

Keywords: collision avoidance system; autonomous ship; database about ship skills; communication on boards of autonomous ships; artificial intelligence (AI)

Vladimir BROZOVIĆ¹, Danko KEZIĆ², Srećko KRILE^{3*}, Filip BROZOVIĆ⁴

A HARDWARE PLATFORM FOR A MARITIME COLLISION AVOIDANCE SYSTEM

Summary. A maritime collision avoidance system built on a database is being developed. Different information important for navigation, like electronic charts, bathymetric data, sea current and forecast weather conditions, etc., as well as knowledge about weather and load-dependent maneuvering capabilities of the involved vessels, are entered into the database tables. When using the system in real-time, the targeted data must be found very quickly in the database. In parallel, for the prediction of ship courses and for the calculation of evasive courses, the system must be capable of solving very many calculations in the shortest possible time, and these calculations often use database tables. Fast communication for updating database tables with the latest data is equally important in such a system. This article briefly presents the hardware developed for the project. We designed such data models for the efficient application of the Postgres database, capable of real-time calculation.

1. INTRODUCTION

The article deals with a *collision avoidance system* (CAS) for ships in the hardware and software segment. The main possibility is that the system can generate suggestions for the commander of the ship as well as produce electrical signals (commands) as input for rudders and propulsion systems (bow and stern thrusters) in the event of a collision threat. In the first case, it could be a significant help for crews in collision situations, or it could improve *the vessel traffic system* (VTS). In the second case, it could be an important step toward the autonomous ship. Different knowledge should be integrated into the system and should be retrievable in algorithms at any time. Such approach offers possibility the whole thing to be built on a database. In this way, the system knowledge can be easily extended with new data at any time. The stored knowledge can then be searched with well-developed database search algorithms according to different search criteria. With deeper knowledge of search principles, the search planer can be influenced via parameters, and the search process can be sped up in this way. Certain requirements must be met on hardware in order for these tasks to be performed in real-time. One of the most important elements is ship maneuvering skills, which are crucial for the correctness of further calculations. Shipping skills on the scene and potential capabilities for collision avoidance are defined by many factors. A database was created to store the maneuvering skills of the individual ship in certain circumstances, which enables adaptation to real possibilities in maneuvering in certain situations in the navigation area.

¹ SEAL AG; Landstrasse 176, 5430 Wetztingen, Switzerland; e-mail: v.brozovic@seal.ch; orcid.org/0009-0004-3809-2227

² University of Split, Faculty of Maritime Studies; Ruđera Boškovića 31, 21000 Split, Croatia; e-mail: danko@pfst.hr; orcid.org/0000-0003-2055-8039

³ University of Dubrovnik, Department of Electrical Engineering and Computing; Ćira Carića 4, 20000 Dubrovnik, Croatia; e-mail: srecko.krile5@gmail.com; orcid.org/0000-0002-0748-1359

⁴ SEAL AG; Landstrasse 176, 5430 Wetztingen, Switzerland; e-mail: fbrozovic@seal.ch; orcid.org/0009-0009-7791-8569

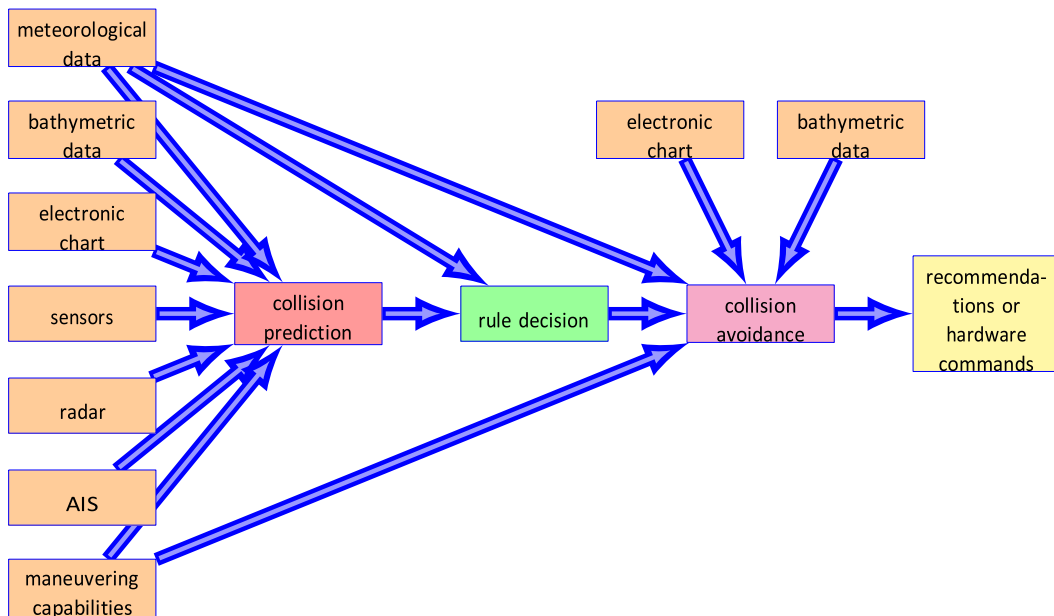
* Corresponding author. E-mail: srecko.krile5@gmail.com

Also, we organized a special database where data about meteorological conditions can be forecasted. Postgres is a unique database that, with its software modules, solves all database tasks during the operation of the expert system.

In Section 4, we present the data model set up for ship maneuvering capabilities and their references in more detail. In Section 5, we explain the data model to forecast weather conditions. In Section 6, we explain the maneuvering capabilities of the ship in particular situations connected with the dependencies of the ship on neighboring ships, speed and course, with the last refreshment of data about meteorological conditions. Section 7 describes the developed hardware and is the main part of this presentation.

2. SYSTEM OVERVIEW

Fig. 1 presents the expert system for collision detection and collision avoidance of the ships on the sea. There are different ways to solve that problem, but a similar solution is given by Huang et al. [1].



Source: Authors

Fig. 1. Architecture of expert system for collision detection and avoidance system. Source: Authors

Input data comes from various peripheral units, such as sensors or other main and auxiliary navigation systems, including nautical charts, environmental condition detectors, and meteorological forecasts. We distinguish three main functional blocks. The first one is for collision detection, where the possibility of a collision is predicted based on the prediction of the direction of movement and speed of ships in a certain area (i.e., it predicts the intercrossing of ship routes). More about it can be found in [2] and [3]. The second block deals with selecting collision avoidance rules related to a certain collision situation (see [4-7]). The third block is crucial for collision avoidance and calculates new shipping routes to ensure avoidance of potential collision. All input data flows into the expert system (orange boxes) are performed via the database. Therefore, the system is designed to be adapted to different data formats that we need to transfer from a particular peripheral unit. Postgres is a software database with the PostGIS extension used throughout the project. Detailed looks into Postgres and PostGIS can be found in [8] and [9], respectively.

3. DATA MODEL FOR NAUTICAL CHARTS

If we have to store all data flows (orange boxes) presented in Fig. 1 in the database, it is necessary to define the corresponding data models. In the model design for each individual data group, care was always taken to ensure that the data of this group could be efficiently searched in the database according to the most frequent search criteria for this group.

The first data model is for nautical charts. The digital chart data in the new S-101 format standard as a part of a new S-100 standard group [10] is stored in the database for use in the CAS. More about S-101 can be found in [11]. A short S-100 overview is given in the authors' conference presentation [12]. These chart data are used in the system in various algorithms (e.g., in the maximum likelihood predictor) for the optimal prediction of the path of a potentially dangerous ship or in the calculation of the avoidance route in case of danger of collision. The storage data model used for nautical charts in this project is well described in the authors' conference presentation [13].

4. DATA MODEL FOR MANEUVERING CAPABILITIES

4.1. Multidimensional Interpolation

We can use linear interpolation to efficiently and quickly calculate the value at the specified operating point from the database access data. To create functions, we use the Taylor series with m variables according to the formula below:

$$F(x_1 + h_1, x_2 + h_2, \dots, x_m + h_m) = F(x_1, x_2, \dots, x_m) + \left(h_1 \frac{\partial}{\partial x_1} + h_2 \frac{\partial}{\partial x_2} + \dots + h_m \frac{\partial}{\partial x_m} \right) F(x_1, x_2, \dots, x_m)$$

If there are many records in the database for non-linear dependencies, we can apply linearization of the interpolation. To replace the non-linear dependence, we use a series of linear segments and thus linearize the interpolation. It is necessary to enter a large number of records into the database. Such algorithms use these possibilities with a safety factor of 0.3.

4.2. Database *ship_skills*

The *ship_skills* database contains the ship's maneuvering capabilities, and it is composed of five tables. These tables are presented very rudimentarily in the following subsections. For more details, see the reference presentation [14]. The input data for each table are the function values of F_i in particular working points (x_1, \dots, x_m) and, respectively, the partial derivative. This means that the interpolation can be calculated with the formula from Section 4.1.

The main characteristic of the applied model is that new details can be easily added to the *ship_skills* database, which will not reduce the search speed.

i. Table *ship*

In this table, we have records with basic data about all known ships. The number of ships is virtually unlimited.

ii. Table *skills*

The data in this table indicate the ship skills under conditions given in the same entry. The corresponding ship is identified via index from I.

iii. Table *meteo*

Each entry in this table identifies a combination of the meteorological conditions for a particular ship. Entries from this table are referenced via index from entries from ii.

iv. Table *gradients*

In this table, we have partial derivatives that we use for multidimensional interpolation, which we mention in Section 4.1.

v. Table *costs*

In this table, we have records for several costs about way corrections; they are referenced via index from entries from ii (table *skills*).

Of course, the data in the presented tables can be reduced only to vessels that may appear in waters where such an expert system is used.

5. DATA MODEL FOR FORECASTING METEOROLOGICAL CONDITIONS

We use several tables to represent all meteorological values in the database. These tables are explained below.

5.1. Database table *meteo_conditions_range*

In this table, we store data about indices for a much larger amount of data stored in tables that represent the next values:

- *wind_values*,
- *wave_values*,
- *current_values*,
- *swell_values*.

Records from this table specify the value tables, the indices in the value tables, geographical area, and validity time interval.

An example of such an entry in this range table is as follows: Table *wind_values*, longitudes from 16.2 to 17.4, latitudes 42.5 to 43.8, times (epoch time) 1701966552 to 1701986552 and indices 189234 to 199234.

5.2. Database tables with forecasted values

In these tables, the geographical coordinates are stored for individual points with respective data on meteorological conditions and the associated time for which they are valid.

6. DATA MODEL FOR DETERMINING THE SHIP'S CAPABILITIES BASED ON INPUT DATA

The maneuvering skills for a particular position on the route (space) and for each moment (time) should be calculated using further steps:

1. The input parameters are:
 - position coordinates (*long*, *lat*),
 - moment of time (*given_time*) and
 - unique identification of the ship (*MMSI* or *IMO*).
2. The most important database is that in which the meteorological data is stored. For given coordinates at a given time, particular values are searched. This happens in the following way:
 - (i) In Table 5.1, for each value, we need to find that for which the following condition is fulfilled: $given_time \in [start_time, end_time]$.
 - (ii) From the last step, for the given point coordinates (*long*, *lat*), the following conditions are valid:

$$(area_long_west \leq long \leq area_long_east) \wedge (area_lat_south \leq lat \leq area_lath_north).$$
 - (iii) Because the positions and times of the route points differ in most cases from the positions and times in a meteorological forecast, some procedures have to be done respectively. The meteorological values for the route point are first calculated using *inverse distance weighting* (IDW) interpolation for two neighboring time points and then from these values we are using linear interpolation for the desired time point. If meteorological data in tables (mentioned above) was sorted with the presumption that the longitude of input points are in increasing order and then sorted by time, four points have to be searched ($long_{i,t}$, $lat_{i,t}$), $i \in [1,4]$, forming the particular geographic area surrounding the given point (*long*, *lat*).

- (iv) We use the IDW (see [15] and [16]) to calculate the meteorological values for each coordinate (*long*, *lat*). On the values from the step above, the following interpolation formulas are performed:

$$F_{t_1} = \frac{\sum_{i=1}^4 \frac{1}{d_i} F_{i,t_1}}{\sum_{i=1}^4 \frac{1}{d_i}} \text{ and } F_{t_2} = \frac{\sum_{j=1}^4 \frac{1}{d_j} F_{j,t_2}}{\sum_{j=1}^4 \frac{1}{d_j}}$$

$$d_i = \text{distance}_f((\text{long}, \text{lat}), (\text{long}_{i,t_1}, \text{lat}_{i,t_1})), i \in [1,4]$$

$$d_j = \text{distance}_f((\text{long}, \text{lat}), (\text{long}_{j,t_2}, \text{lat}_{j,t_2})), j \in [1,4]$$

ST_Distance from PostGIS is used as *distance_function*.

- (v) The meteorological value *F* for a particular moment *given_time* is calculated as follows:

$$F(\text{given_time}) = \frac{\frac{F_1}{\text{given_time} - t_1} + \frac{F_2}{t_2 - \text{given_time}}}{\frac{1}{\text{given_time} - t_1} + \frac{1}{t_2 - \text{given_time}}}$$

The formula for linear interpolation is as follows:

$$F(\text{given_time}) = F_1 + (\text{given_time} - t_1) \frac{F_2 - F_1}{t_2 - t_1}$$

3. In the next step, we need to interconnect data from the step above with the ship's *course over ground* (COG). The results are represented as meteorological conditions (wind, current, waves and surf) corresponding to the ship's course. Explanations of the programming techniques are given in [17] and [18].
4. The maneuvering skills we can get from the database in the next steps are:
 - (i) The *ship_id* represents characteristics of a particular ship for a given *MMSI* or *IMO* number. The data *calc_draft_percentage* can be found from the data *min_draft*, *max_draft* and the *current_draft*.
 - (ii) All values for *ship_id* have to be detected in the table *skills*, where *draft_percentage* is greater than/equal to the calculated *calc_draft_percentage*. Then, the detected values have to be sorted by *draft_percentage* (first criteria) and *vessel_speed* in increasing order.
 - (iii) The found entries are used only if *draft_percentage* is equal or immediately greater than the *calc_draft_percentage* and the *vessel_speed*, which are the closest values to the current *speed over ground* (SOG). These two closest values are called *speed_below_SOG* and *speed_above_SOG*.
 - (iv) We have to compare which of the two speed entries is closer to the current SOG. Such values are used for further interpolation.
 - (v) In the next step for the selected *vessel_speed*, the meteorological conditions are extracted to the most similar amplitude and relative angle.
 - (vi) In the last step, the index *gradients_id* is extracted from the data calculated from the previous step. This *gradient_id* is then used as an index to access the gradient value in the respective table of gradients; it then enables the interpolation explained in Section 4.1.

7. HARDWARE

This section describes the hardware developed for this project. The hardware consists of the following main parts:

1. Processor LX2160A with DDR4 RAM and eMMC
2. Two NVME flash interfaces
3. NVIDIA platform AGX Xavier on Xavier socket
4. NVIDIA Graphic Card on PCIe (*Peripheral Component Interconnect Express*) connector
5. Communication interfaces
6. Sensor inputs
7. Process outputs
8. Communication with the outside world
9. Power supply

7.1. Processor LX2160A

The following criteria are used for processor choice:

- A large number of SerDes/PCIe lines for very high-speed connection to the Ethernet, database disk devices and NVIDIA platforms;
- Wide and fast RAM memory interface;
- Moderate power requirements.

The NXP LX2160A processor was selected based on the above criteria. The LX2160A processor has sixteen 32-bit/64-bit Arm®Cortex®-v8 A72 CPUs, running on 2.2GHz. This processor has 24 SerDes / PCIe 3.0 lines, and it works at up to 25GHz. The processor power consumption with $\approx 27W$ is moderate.

7.2. DDR4 RAM

64Gbyte 128 bits (16 Byte) wide memory DDR4 interface with 3.2 GTransfer/s x 16 Byte = 51.2 Gbyte/s.

This amount of RAM memory allows the Postgres database software to keep some table data in RAM after reading from the mass storage. In combination with a very high RAM transfer rate, this enables faster searching in database tables.

7.3. eMMC Flash

An embedded multi media card (eMMC) is used as a boot device and primary Linux file system device. It is soldered onto the board and connected to the processor via an 8-bit eMMC interface. eMMC and *solid-state drive (SSD)* storage are similar options, as both use NAND flash memory, but SSD uses more memory gates (more expensive) than eMMC, meaning only similarity of transfer speed.

7.4. NVME Flash

Two M.2 Key M slots with 4 SerDes/PCIe 3.0 lines each are provided for NVME flash devices. NVME cards for M.2 slots with four PCIe 3.0 lines are capable of transferring data up to 1Tbyte and randomly reading and writing data up to 1,000,000 x (4kbyte blocks)/s each. Two storage devices in two M.2 slots are used for access to two devices that enable very fast search, even in huge databases. The speed of access is particularly important when searching the electronic chart for possible obstacles in the calculated avoidance route. And it is even more important for searches in the database with ship capabilities to check whether the ship is capable to make a given maneuver at a given displacement and current meteorological conditions in calculated points of the avoidance route.

Non-volatile memory express (NVME) is a protocol designed for fast SSDs that use the PCIe bus to communicate with the motherboard/processor. SSDs that use the PCIe bus are much faster than those that use the traditional Serial AT Attachment bus (SATA), another type of SSD memory, and it uses a direct approach to the storage device to enable high-speed communication. NVME drives are limited only by the speeds that SSD drives can achieve and the speed of their interface (each PCIe 3.0 lane provides 985 megabytes per second of bandwidth, and an M.2 slot can provide up to four lanes).

7.5. NVIDIA platform AGX Xavier

XGA Xavier is given on board to calculate some algorithms parallel to the main processor. The eight-lane PCIe 3.0 interface is used for communication between the LX2160A processor and NVIDIA platform, with a full duplex transfer capacity of 8 Gbyte/s. This platform can be used, for example, to run prediction algorithms from the video data.

7.6. NVIDIA graphic card on PCIe connector

An NVIDIA graphic card, with its many cores, accelerates the execution of search algorithms in the Postgres database. It also accelerates calculations on the Spheroid.

The four-lane PCIe 3.0 interface is used for communication between the LX2160A processor and NVIDIA graphic card on the PCIe connector, with a full duplex transfer capacity of 4 Gbyte/s.

Thanks to the many cores of a graphics card, the risk of collision with all ships in the vicinity of a particular geographic area can be checked for each ship via all prediction points of the ship routes on a course exclusively assigned to it for this task.

PCI Express (PCIe) is a serial connection that works more on the principle of a network than a bus, providing better throughput [20]. Instead of a single bus handling data from many different sources, PCIe is organized as a switch with multiple point-to-point serial lanes. These links lead directly to the devices for which the data is intended to connect different sources.

Each PCIe channel consists of two pairs of wires: one pair for sending and the other for receiving. Data packets move in the channel at the rate of one bit per circuit. The x1 connection, which is the smallest PCIe connection, has one channel consisting of four wires. It transmits one bit per circuit in both directions. The x2 connection consists of eight wires and sends two bits at a time, the x4 connection sends four bits at a time, and so on. When the central device is turned on, PCIe reads what device is connected to the motherboard.

7.7. Interfaces to the outside world

vi. Serial RS232 Interfaces

RS232 interfaces are realized via USB/RS232 convertors. The *automatic identification system* (AIS) receivers and the *global positioning system* (GPS) receivers are connected via the RS232 port as external devices. On ships, we have traditionally serial data standards based on the *National Marine Electronics Association* (NMEA) dedicated to transferring data from source to distant location [21].

NMEA 0183 v1.0 has signal levels similar to the RS232 standard used for serial data transfer in computation.

vii. Microphone Interfaces

Inputs for two pairs of MEMS microphones are provided on the platform. One pair is installed on the bridge, and the other pair is installed outside. The pairs of high-quality microphones allow the detection and analysis of acoustic signals, including accurate directional detection of the signal source.

viii. Loudspeaker Outputs

Two audio outputs with 36 Wrms/4 Ohm for audio commands and alarms are integrated into the system.

ix. Ethernet Interfaces

Radars, cameras, and other sensors are connected via high-speed Ethernet ports. The following interfaces are present:

- Ethernet 1 10 Gbit/s interface connected to the processor via a 25 Gbit/s SerDes interfaces
- Ethernet 2 10 Gbit/s interface connected to the processor via a 25 Gbit/s SerDes interface
- Ethernet 3 10 Gbit/s interface connected to the processor via a 25 Gbit/s SerDes interface
- Ethernet 4 10 Gbit/s interface connected to the processor via a 25 Gbit/s SerDes interface
- Ethernet 5 1 Gbit/s interface connected to the processor via an RGMII interface

Many fast Ethernet ports enable the connection of several cameras in parallel to the most modern radar devices.

x. WLAN

An M.2 Key E slot is provided for the WLAN module. WLAN is used for connections to portable devices used by officers. WLAN is connected to the processor via a PCIe 3.0 (1 Gbyte/s) single lane. They usually use an M.2 “form factor” (a small card that plugs into an M.2 slot, supporting NVME, assuming the slot supports it). A standard SSD uses a SATA interface and is therefore limited in transfer speed because such a controller is necessary to access the disk. NVME is roughly six to eight times faster than a typical hard drive. For day-to-day tasks, not much of a difference can be noticed in load

times, start-up/shutdown, running games, or anything else. When working with very large files, such as video editing and huge amounts of data, one might notice a big difference in access speed.

xi. WAN

An M.2 Key B slot is provided for the WAN module. WAN can be used for database updates, for example, with *new_chart_data* or *ship_skills* data. Support for 2G to 5G modules is provided.

WAN is connected to the processor via a USB 3.0 SIM card connector provided on board.

7.8. Hardware Block Diagram

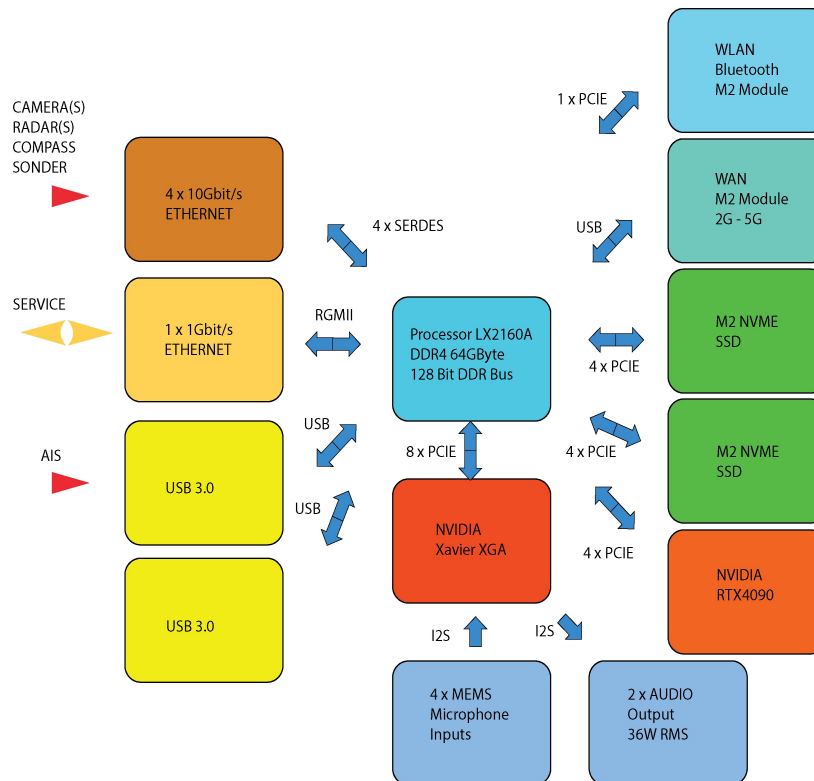


Fig. 2. CAS Hardware Block Diagram. Source: Authors

As we can see from the block diagram, the left side is a peripheral surrounding of the ship, mostly equipped with mandatory devices (e.g., radar(s), compasses, sounders) that are necessary for maneuvering and navigation, but it is regulated by the *International Maritime Organization* (IMO). The *automatic identification system* (AIS) has a special role in collecting data about ships in the vicinity. The *global positioning system* (GPS) updates the accurate position of the ship, and we can find it mostly on all ships. Modern ships can be equipped with advanced positioning systems such as a *differential GPS* (DGPS) or an *inertial navigation system* (INS), which can be necessary for special purposes and for particular regions of navigation (traffic channels, etc.). From these devices, we do not expect a big data stream, and luckily, we are not moving at a high speed like an airplane.

On the right side of the block diagram, we have communication units that enable various connections with the coast, from which we have to collect all necessary data. In the deep sea, we use satellite-based networks (Inmarsat, Iridium etc.), but in coastal areas, we use public mobile networks, too. We have to collect the necessary data in time to be prepared for search when a potential collision situation appears.

7.9. Layout

Fig. 3 shows the signal routing of the mainboard (without power planes). High-speed signals (PCIe, USB) are routed using smooth corners instead of 45° corners to improve signal integrity. The top layer is shown in green, and the bottom is in yellow.

The width of the lines and the distance between the lines, together with the distance to the reference plane and the relative dielectric constant, determine the differential impedance of the differential pairs. These parameters are selected according to the layer structure of the PCB so that the appropriate impedance is achieved on each layer for each signal type (e.g., 85 Ohm for PCIe, 90 Ohm for USB).

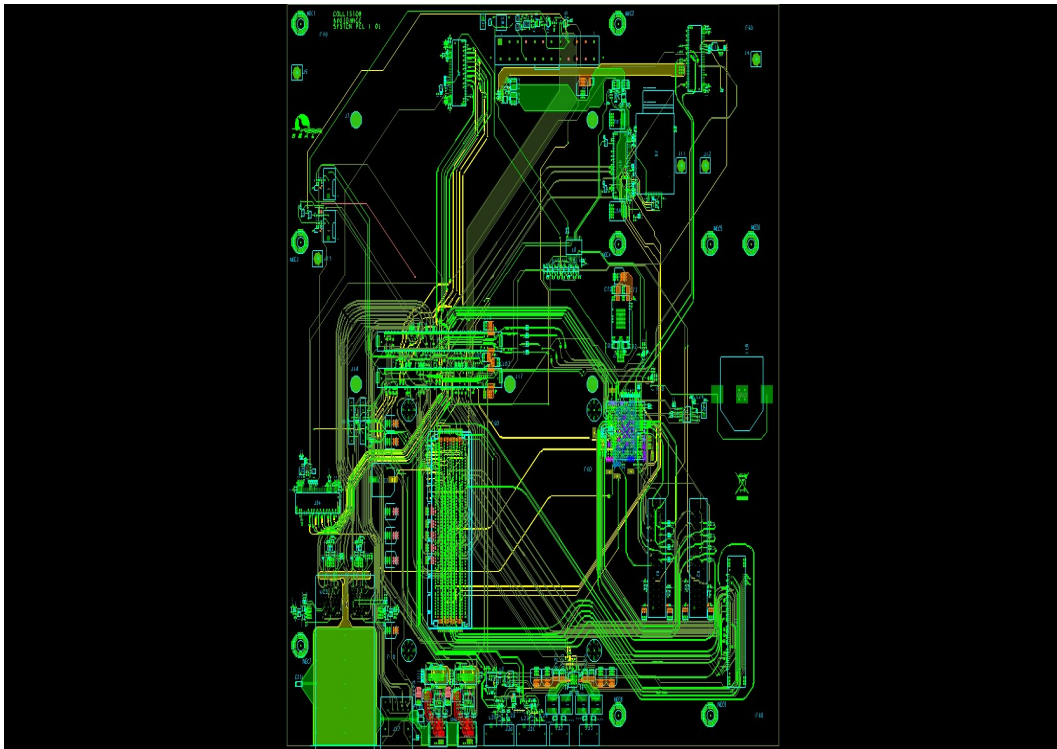


Fig. 3. Mainboard Layout. Source: Authors

7.10. 3D CAD View

Figs. 4 and 5 show the top and bottom sides of the mainboard 3D model as exported from the CAD system. The CPU and NVIDIA Xavier modules, which are connected to the pink connectors on the top side, are not shown here.

7.11. Board Picture

Fig. 6 shows the assembled mainboard with populated CPU, LTE and WiFi module sockets.

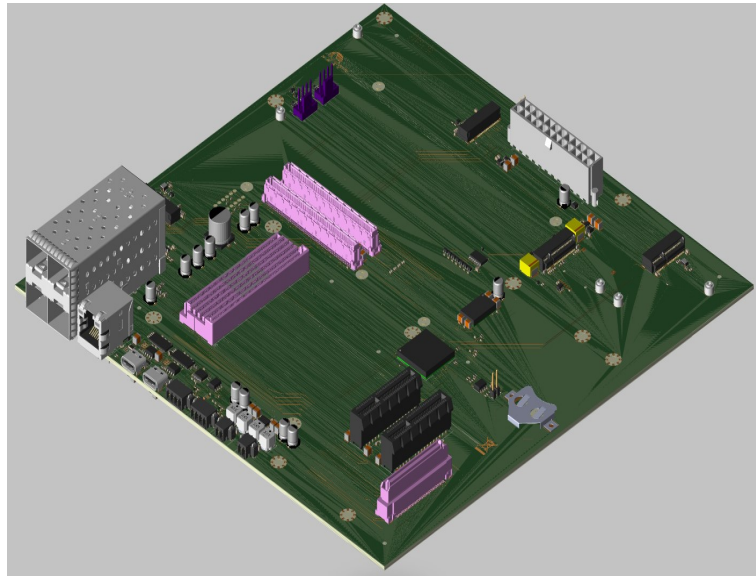


Fig. 4. Mainboard 3D CAD top. Source: Authors

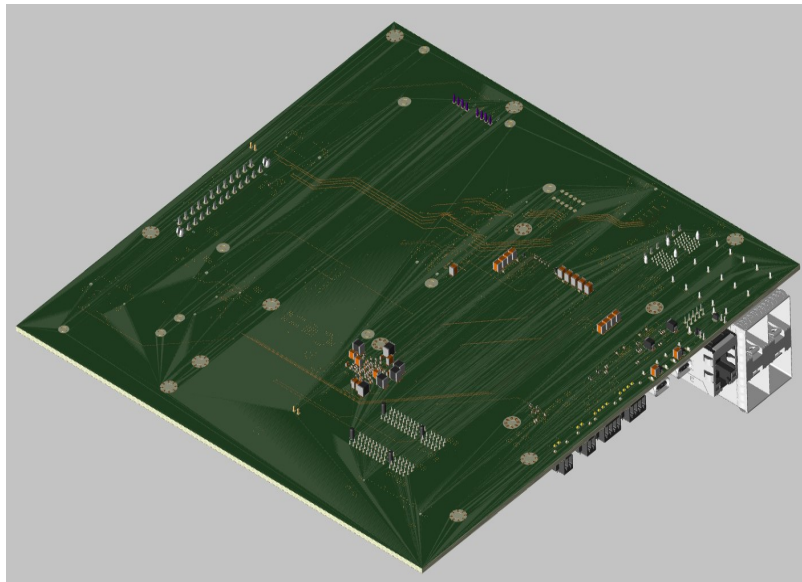


Fig. 5. Mainboard 3D CAD bottom. Source: Authors

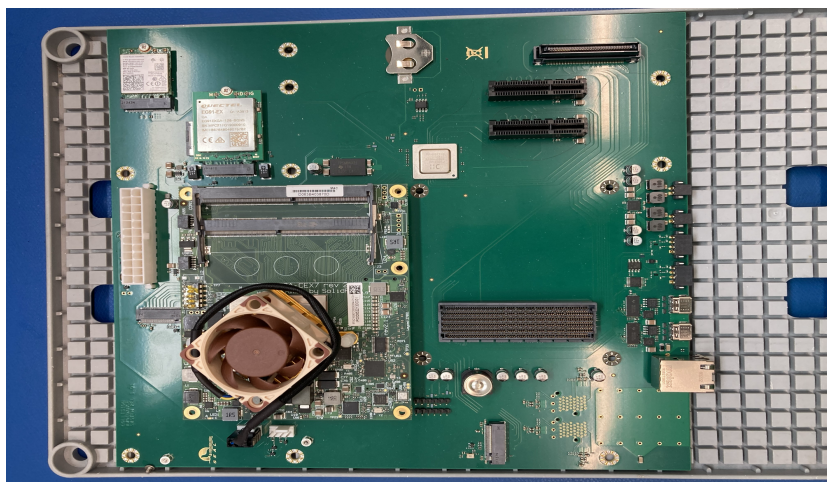


Fig. 6. Mainboard Picture. Source: Authors

8. CONCLUSIONS

In the first part of the report, we explained the complexity of the data models used in the CAS, as shown in Fig. 1. Such a system can produce maneuvering commands to the commander of the ship, or it can be connected to the rudders and thrusters.

This proposed system consists of three basic blocks: the collision prediction block, the rule decision block, and the collision avoidance block. The first one gets data from various sensors. Some of them are cameras that detect fog, differentiate night from day, and measure speed, route direction, and sea depth. Also, we need information from electronic charts, AIS, meteorological information, and radar data. The second block is explained in [3]. The third one calculates the optimal route using various algorithms. It is based on input data from outside the ship, but important data comes from electronic charts, the same as from bathymetric devices. This calculation is performed every 10 s for all vessels in the area of interest. At these times, the future positions are calculated for the time interval (t_{now} , $t_{now}+1800$ s], again in 10-s increments.

Variations in the way of assessing a ship's abilities under certain meteorological conditions are explained separately.

Different problems explained in Section 2 are solved with embedded hardware. Postgres database is configured in such a way that it can run in real-time. Specifically, we ensured the possibility that different computational tasks related to each data model (explained in Sections 3–6) can be distributed (by the operating system) to different cores of processors. In that way, we ensure the maximal usage of the hardware platform. Due to the real-time processing of a large amount of data, special emphasis is placed on fast access to SSD memory, for which the new NVMe serial protocols are used via PCIe lanes.

From the block diagram, it is obvious that communication to and from the surroundings (on the board of the ship and with the coast) is very important. Huge amounts of data should be transferred continuously, and many simultaneous data flows have to be ensured. Such a solution represented in this paper shows very good behavior in different circumstances and opens good perspectives toward autonomous navigation on the sea.

References

1. Huang, Y. & Chen, L. & Chen, P. & Negenborni, R.R. & van Gelder, P.H.A.J.M. Ship collision avoidance methods: State-of-the-art. *Safety Science*. 2020. Vol. 121. P. 451-473.
2. Bošnjak, R. & Kezić, D. & Krile, S. Modeling of VTS supervisor by algorithm based on Petri Net: case study of dover incident. *Transport*. 2021. Vol. 36. No. 4. P. 305-316. DOI: 10.3846/transport.2021.15824.
3. Brozović, V. & Kezić, D. & Bošnjak, R. & Krile, S. International regulations implementation for preventing collisions at sea by using coloured Petri Nets. *J. Mar. Sci. Eng.* 2023. Vol. 11. No. 7 (1322). DOI: 10.3390/jmse11071322.
4. Sun, T. & Liu, C. & Xu, S. & Hu, Q. & Li, C. COLREGS-complied automatic collision avoidance for the encounter situations of multiple vessels. *J. Mar. Sci. Eng.* 2022. Vol. 10. No.1688.
5. Heiberg, A. & Larsen, T.N. & Meyer, E. & Rasheed, A. & San, O. & Varagnolo, D. Risk-based implementation of COLREGs for autonomous surface vehicles using deep reinforcement learning. *Neural Networks*. 2022. Vol. 152. P. 17-33.
6. Lyu, H. & Yin, Y. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field. In: *Proceedings of the 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*. Shanghai, China, 29-31 December 2017.
7. Piera, M.A. & Radanovic, M. & Leal, X. Multi-agent systems for air traffic conflicts resolution by using a causal analysis of spatio-temporal interdependencies. *Proceedings of the SCSC*. 2016. Montreal, QC, Canada, 24-27 July 2016. P. 1-8. ISBN: 978-1-5108-2424-9.
8. T.P.G.D. Group. *Postgresql 14.1 documentation*. 2021. Available at: <https://www.postgresql.org/files/documentation/pdf/14/postgresql-14-A4.pdf>.

9. T.O.S.G. Foundation. *Postgis 3.3.0dev manual*. 2022. Available at <https://postgis.net/docs/manual-dev/>.
10. I.H. Organization. *S-100 specification numbers*. 2020. Available at: <http://s100.iho.int/S100/home/s-100-specification-numbers>.
11. *Iho electronic navigational chart product specification*. 1.0.0, IHO Publication S-101, 4b quai Antoine 1er, Principauté de Monaco. International Hydrographic Organization. 2018.
12. Bošnjak, R. & Kezić, D. & Brozović, V. Overview of S-100 group of standards for use in nautical navigation. In: *Proceedings of ICTS 2020*. Portorož, Slovenia. Faculty of Maritime Studies and Transport. Sep. 2020. P. 54-60. ISBN: 978-961-7041-08-8.
13. Brozović, V. & Bošnjak, R. & Kezić, D. & Bojić, F. Storing S-101 chart data into a common database. In: *Proceedings of IMSC 2021*. Varna, Bulgaria. NVNA. 9.-10. September 2021.
14. Brozović, V. & Bošnjak, R. & Kezić, D. & Brozović, F. Storage of the ship maneuvering capabilities' into the Postgres Database. *ICTS 2022 Conference Proceedings*. Portorož, Slovenia. Faculty of Maritime Studies and Transport. 23.-24. May 2022. P. 69-75. ISBN: 978-961-7041-11-8.
15. Shepard, D. A Two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*. Association for Computing Machinery. New York, NY, United States. 1968. P. 517-523. ISBN: 978-1-4503-7486-6.
16. Mitas, L. & Mitasova, H. Spatial interpolation. In: Longley, P.A. & Goodchild, M.F. & Maguire, D.J. & Rhind, D.W. (Eds). *Geographical Information Systems (GIS): Principles, Techniques, Management and Applications*. John Wiley. New York, NY, United States. 1999. P. 481-492.
17. Smolka, G. *Programmierung – Eine Einführung in die Informatik Mit Standard ML*. Oldenbourg Wissenschaftsverlag GmbH: Munich, Germany. 2008. ISBN: 978-3-486-58601-5.
18. Hansen, M.R. & Rischel, H. *Introduction to Programmierung Using SML*. Pearson Education Limited: London, UK. 1999. ISBN: 0-201-39820-6.
19. Ajanovic, J. PCI express 3.0 overview, 2009 IEEE Hot Chips 21 Symposium (HCS), Stanford, CA, USA. 2009. P. 1-61. DOI: 10.1109/HOTCHIPS.2009.7478337.
20. Lazarowska, A. Research on Algorithms for Autonomous Navigation of Ships. *WMU Journal of Maritime Affairs*. 2019. Vol. 18(23). DOI:10.1007/s13437-019-00172-0.

Received 29.10.2022; accepted in revised form 04.03.2024