The Collaborative Autonomous Shipping Experiment (CASE): Motivations, Theory, Infrastructure, and Experimental Challenges

Ali Haseltalab,[a*], Vittorio Garofano[a], Muhammad Raheel Afzal[c], Nicoló Faggioni[b], Shijie Li[d], Jialun Liu[e,f], Feng Ma[e,f], Michele Martelli[b], Yogang Singh[c], Peter Slaets[c], Xu You[e,f], Rudy R. Negenborn[a]

[a]*Researchlab Autonomous Shipping, Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands.*

[b]*Section of Naval Architecture and Marine Engineering, Department of Naval Architecture, Electric, Electronic and Telecommunication Engineering, University of Genova, Genova, Italy.*

[c]*Intelligent Mobile Platform (IMP) Research Group, Department of Mechanical Engineering, KU Leuven, Leuven, Belgium.*

[d]*School of Logistics Engineering, Wuhan University of Technology, Wuhan, 430063, P. R. China.*

[e]*Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan, 430063, P. R. China.*

[f]*National Engineering Research Center for Water Transport Safety, Wuhan, 430063, P. R. China.*

*Corresponding author. Email: a.haseltalab@tudelft.nl

Synopsis

The future autonomous ships will be operating in an environment where different autonomous and non-autonomous vessels with different characteristics exist. These vessels are owned by different parties and each uses its owned unique approaches for guidance and navigation. The Collaborative Autonomous Shipping Experiment (CASE) aims at emulating such an environment and also stimulating the move of automatic ship control algorithms towards practice by bringing together different institutes researching on autonomous vessels under an umbrella to experiment with collective sailing in inland waterways. In this paper, the experiments of CASE 2020 are explained, the characteristics of different participating vessels are discussed and some of the control and perception algorithms that are planned to be used at CASE 2020 are presented. CASE 2020 will be held in parallel to iSCSS 2020 at Delft University of Technology, the Netherlands.

Keywords: Autonomous Shipping, Trajectory Tracking, Obstacle Avoidance, Path Planning, Collaborative Shipping, Model Vessels.

1 Introduction

Autonomous shipping has been extensively investigated by the academic and industrial communities due to several advantages that it introduces. Autonomous vessels are going to operate in environments where there are manned and unmanned vessels belonging to different parties and/or companies. The manned vessels are navigated by captains having different maneuvering behaviors and autonomous vessels are going to function with different control and perception algorithms, but all have to respect the national and international maritime laws.

This difference in maneuvering behavior and control software makes the problem of safe autonomous shipping challenging as it must be guaranteed that these different systems and operators should coordinate with each other to avoid collision and any other accidents. Moreover, the vessels should be capable of communicating and collaborating with each other to carry out some joint tasks, e.g., platooning, pulling, and pushing other vessels.

Collaborative Autonomous Shipping Experiment (CASE) aims at emulating the inland shipping environment of future by gathering different institutes researching on autonomous vessels under an umbrella to experiment collective sailing in inland waterways. In this challenge, the institutes who poses a model autonomous vessel will bring their vessels to participate in this challenge and experiment their control and perception approaches in fulfilling various tasks in the presence of other vessels. The vessels from different parties should coordinate with each other to avoid collision while moving autonomously. The vessels communicate using an open source software which shares the position of all the vessels in the sailing environment with all the parties. Then, each individual vessel should decide about its control actions based on the info it receives and it collects.

The first CASE will take place in parallel and as a part of the International Ship Control Systems Symposium (iSCSS) 2020. In this year's experiment, vessels from Delft University of Technology (TU Delft), University of Genova (UNIGE), KU Leuven, and Wuhan University of Technology (WUT) will participate. In this paper, after explaining CASE and its objectives, the aim is to describe the hardware and software of these vessels. For each vessel, first, the hardware of the model vessel is described. Then, a description of the five different tests planned and the indoor infrastructure used are presented, including the shared communication layer. Approaches used for path and trajectory planning, trajectory tracking, and obstacle avoidance are explained next. The paper ends with simulation experiments for some of the vessels which will be revalidated during CASE.

The remainder of this paper is as follows. In section 2, this year's CASE experiments are presented and discussed. In Section 3, the vessels descriptions are given accompanied by some simulation cases for each vessel. The concluding remarks are given in Section 4.

https://doi.org/10.24868/issn.2631-8741.2020.014

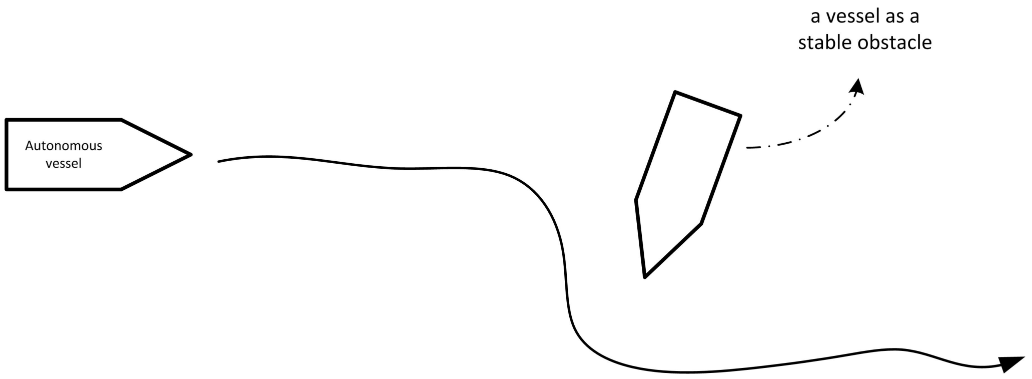


Figure 1: A schematic view of Challenge I.

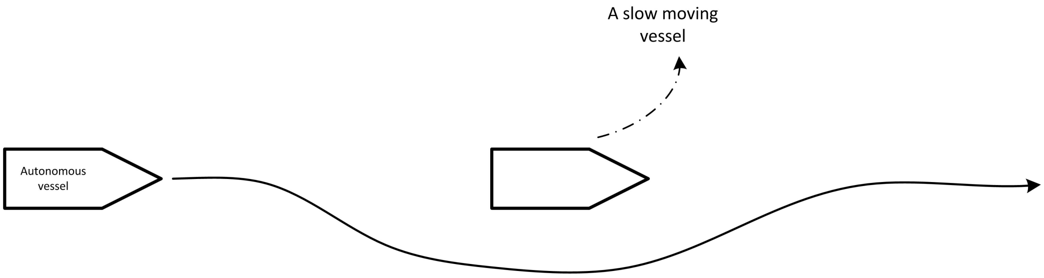


Figure 2: A schematic view of Challenge II.

2 CASE Experiments

For this year's CASE, several challenges have been considered. The theme of this year is collision avoidance between vessels. The considered challenges range from avoiding collision with non-moving vessels to platooning.

2.1 Challenge I: Avoiding Collision with a Non-Moving Vessel

In this challenge, an autonomous vessel should update its trajectory to avoid collision with a non-moving vessel on its way. The position of the non-moving vessel is given through the communication system and using this data, the autonomous vessel should update its trajectory. The schematic view of Challenge I is shown in Figure 1.

2.2 Challenge II: Bypassing a Slow Moving Vessel

In this challenge, a moving vessel is sailing in front of an autonomous vessel and the autonomous vessel should bypass it by updating its trajectory. After passing the slow-moving vessel, the autonomous vessel should get back to its original trajectory. The position of the slow-moving vessel is given through the communication system throughout the operation. The schematic view of Challenge II is shown in Figure 2.

2.3 Challenge III: Two Autonomous Vessels Bypassing Each Other

In this challenge, two autonomous vessels from different parties that are moving in opposite directions should bypass each other without colliding. The positions of the vessels are communicated through the communication system and the vessels should use this information to avoid a potential accident. The schematic view of Challenge III is shown in Figure 3.

2.4 Challenge IV: Heads-on Autonomous Vessels Bypassing Each Other

In this challenge, two autonomous vessels from different parties that are moving in opposite directions should bypass each other without colliding. The difference between this challenge and Challenge III is that the autonomous vessels will be having heads-on initial position, i.e., their are in each other's way. The positions of the vessels are communicated through the communication system and the vessels should use this information to avoid a potential accident. The schematic view of Challenge IV is shown in Figure 4.

https://doi.org/10.24868/issn.2631-8741.2020.014

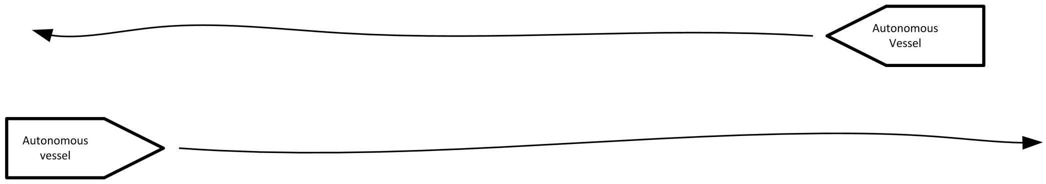


Figure 3: A schematic view of Challenge III.

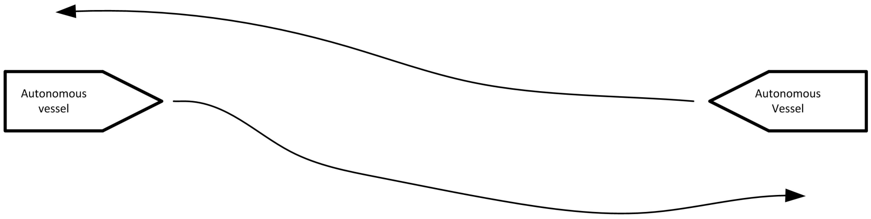


Figure 4: A schematic view of Challenge IV.

2.5 Challenge V: Forming a Vessel Train

In this challenge, multiple autonomous vessels from different parties should follow each other without colliding. Each vessel will receive the position of the vessel that is moving in front of it through the communication system. In this challenge, the objective is to create a multi-body vessel platoon. The schematic view of Challenge V is shown in Figure 5.

3 Participating vessels description

In this section, a description about the participating vessels, and their control and perception algorithms are given.

3.1 Hendrik: System Description

Hendrik is an operational 1/25 scaled inland waterway model of the Watertruck+ self-propelling barge of CEMT- I class. The operational scale model discussed in this paper includes a hull with length 1.54 m, width of 0.2 m and height of 0.2 m with a mass of 32 kg. The comparison between true size and scale vessel being provided in the Table 1 [15]. The height (air draft) is not scaled appropriately, i.e. 18 cm instead of 14.8 cm to compensate for the unscaled environmental factors and water splashing. A 3D CAD model along with the developed model of the vessel is shown in Figure 6. The drive system of the full scale model relies on waterjets for manoeuvring and propulsion. *Hendrik* has a block coefficient, $C_B = 0.95$ with a maximum cruising speed of 0.319 m/s.

Similar to a full scale CEMT- I model, *Hendrik* has a water jet based propulsion system that relies on waterjets for maneuvering and propulsion. The CEMT-I class consists of a propulsion system where water is sucked from below the barge and subsequently pushed outwards in one of four directions, depending on the rotation of the system. The propulsion system in the *Hendrik* comprises of two pumps with four directional movement namely, backward, forward, left and right direction controlled through water jets as shown in Figure 7. The two centrifuge pumps are driven by a brushless DC motor by means of analog input voltage between 0 and 5V. This concept is a novel pump design for propulsion system in the inland vessels and *Hendrik* is one of the first scaled models which have adopted this mechanism. *Hendrik* has the following on-board sensors and equipments:

- *Battery type and charging technology*: Lithium Ion battery of 24 V (charge unit on 220V)
- *Communication technology with the shore side*: Wifi (access point on Raspberry Pi)

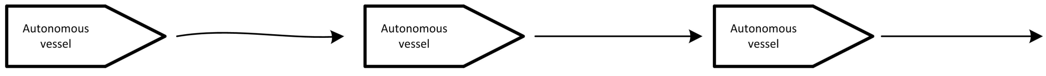


Figure 5: A schematic view of Challenge V.

https://doi.org/10.24868/issn.2631-8741.2020.014

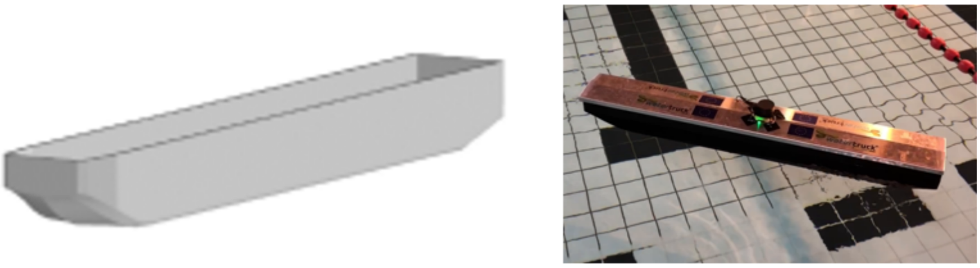


Figure 6: 3D CAD and developed model of *Hendrik*.

- *On-board perception sensors*: LiDAR

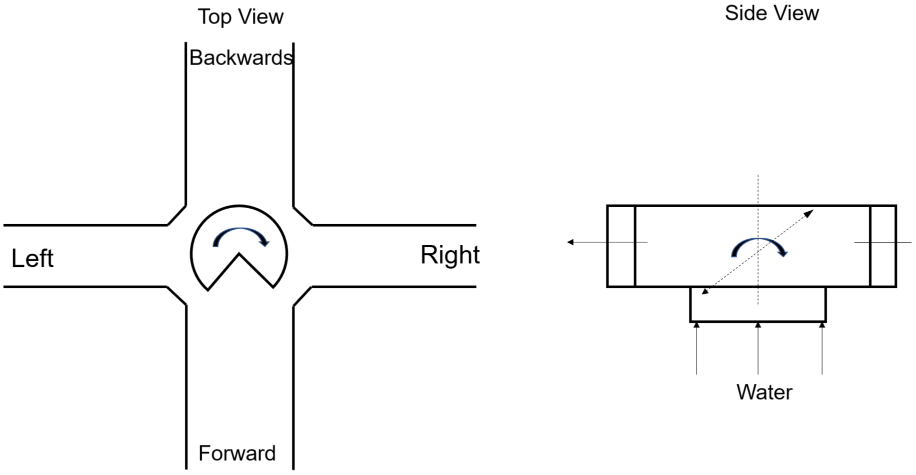


Figure 7: Schematic of propulsion system.

3.2 Hendrik: Collision Avoidance

Hendrik collision avoidance framework focuses on the cooperative framework for coordinated path-following where collision avoidance ensures the safety of the system. The first case of *Coordinated Path-Following* framework is where autonomous vessels follows a predefined reference path maintaining a fixed defined configuration i.e. line, diamond, column to reach the desired destination.

The second case is to coordinate the configuration among vessels where the concept of *Wingman Problem* arises [4]. In this study, the idea at the basis of the coordination task relies on a further use of the virtual target approach, which is at the basis of the single-vehicle path-following addressed in [5] and used in this work as a base for the guidance system development of each vehicle. In addition to this, the coordination task is to compute relative positions between vehicles, thus properly regulating the relative speeds of advance in order to achieve the coordinated motion goal. Moreover, the virtual target based approach allows to completely decouple the path-following task from the coordination one, thus maintaining unaltered the path-following algorithm and simply integrating a speed adaptation control law. As an additional modification to the existing approach, repulsive potential function is defined for external obstacles for vessels to avoid the collision.

Table 1: Parameter comparison between true size and scale vessel

	Length (L) (m)	**Width (B) (m)**	**Air Draft (m)**	**Draft (T) (m)**	**Mass (m) (kg)**
Scale Model	1.54	0.2	0.18	0.11	31.6
CEMT Class I	38.5	5.05	3.7	2.8	407

https://doi.org/10.24868/issn.2631-8741.2020.014

3.3 Hendrik: Simulation Results

3.3.1 Challenge I: Avoiding Collision with a Non-Moving Vessel

In this challenge, an autonomous vessel should update its trajectory to avoid collision with a non-moving vessel on its way. Figure 8 shows the trajectory of the vessel avoiding collision with a non-moving vessel marked as red square on the open space. The starting point for the vessel is at the bottom right of the open space i.e. (x, y) = (230 m, -220 m) and the goal point for the vessel is (x, y) = (300 m, -320 m). Figure9 shows the speed and heading profile of the moving vessel.

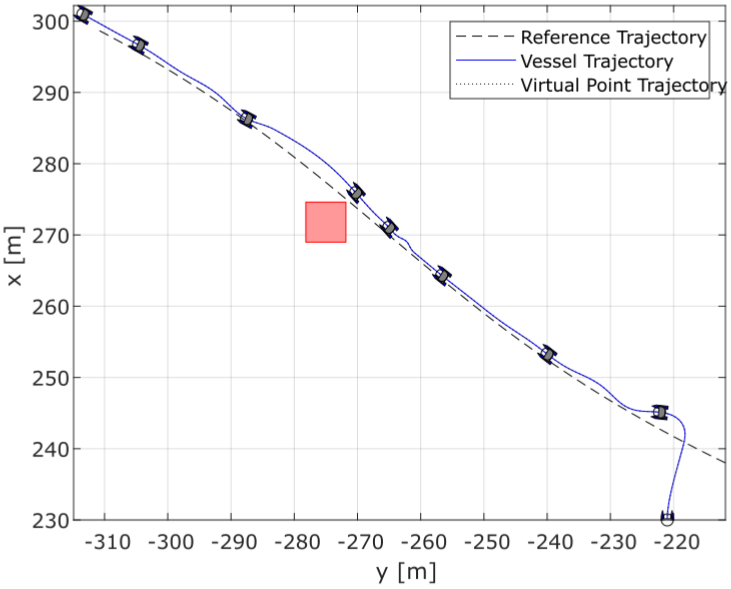


Figure 8: Trajectory of the vessel following virtual target

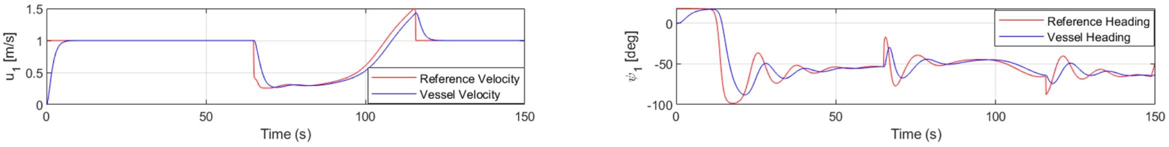


Figure 9: Speed and heading profile of the vessel.

3.3.2 Challenge II: Bypassing a Slow Moving Vessel

In this challenge, a moving vessel is sailing in front of an autonomous vessel and the autonomous vessel should bypass it by updating its trajectory. After passing the slow-moving vessel, the autonomous vessel should get back to its original trajectory. Figure 10 shows the trajectory of the follower and obstacle vessel. Figure 11 shows the velocity profile of the follower and the obstacle vessel. Figure 11 shows that follower vessel has a decreasing velocity profile while the obstacle vessel has a increasing velocity profile. This is done in order to make sure that both vessels can reach the goal point at the same time. Figure 12 shows the speed and heading profile of the follower and obstacle vessel.

3.3.3 Challenge V: Forming a Vessel Train

In this challenge, the objective is to create a multi-body vessel platoon. Multiple autonomous vessels from different parties should follow each other without colliding. Figure 13 shows the trajectory of the three vessels creating a multi-vessel platoon. Figure 14 shows the speed and heading profile of the three vessels.

3.4 UNIGE's autonomous vessel

The UNIGE testing model is a representation of a tugboat with 1:33 scale; the original design is provided by TU Delft. The generous hull shapes allow the researcher to work efficiently inside of it, facilitating the preparation of the hardware, as there is a large volume that allows an easy manage to set up. Besides, the hull has excellent

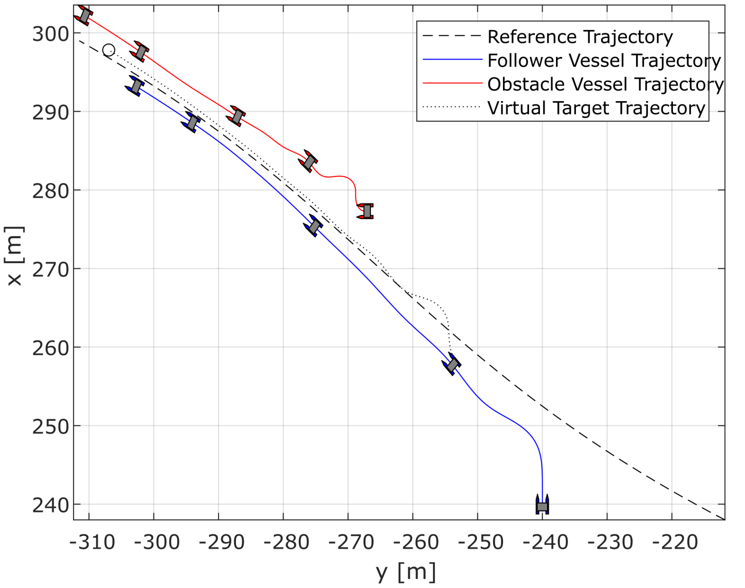


Figure 10: Trajectory of the follower and obstacle vessel following a virtual target

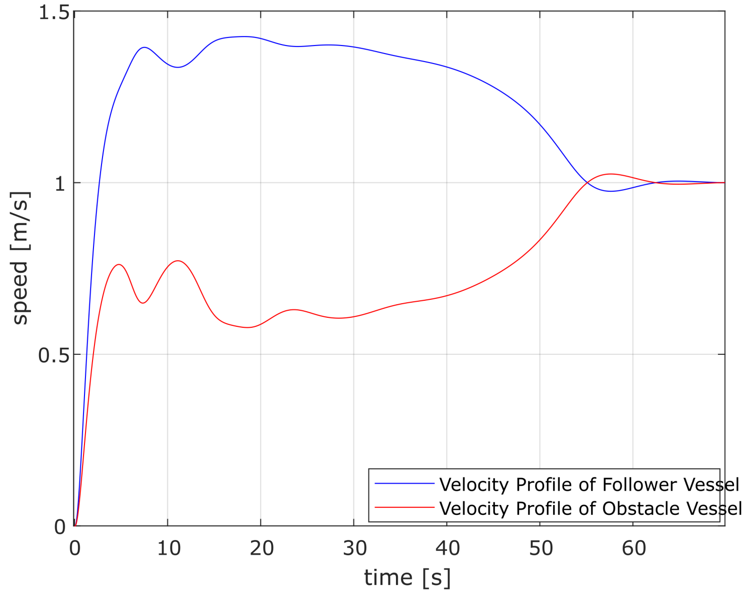


Figure 11: Trajectory of the follower and obstacle vessel following a virtual target

seakeeping behaviour in order to successfully carry out without any waterboarding in future tests in an outdoor environment. This characteristic is fundamental because all hardware components are contained inside of the hull, and the tug's deck has not the watertight feature. The tugboat model called ”Tito Neri” has a length (Loa) equal to 0.97 meters and a maximum width of 0.30 meters.

The tugboat model propulsion plant is composed of:

- Two azimuth thrusters: one for each shaft-line of the ship, they are a configuration of marine propellers placed in pods that can be rotated to any horizontal angle (azimuth), making a rudder unnecessary. The type of azimuth thruster present on the model has a duct. It is used to improve the efficiency of the propeller in heave load conditions near the condition of zero speed.
- Two electric motors: they are coupled with the azimuth propellers, by a Z-drive gearbox; DC motors can be controlled by a dedicate DC drive

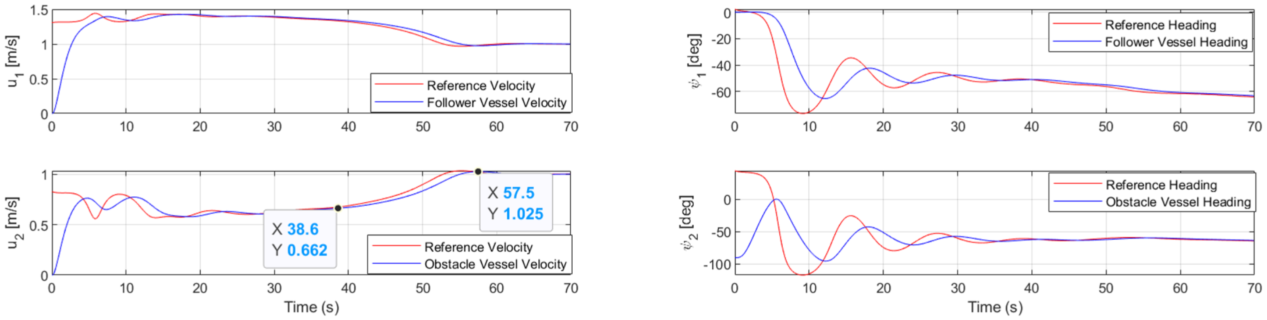


Figure 12: Speed and heading profile of the follower and obstacle vessel

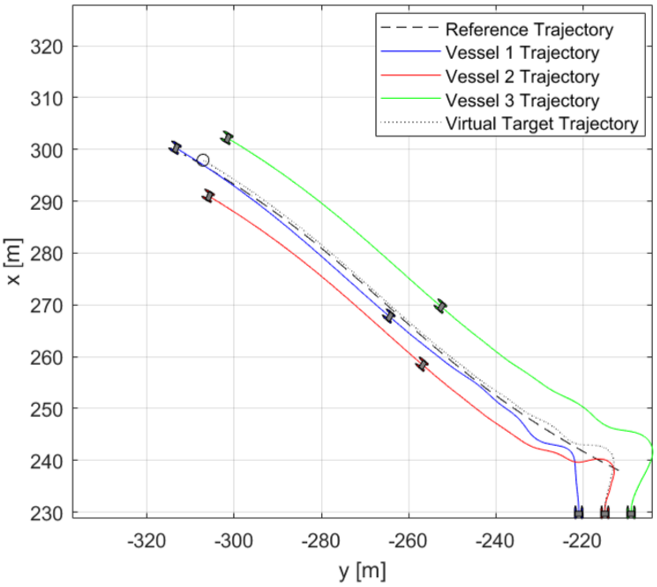


Figure 13: Trajectory of the three vessels following a virtual target

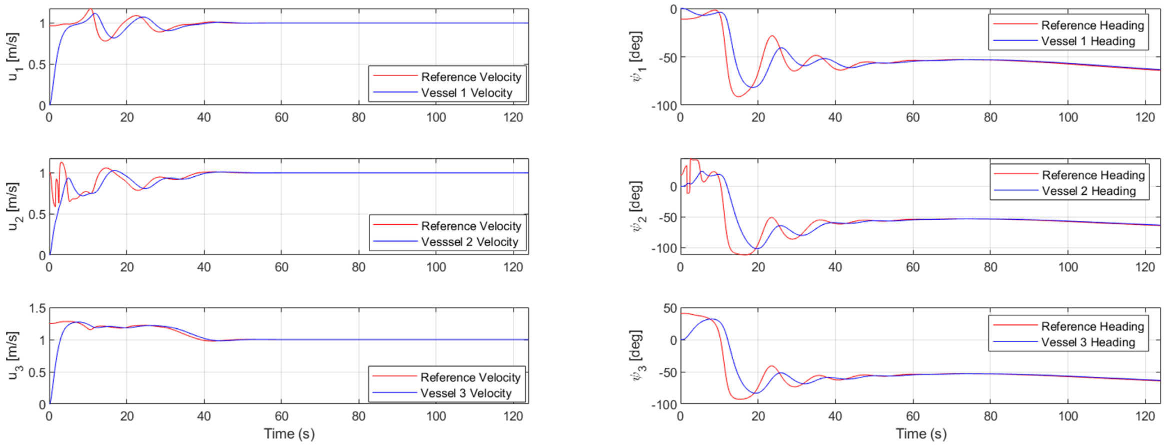


Figure 14: Speed and heading profile of the three vessels following a virtual target

- Bow thruster: it has a separate electric motor, controlled by its relative DC drive. It aims to improve the manoeuvrability of the tug, by means generating a controllable thrust in the bow part of the hull.

The tugboat model propulsion gives the capability to the tug model to be fully-actuated in the horizontal plane, In addition, the boat is not only equipped with actuators, but also with sensors:

- Two rpm encoder: on the shaft line; they are mounted between the DC motor and azimuthal component. The information of the revolutions of the main propellers is given by these types of equipment, which allow for

https://doi.org/10.24868/issn.2631-8741.2020.014

closed-loop control, through the action of different input parameters.

- • Three current sensors: installed on downstream of DC drive. This layout will be used for feedback logic application; indeed, it enables knowing the current absorbed by DC motor. Moreover, by means of the information of current and voltage is possible to evaluate the DC motor delivered power.
- Four ultrasonic sensors: Through an ultrasonic pulse, it is possible to measure the "small" distances in an acceptably accurate manner. An ultrasonic transducer emits some pulses of a signal with a frequency of 40 kHz. The reflected signal is captured by the second transducer and amplified. By means of measuring the time between sending and receiving pulses, the distance from the transducers to the object can be evaluated, useful for auto berthing and collision avoidance.
- One humidity and temperature sensor: To evaluate this distance, the sound speed in air is used. To obtain a better measure of the distance, the humidity and temperature sensor is used to locally evaluate the sound speed in the air as it is very influenced by air temperature and humidity.
- IMU + GPS: The MTi-G-710 is a GNSS / INS system capable of providing information relating to position and speed (integrated GPS and GLONASS receiver) and movements in space (Roll, Pitch, and Yaw). It offers a high value of protection from the surrounding environment as the system is IP67. Through the proprietary software, orientation data are provided in real-time with high accuracy. The supplied libraries allow the users to easily integrate sensory data with custom software or with existing analysis applications such as MATLAB.
- The new superstructure was re-designed and 3D printed with 5 (infrared) LEDs. The particular shape of the superstructure, in particular the position of the LEDs and their relative distances, has been designed to increase the performance of the video-tracking system; for this purpose. The LEDs have been positioned on different planes and it is possible to detect the ship motion in 6 DOF.

The autonomous navigation and control algorithms implemented in the model scale tug are the following:

- Dynamic Positioning: The capability to maintain the commanded position with environmental disturbances acting on the hull [1, 7].
- Collision Avoidance: The capability to avoid the fixed and moving obstacles by a replanning of the initial route, the algorithm is COLREG compliant [6, 17, 13, 14, 18].
- Autopilot: The capability to maintain a fixed heading angle during the navigation in both low and high speed condition [12, 3].
- Speed control: The capability to maintain a constant speed also with environmental disturbances [8].
- Track Following: The ship is capable of following a series of waypoints, with a specific speed and heading profiles [2].
- Interception: The capability to follow and reach a moving target with different guidance laws.
- Follow The Leader: The capability of the ship to follow a moving object.

The general structure of Guidance, Navigation and Control (GNC) is reported hereinafter.

Analyzing the flow chart from the top, firstly there is the input function provided by the commander (one of the navigation mode listed before). From the navigation mode, the command that identifies the desired manoeuvre is generated. At each time, the collision avoidance system checks for possible obstacles. If no collision is possible, the waypoints are directly sent to the navigation system that translates the waypoints into heading requirements to ensure the vessel is able to regain course within a certain user-defined length, [10]. Otherwise, the system calculates an evasive manoeuvre that provides for the subsequent return into a course by generating new waypoints that are sent to the guidance system, also considering COLREG [11]. In the present paper, three different guidance methods are compared to check the target tracking scenario: pure pursuit, constant bearing and line of sight. Then the controller structure evaluates the errors, and the force and thrust allocation logics generate the high – level setpoint. On the bottom of the figure, the setpoint generation converts the course requirements signals into propulsors' rpm and angles and bow-thruster commands, in accordance with [4].

By using the previous control scheme, a simulation campaign is carried out in order to validate the proposed approach. The scenario includes a target that is moving following a zig-zag path, and the interceptor is guided to its final goal (such as a forming a vessels train) by using the three different guidance laws. The slideshow of the proposed manoeuvres with the introduction of the current is shown in Figure 16.

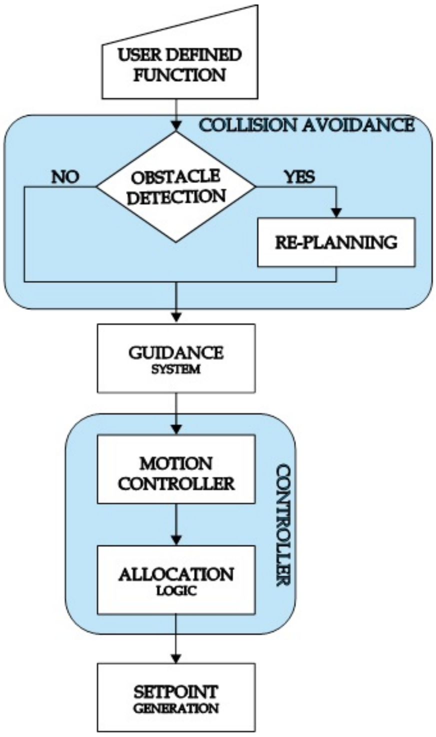


Figure 15: Control structure of the UNIGE's autonomous vessel.

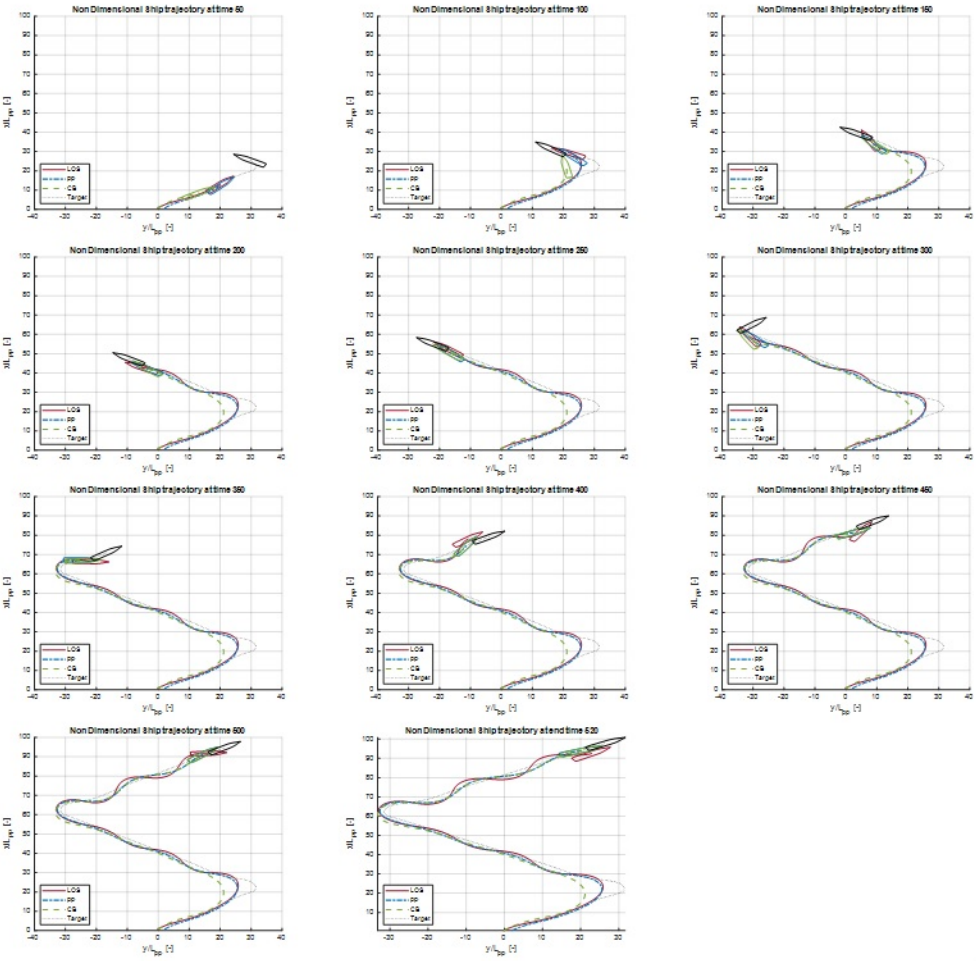


Figure 16: Slideshow of the vessel Trajectory following virtual target.

https://doi.org/10.24868/issn.2631-8741.2020.014

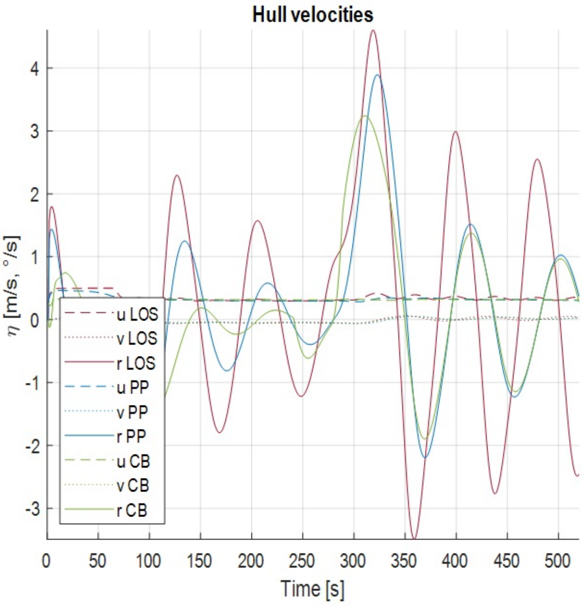


Figure 17: Hull velocity profiles.

3.5 *Pallas*

The model vessel from WUT, named Pallas shown in Figure 18, is a 1 m long model scaled from a 64 TEU inland container vessel with a length overall 71.4 m. The main particulars of the Pallas model and its full-scale reference ship are listed in Table 2. The research of this model vessel is aimed at developing autonomy on intelligent ships, specifically on control capability. For that purpose, this model vessel has been equipped with two necessary structures: the hardware structure and the software structure as shown in Figure 19. The hardware structure consists of several sensors and actuators that are used for real-time navigation and control experiments. The sensors consist of GPS unit, IMU unit, industrial Wi-Fi with 4G unit, mini-PC unit, PLC unit, MCU unit, DC power supply unit for information collection on environment situation, and the model vessel itself. Moreover, the actuators are propulsion systems and rudder systems that are controlled by electric motor speed orders and rudder angle orders respectively. The data from sensors and actuators, including feedback of propulsion and rudder system, are mainly collected by PLC or MCU. Subsequently, the data is transmitted to mini-PC through advanced algorithms related to GNC and NBS structure. After that, the control orders are conveyed to actuators through PLC and MCU units. The ground station usually is a PC, can communicate with this ship by Wi-Fi or 4G signal through the Internet and monitors the status of this vessel when testing. The software structure is mainly programmed by Python that is divided into several functions: environment perception, planning, and decision, control, and implementation. The graphic display of this software shows the vessel status, the designed path, and the implementation in real-time. Combing these functions, several modes are designed for different steps in autonomous ship implementation, including local control, remote control, autonomous control. And the tests and verification in different steps are considered to suit the different modes.

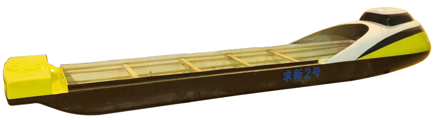
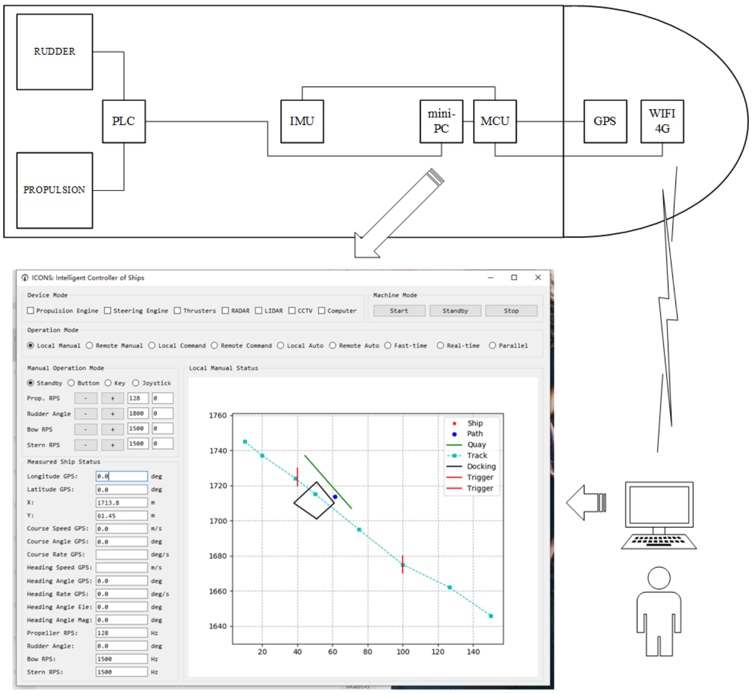


Figure 18: A 1 m Pallas model vessel.

Table 2: Main particulars of the Pallas model and its reference ship.

Main particulars	Pallas (Scaled)	64 TEU
Scale factor	1:69	1:1
Length overall (m)	1.035	71.4
Length between perpendiculars (m)	1.000	69.000
Draft (m)	0.038	2.6013
Displacement (m^3)	0.006	2040.3
Wetted surface (m^2)	0.243	1154.8026
Block coefficient $(-)$	0.874	0.874



● Local Manual ○ Remote Manual ○ Local Command ○ Remote Command ○ Local Auto ○ Remote Auto ○ Fast-time ○ Real-time ○ Parallel

● Standby ○ Button ○ Key ○ Joystick

Prop. RPS 128 0
Rudder Angle 1800 0
Bow RPS 1500 0
Stern RPS 1500 0

Measured Ship Status
Longitude GPS: 0.0 deg
Latitude GPS: 0.0 deg
X: 1713.8 m
Y: 61.45 m
Course Speed GPS: 0.0 m/s
Course Angle GPS: 0.0 deg
Course Rate GPS: deg/s
Heading Speed GPS: m/s
Heading Angle GPS: 0.0 deg
Heading Rate GPS: 0.0 deg/s
Heading Angle Ele: 0.0 deg
Heading Angle Mag: 0.0 deg
Propeller RPS: 128 Hz
Rudder Angle: 0.0 deg
Bow RPS: 1500 Hz
Stern RPS: 1500 Hz

Figure 19: The structure for control of the Pallas model vessel.

https://doi.org/10.24868/issn.2631-8741.2020.014

To achieve the autonomy of surface ships, the experiments of this model vessel started with building a motion model building to enhance the control accuracy and verify the control ability between a virtual vessel and a real vessel. For path planning, the Fast Marching Method (FMM) and Fast Sweeping Method (FSM) are introduced to generate a global path. Moreover, Artificial Potential Field Method (APF) is mainly used to generate a local path, which is used for obstacle avoidance. Considering the ship motion model and its maneuverability, global path planning and local path planning have a better performance in the real-ship experiment. To follow the path generated by FMM, FSM, and APF, self-turning PID and LOS are combined to suit complex routes. For the vessel's docking and undocking research, a heuristic algorithm plus predictive control algorithm or classical control algorithm like PID shows the effectiveness in theories and will be implemented in a real situation in this model vessel. Considering the communication ability, remote control can be realized through different equipment and networks. WIFI, 4G or 5G, and satellite communication are applicable in short, medium, and long-distance scenarios respectively. The remote control has been achieved on a 7m KVLCC2 model (see in Figure 20) ship in the same structure by 4G. These algorithms and related experiments focus on a single ship control and neglect the communication ability. Considering the development of autonomous ships in real situations, ship formation control has got more attention, especially in cargo delivery in inland rivers and short-sea. This model vessel can be utilized to verify the effectiveness and practicability of different formation strategies and control strategies. Detailed information about the design and structure of the model ships is referred to [16].

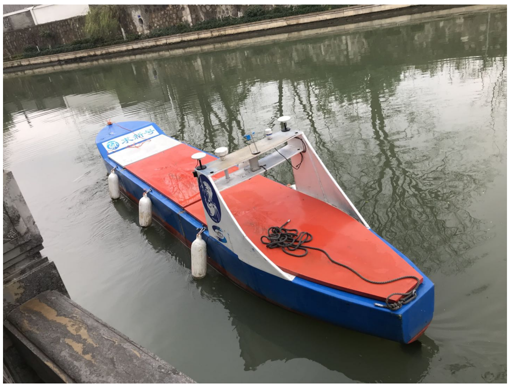


Figure 20: A 7m KVLCC2 model ship.

3.6 Tito-Neri and Grey Seabax

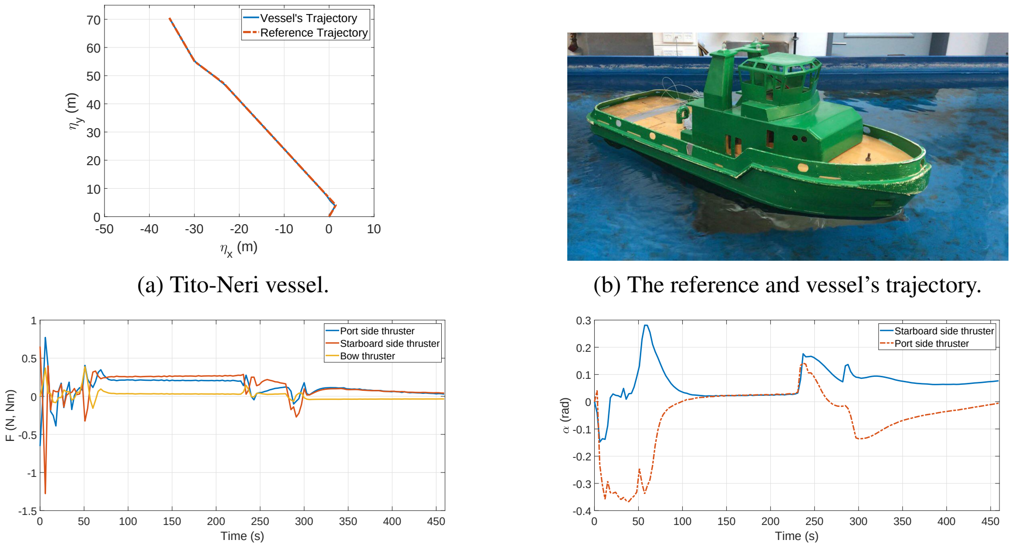
The Grey Seabax is a 1:30 scale model of an offshore support vessel named "Seabex One", and the Tito Neri is a 1:30 scale model of a harbor tug. Along with several control approaches possible to to control these vessels, model-based control of the Tito-Neri is also possible since its maneuvering model is extracted [9]. Several advanced control approaches have been designed for Tito-Neri and Seabax including Model Predictive Control (MPC) and neural networks-based adaptive control [10, 11]. Figure 21 shows a trajectory tracking simulation result of the Tito-Neri vessel using an MPC approach.

Grey Seabax is mainly used for outdoor trajectory tracking and path following applications. A radio link based on Xbee modules uses a so-called point to point connection. In the situation that there is one control station and one ship, a sequential connection is sufficient. Problems arise when there are multiple ships in the communication network since this type of communication requires a simultaneous and parallel connection. Figure 23 underlines the difference between the two communication protocols.

The way to move from a point-to-point link to a parallel and simultaneous communication infrastructure is using the Internet Protocol (IP). The IP protocol allows devices running different operating system to communicate in an asynchronous way. A unique IP address is allocated to every connected device. This allows them to receive and send data to every node in the network simultaneously. In the case of model scale ships, a router using WiFi will enable communication, however using a 5G device would obtain the same effect, without the constraint imposed by the range of reachability due to the modem signal power. The difference between the radio link communication (using Xbee radio modules) and WiFi (using both a modem or a 5G device) is depicted in Table 3.

In order to allow the model scale ships to share their state to the other nodes in the network (e.g. to the monitoring control station and/or the other ships in the same network), there is the need to design a software framework that manage the protocols and the messages between the different agents. This can be obtained using the Robotic Operating System (ROS).

ROS is a platform that is able to connect different algorithms and sensors on a local network. The system links different devices and software together and allows data to be sent and recognized by all the agents involved. ROS



(a) Tito-Neri vessel.

(b) The reference and vessel's trajectory.

(c) Applied thruster forces.

(d) The angle of thrusters.

Figure 21: Model predictive trajectory tracking performance of Tito-Neri.

Figure 22: Grey Seabax model vessel.

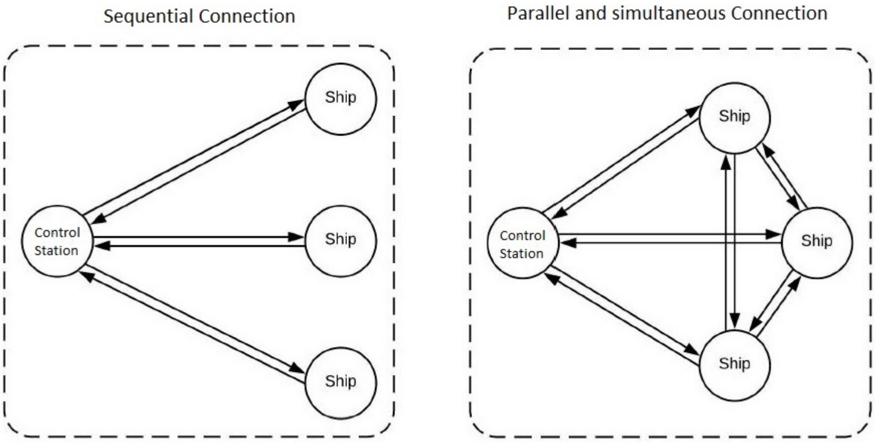


Figure 23: Communication structures of the vessels.

https://doi.org/10.24868/issn.2631-8741.2020.014

System	XBee	WiFi	5G
Range [m]	90	50	Whenever there is coverage
Frequency [GHz]	2.4	2.4 or 5	2.4 or 5
Protocol	80.2.15.4, ZigBee, DigMesh	IP-protocol (802.11)	IP-protocol (802.11)
Data Rate	250 Kbps	300 or 600 Mbps	300 or 600 Mbps
Antenna	PCB	.	.
Latency [ms]	40	3 - 4	3 - 4

Table 3: Comparison between different communication systems.

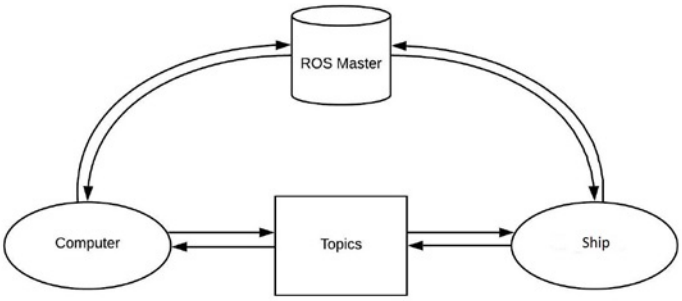


Figure 24: ROS structure.

enables different parts of a system to communicate via WiFi using nodes. Every node can publish or subscribe to a topic that contains data and a device can be designed to have multiple nodes that subscribe and publish to a different or the same topics. In order to use ROS, a ROS-master platform need to be settled up on one of the devices, usually on the control monitoring computer that acts as a hub. This ROS-master device has its own IP-address and all the parts must connect to the master via this address. A node has to say to the master that it wants to send data and publish that data to a certain topic. The master stores this information. When there is another node that wants to get data, it has to say to the master that it wants to subscribe to an earlier made topic containing the needed data. The master then says to the subscribing node which node is publishing on that topic and on which address the publishing node can be found. From there on the communication takes place between the two nodes. In short ROS-master is only needed to control and to setup the communication between nodes. In Figure 24 is a schematic overview of the possibilities in ROS. It offers standardized functionalities performing hardware abstraction and this flexibility allows the ROS to be compatible with different coding languages.

The communication can be divided in a local and a global level. The local level includes all the communication on the ship itself and the global level is the overlaying communication between the ship and a monitoring control station. For the local level communication layer, to achieve faster communication, a WiFi connection had to be integrated. The system consists of two embedded boards: one is responsible for reading out the GPS and IMU data, while the second is designed to manage the data traffic concerning the hardware and the sensors. In order to implement the WiFi connection, a laptop is used. The laptop is connected to the embedded boards via a serial connection. Using a Python script the laptop acts like a 'bridge' to connect the sensors on the model scale ships to the ROS-network. On a global level, Python had to be able to send and receive data from Matlab via WiFi and forward it to the Arduino. In order to start the communication between the devices, a ROS master was required. A separate computer running Ubuntu, is the ROS master in order to reduce latency and avoid crashes of the Raspberry Pi. (For a schematic overview of the ROS environment, refer to Figure 24.) On the PC, several topics and nodes will be initiated within Matlab using the ROS-toolbox. Matlab can then publish a request for data on topic A, which Python is subscribed to. As soon as Python receives this request, it forwards it to the Arduino. This Arduino then reads data from the GPS and the IMU and sends it back to Python. Python then publishes this data on a different topic B. In return Matlab can subscribe to topic B and thus receive the data. In the interest of preventing data loss, every data stream is being placed on a different topic. In total there are three different processes requiring topics: The initialization of the ship, the transmission of telemetry and the termination process. These three processes each get two topics, one for forward communication and one for backward communication. Lastly there is one extra topic that contains the vector with the required RPM's and azimuth for the thrusters on the ship. A flow chart representation at network level is given below.

Figure 25 shows the overall control structure of the model scale ship. A division between high level control and low level control is made. The high level control governs the system as a whole and heading control can be performed such that the ship is able to navigate towards a waypoint with a constant RPM set point. To be able to

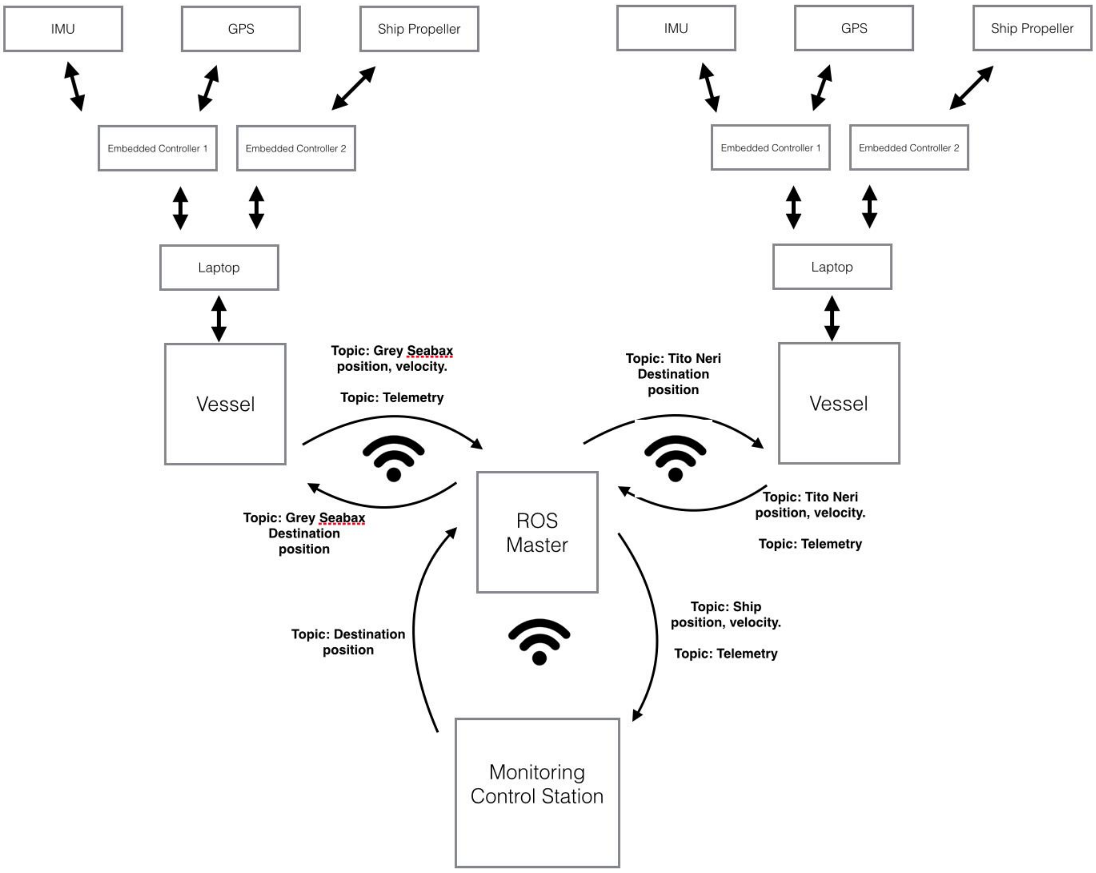

IMU
GPS
Ship Propeller
Embedded Controller 1
Embedded Controller 2
Laptop
Vessel
IMU
GPS
Ship Propeller
Embedded Controller 1
Embedded Controller 2
Laptop
Vessel
Topic: Grey Seabax position, velocity.
Topic: Telemetry
Topic: Tito Neri Destination position
Topic: Grey Seabax Destination position
ROS Master
Topic: Tito Neri position, velocity.
Topic: Telemetry
Topic: Destination position
Topic: Ship position, velocity.
Topic: Telemetry
Monitoring Control Station

Figure 25: Conceptual block diagram of the control structure.

perform trajectory tracking, the velocity has to be controlled. This means that the constant RPM set point in the low level control will be the output of the trajectory generator algorithm in the high level control loop.

4 Conclusion

In this paper, a description has been given about the Collaborative Autonomous Shipping Experiment (CASE) and its objectives have been discussed. Then, several experiments that are considered for this year's CASE have been introduced. Next, the participating vessels are introduced and some of their capabilities are discussed. CASE 2020 will be held in parallel and as a part of the International Ship Control Systems Symposium (iSCSS) 2020, at Delft University of Technology.

Acknowledgement

This research is supported by the Researchlab Autonomous Shipping (RAS) of Delft University of Technology, The Netherlands, Innovation and Entrepreneurship Agency (VLAIO) under the grant agreement No. HBC.2019.0045 associated with the project SSAVE of the KU Leuven, National Key R&D Program of China (2018YFB1601505), and the University of Genova, Italy.

References

[1] Alessandri, A., Donnarumma, S., Luria, G., Martelli, M., Vignolo, S., Chiti, R., Sebastiani, L., 2015. Dynamic positioning system of a vessel with conventional propulsion configuration: Modeling and simulation , 725–734doi:10.1201/b17494-97.

[2] Alessandri, A., Donnarumma, S., Martelli, M., Vignolo, S., 2019. Motion control for autonomous navigation in blue and narrow water using switched controllers. Journal of Marine Science and Engineering 7, 196.

[3] Altosole, M., Martelli, M., 2017. Propulsion control strategies for ship emergency manoeuvres. Ocean Engineering 137, 99 – 109.

[4] Bibuli, M., Caccia, M., Lapierre, L., 2010. Virtual target based coordinated path-following for multi-vehicle systems. IFAC Proceedings Volumes 43, 336–341.

[5] Bibuli, M., Singh, Y., Sharma, S., Sutton, R., Hatton, D., Khan, A., 2018. A two layered optimal approach towards cooperative motion planning of unmanned surface vehicles in a constrained maritime environment. IFAC-PapersOnLine 51, 378–383.

[6] Donnarumma, S., Figari, M., Martelli, M., Vignolo, S., Viviani, M., 2018. Design and validation of dynamic positioning for marine systems: A case study. IEEE Journal of Oceanic Engineering 43, 677–688.

[7] Donnarumma, S., Martelli, M., Vignolo, S., 2015. Numerical models for ship dynamic positioning. MARINE 2015 - Computational Methods in Marine Engineering VI , 1078–1088.

[8] Figari, M., Alessandri, A., Donnarumma, S., Vignolo, S., Martelli, M., 2015. System control design of autopilot and speed pilot for a patrol vessel by using lmis. doi:10.1201/b18855-76.

[9] Haseltalab, A., 2019. Control for Autonomous All-Electric Ships. Ph.D. thesis. Delft University of Technology.

[10] Haseltalab, A., Negenborn, R.R., 2019a. Adaptive control for autonomous ships with uncertain model and unknown propeller dynamics. Control Engineering Practice 91, 104116.

[11] Haseltalab, A., Negenborn, R.R., 2019b. Model predictive maneuvering control and energy management for all-electric autonomous ships. Applied Energy 251, 113308.

[12] Martelli, M., Figari, M., 2017. Real-time model-based design for codlag propulsion control strategies. Ocean Engineering 141, 265 – 276.

[13] Martelli, M., Figari, M., Zaccone, R., 2018. An optimization tool for ship route planning in real weather scenarios.

[14] Martelli, M., Zaccone, R., 2018. A random sampling based algorithm for ship path planning with obstacles. doi:10.24868/issn.2631-8741.2018.018.

[15] Van Baelen, S., Drijkoningen, S., Moons, C., Afzal, M.R., Slaets, P., 2019. Experimental identification of the dynamic characteristics for a 1/25 scale model of the watertruck+ self-propelling barge, in: OCEANS 2019-Marseille, IEEE. pp. 1–7.

[16] You, X., Ma, F., Lu, S., Liu, J., Yan, X., 2020. An integrated platform for the development of autonomous and remote-control ships, in: Porceedings of the 19th Conference on Computer and IT Applications in the Maritime Industries (COMPIT 2020), pp. 316–327.

[17] Zaccone, R., Martelli, M., 2020. A collision avoidance algorithm for ship guidance applications. Journal of Marine Engineering & Technology 19, 62–75.

[18] Zaccone, R., Martelli, M., Figari, M., 2019. A colreg-compliant ship collision avoidance algorithm, in: 2019 18th European Control Conference (ECC), pp. 2530–2535.