters is a difficult task [8]. Although wave parameters can be drawn from a wave spectra [19], each spectrum provides approximate results and thus needs to be tuned for a particular area [4]. The tuning requires long time series of wave parameters, which could not be collected during the voyage. Moreover, a ship, in general, does not have the machinery required to measure wave parameters, e.g. period or wavelength. The possible solution is to install accelerometers and gyroscopes onboard and directly measure rolling and pitching parameters. The data received from these sensors form a time series that can be analyzed with Deep Neural Networks, possibly predicting future parameters [10,48], anomaly detection techniques or other approaches to identify patterns and make decisions accordingly.

Hull consistency affects the ship's buoyancy and stability. In case of a breach, the hull's compartment gets flooded thus changing the list angle and reducing ship's stability. Additional water volume inside the hull also reduces the ship's buoyancy. Therefore it is important that the onboard control system is able to monitor the hull's integrity and compartment flooding. This can be achieved using Hull Monitoring System [14] and flooding detection sensors installed in all compartments. Hull Monitoring System consists of several sensors located around the hull measuring tensions and stress of the hull caused by the sea state [14, 47]. The collected data is used to analyze the effects of the sea state and the ship's movements. The data can be used to predict possible breaches caused by extreme wave effects, e.g. slamming and detect existing breaches. Flooding detectors provide the data whether a compartment is flooded and to what extent. The onboard control system uses the data from the Hull Monitoring System and flooding detectors to ensure the hull's integrity. In case of flooding the control system uses stability curves to select the appropriate actions (e.g. to flood the opposite compartment).

Since Machine Learning is known to be able to handle data with missing values [10] (e.g. because of temporary failure of a sensor) and adapt to changes in data patterns happened since training [27], it is expected that Machine Learning and Deep Learning will form a general framework for the analysis of the ship's state needed to ensure its seaworthiness. To be able to analyze its state the ship needs to measure its static and dynamic motion parameters, like roll angles, roll acceleration, and amplitude. Therefore, it should be equipped with sensors, accelerometers and detectors and Hull Monitoring System that measure the ship's state. These sensors form the digital model of the ship. With this model available it is possible not only to analyze the instant ship's state but also to predict the future development of the current situation [32].

4 AUTONOMOUS SHIP REMOTE MONITORING

In order to ensure seaworthiness, the smart and autonomous ships will be equipped with a great number of sensors that measure their dynamics. This data can be transmitted to the onshore control centre thus making it possible to remotely analyze the ship's behaviour [12, 13, 37]. Because of the distributed nature of the onboard ship machinery, a ship can be thought of as an IoT-enabled multiagent system where each machinery unit is an independent agent [35]. The amount of data generated by these IoT agents for a single ship is expected to make up a large dataset [35]. If a shipowner has a fleet of smart ships, the amounts of data increase by orders of magnitude. However, in the latter case, the shipowner can benefit from the large datasets collected from the entire fleet by getting the overall statistics computed using the data from all of the ships, which increases prediction accuracy. This, in turn, requires the ship owners to have a Data Science or Data Engineering departments to successfully handle, analyze and gain insights from this data using Big Data and Machine Learning techniques. It is expected that the entire marine industry would benefit from these datasets and related software [3].

Collecting datasets with the data related to the ship's operations enables ship owners to employ maintenance prediction techniques [6, 37]. These techniques let the ship owners analyze the behaviour of the onboard machinery and predict whether it is going to go out of order. This lets the owners optimally plan the maintenance operations [35, 37]. Actually, such a monitoring system can be installed on an ordinary ship and the monitoring data can be available both to the shipowner and the crew.

5 ONBOARD CONTROL SYSTEM ROBUSTNESS

The main source of trouble onboard is the crew [12]. Since there is no crew on board, its effect on the ship's safety is reduced, making technical issues the main concern. In the case of an autonomous ship, the onboard control system becomes a single point of failure [13]. If the onboard control system goes out of order, the ship is not able to operate. Since the ship has no crew on board, there is no way to repair the failed systems. Therefore, the onboard control system should be able to detect its own failures and either entirely recover from them or rebalance the workload in such a way that the components that are still alive can keep the system operating.

To ensure robustness, the development team needs to use distributed approaches, where a computer system consists of several network nodes connected to accomplish a single task. If a single node goes out of order, another one takes its workload. One possible solution for that is to adopt the multi-agent approach, also known as actor approach. There are different actor implementations available for different programming languages and environments [5, 9, 29].

In the case of the actor approach, the software system is represented as a set of independent components (called actors). The actors communicate by sending messages, not through direct interaction

by function calls like in traditional software systems. Since the actors communicate through message passing, different actors can be located on different network nodes [5] enabling effective development of distributed applications. Here, the sender actor does not need to know where (on which network node) the receiver is located and the underlying actor framework hides all the complexity of the network communication [5,9] making it easier for the developer to concentrate on the domain, not infrastructure.

The distributed nature of applications built using the actor approach makes them more error-resistant than traditional centralized applications. Once a network node goes down, the actor framework detects this failure and deploys the actors that have been on the failed node on the nodes that are still alive and routes the messages to the new instances [5]. Each actor is responsible for persisting its internal state and loading it once it is deployed, thus making the application survive through node failures without data losses.

Since the ability to operate regardless of the failures is essential for the onboard control systems of autonomous ships, we argue that they can benefit from adopting the actor approach making them more error-resistant and letting to gracefully recover from failures. It is important to distribute the network nodes across the entire ship in order to ensure that physical damage or flooding of a single compartment does not lead to the destruction of the entire network.

Nevertheless, the actor model is not able to overcome hardware failures if the particular hardware exists as a single instance. Therefore, sensors that are used to measure the ship's state need to be duplicated in order to increase their robustness.

However, it is still possible that the onboard control system or the onboard machinery goes out of order. It is expected that in this case, a ship should activate a kind of fail-to-safe mode, that lets it safely operate using the machinery it still has and wait for the assistance from the onshore services [37].

6 CONCLUSION

There is still much to do before an ocean-going autonomous ship goes to its first voyage. This means the need for research on topics related to ship architecture, legal regulations, port operations, information technology, and others. In this paper, we have discussed the information technology aspects of the autonomous and smart ships. It is expected that such ships will have an onboard control system responsible for planning route, handling the ship during the voyage and monitoring its state.

We argue that the route planning component should not be implemented using techniques that require training because this needs a great number of real-life experiments including those that can lead to an accident; otherwise, it requires a complicated simulation environment that precisely models the ocean or the agent knowledge would be wrong.

Ship handling during the voyage is expected to be based on intelligent techniques that analyze the ship's state and the environment around it making decisions on what actions to perform in order to keep the ship on the route and maintain its seaworthiness.

The need for the onboard control system to monitor the ship's state and the environment means that the autonomous and smart ships will be equipped with a great number of sensors that capture the properties of the ship's movements and the surrounding environment. This enables for remote monitoring of the ship's state and actions thus making it possible to make decisions based on collected datasets. This also enables predictive maintenance thus reducing the downtime due to spare part delivery and reducing costs. The maritime industry will also benefit from the large datasets collected during autonomous ship operations.

In order to ensure the robustness of the onboard control system, it can be developed using the actor approach thus making it inherently parallel and distributed.

Most of the technology needed to support the onboard control system already exists but needs to be adapted to the requirements of the maritime industry to make the future of ocean transportation closer.

REFERENCES

- [1] Osman Balci. Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of Operations Research*, 53(1):121– 173, Dec 1994.
- [2] Osman Balci. Verification, validation, and accreditation. In *Proceedings of the 30th Conference on Winter Simulation, WSC '98*, pages 41–4, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.
- [3] C.L. Benson, P.D. Sumanth, and A. P. Colling. A quantitative analysis of possible futures of autonomous transport. In *INEC 2018 Conference*, 2018.
- [4] Lars Bergdahl. Comparison of measured shallowwater wave spectra with theoretical spectra. *Proceedings of the 8th European Wave and Tidal Energy Conference*, Uppsala, Sweden, 2009, 2009.
- [5] Phil Bernstein, Sergey Bykov, Alan Geller, Gabriel Kliot, and Jorgen Thelin. *Orleans: Distributed Virtual Actors for Programmability and Scalability*. March 2014.
- [6] Mogens Blanke, Michael Henriques, and Jakob Bang. *A pre-analysis on autonomous ships*. Technical University of Denmark, 2016.
- [7] E. Cantu'-Paz. A survey of parallel genetic algorithms. *CALCULATEURS PARALLELES, RESEAUX ET SYSTEMES REPARTIS*, (10), 1998.
- [8] A. Castro, R. Carballo, G. Iglesias, and J.R. Rabun'al. Performance of artificial neural networks in nearshore wave power prediction. *Applied Soft Computing*, 23:194 – 201, 2014.
- [9] Dominik Charousset, Thomas C. Schmidt, Raphael Hiesgen, and Matthias Wahlich. Native Actors– A Scalable Software Platform for Distributed, Heterogeneous Environments. In *Proc. of the 4rd ACM SIGPLAN Conference on Systems, Programming, and Applications (SPLASH '13), Workshop AGERE!*, pages 87–96, New York, NY, USA, Oct. 2013. ACM.
- [10] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

- [11] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2009.
- [12] John F. Cross and Gordon Meadow. Autonomous ships 101. *Journal of Ocean Technology*, Vol 12:23– 27, 09 2017.
- [13] V. I. Dmitriev and V. V. Karetnikov. Methods of ensuring the safety of navigation when implement unmanned technology (in russian). *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 9.6 (2017), 2017.
- [14] Global Maritime Engineering. Hull Stress Monitoring System Electric Sensors' Solution. Measure stresses, torsions, caused by Waves, Cargo Operations and Motions.
- [15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [16] M. Grifoll, F. X. Mart'nez de Ose's, and M. Castells. Potential economic benefits of using a weather shiprouting system at short sea shipping. *WMU Journal of Maritime Affairs*, 17(2):195–211, Jun 2018.
- [17] Manel Grifoll, Llu'is Martorell, Marcel·la Castells, and Francesc Xavier Mart'nez de Ose's. Ship weather routing using pathfinding algorithms: the case of Barcelona – Palma de Mallorca. In *XIII Congreso de Ingenieria del Transporte*, pages 1–11. Universidad de Oviedo, 2018.
- [18] A. Hassani and J. Treijs. An overview of standard and parallel genetic algorithms. *IDT workshop on interesting results in computer science and engineering*, 2009.
- [19] Klaus Hasselmann, T. Barnett, E. Bouws, H. Carlson, D. Cartwright, K. Enke, J. Ewing, H. Gienapp, D. Hasselmann, P. Kruseman, A. Meerburg, Peter Muller, Dirk Olbers, K. Richter, W. Sell, and H. Walden. Measurements of wind-wave growth and swell decay during the joint north sea wave project (jonswap). *Deut. Hydrogr. Z.*, 8:1–95, 01 1973.
- [20] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975. second edition, 1992.
- [21] Rolls-Royce Ship Intelligence. *Autonomous ships. the next step*. Rolls-Royce Marine, 2016.
- [22] Alex Irpan. Deep reinforcement learning doesn't work yet. <https://www.alexirpan.com/2018/02/14/r-hard.html>, 2018.
- [23] Hinnenthal Jo'rn and Harries Stefan. A systematic study on posing and solving the problem of pareto optimal ship routing. In *3rd International Conference on Computer and IT Applications in the Maritime Industries, COMPIT'04*, pages 27–36, 2004.
- [24] V. V. Karetnikov, I. V. Pashchenko, and A. I. Sokolov. Prospects of introducing unmanned navigation on inland waterways of the Russian Federation (in russian). *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 9.3 (2017), 2017.
- [25] Abdullah Konak, David W. Coit, and Alice E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992 – 1007, 2006. Special Issue Genetic Algorithms and Reliability.
- [26] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, jul 1978.
- [27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436 EP, May 2015.
- [28] R. Leigh, S. J. Louis, and C. Miles. Using a genetic algorithm to explore a*-like pathfinding algorithms. In *2007 IEEE Symposium on Computational Intelligence and Games*, pages 72–79, April 2007.
- [29] Hugh McKee and Oliver White. *Akka A to Z. An Architect's Guide To Designing, Building, And Running Reactive Systems*. Lightbend, Inc., 2018.
- [30] Yu. I. Nechaev. *Catastrophe theory: the modern approach to decision-making* (in Russian). ArtExpress, Saint-Petersburg, 2011.
- [31] Yuri I. Nechaev and Oleg N. Petrov. Intellectual technology for control of dynamic unsinkability on the unmanned vessels (in Russian). *Marine intellectual technologies*, 3(1), 2020.
- [32] Nikitas Nikitakos. *Fourth industrial revolution in maritime sector*. University of the Aegean, 2019.
- [33] Masoud Nosrati, Ronak Karimi, and Hojat Allah Hasanvand. Investigation of the * (star) search algorithms: Characteristics, methods and approaches. *World Applied Programming*, 2(4):251–256, April 2012.
- [34] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 305–319, Philadelphia, PA, jun 2014. USENIX Association.
- [35] Lokukaluge Perera. Deep learning towards autonomous ship navigation and possible colregs failures. *Journal of Offshore Mechanics and Arctic Engineering*, 05 2019.
- [36] Alexandr Pinskiy. E-navigation and autonomous ship handling (in russian). *Transport of the Russia Federation. A journal about science, practice, economics*, 2016.
- [37] MUNIN Project. *Research in maritime autonomous systems project results and technology potentials*, 02 2016.
- [38] Wei Shao, Peilin Zhou, and Sew Kait Thong. Development of a novel forward dynamic programming method for weather routing. *Journal of Marine Science and Technology*, 17(2):239–251, Jun 2012.
- [39] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016.
- [40] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815, 2017.
- [41] Martin Hjorth Simonsen, Erik Larsson, Wengang Mao, and Jonas W. Ringsberg. State-of-the-art within ship weather routing. In *ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering*, page 11, 2015.
- [42] Alexander N. Suslov, Olga V. Odegova, and Sun Tanshi. Approaches for the stability automation monitoring creation (in Russian). *Marine intellectual technologies*, 3(1), 2020.
- [43] K. Takashima, B. Mezaoui, and R. Shoji. On the Fuel Saving Operation for Coastal Merchant Ships using Weather Routing. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, 3:401–406, 2009.
- [44] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michae'l Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Re'mi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wu'nsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using

- multi-agent reinforcement learning. *Nature*, 575(7782):350–354, Nov 2019.
- [45] Anna Ja. Voitkunskaia and Petr N. Zvyagin. A routing of a vessel in drifting ice (in russian). In *Marine Intellectual Technologies. Proceedings of the Saint-Petersburg State Marine Technical University.*, volume 2 (40), 1, pages 166–172, 2018.
- [46] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *CoRR*, abs/1704.03470, 2017.
- [47] Wärtsilä. Stress monitoring system, hull stress surveillance system. In *Wärtsilä Encyclopedia of Marine Technology*, 2020.
- [48] Haimin Yang, Zhisong Pan, and Qing Tao. Robust and adaptive online time series prediction with long short-term memory. *Computational Intelligence and Neuroscience*, 2017:1–9, 12 2017.