Path-Following with LiDAR-based Obstacle Avoidance of an Unmanned Surface Vehicle in Harbor Conditions

Jose Villa, Jussi Aaltonen, and Kari T. Koskinen

***Abstract*—This article studies the design, modeling, and implementation challenges of a path-following with obstacle avoidance algorithms as Guidance, Navigation, and Control (GNC) architecture of an Unmanned Surface Vehicle (USV) in harbor conditions. First, an effective mathematical model is developed based on System Identification (SI), validating the USV model with field-test data. Then, a guidance system is addressed based on a Line-Of-Sight (LOS) algorithm, which uses a LiDAR as the main perception sensor for the obstacle avoidance algorithm. The GNC architecture uses a modular approach, including obstacle detection, path-following, and control in the USV platform. Finally, an implementation challenge in two control scenarios, simulation and field-test, is addressed to validate the designed GNC architecture.**

***Index Terms*—Path-following, obstacle avoidance, system identification, model-validation, USV.**

I. Introduction

DIFFERENT aspects in society, such as occupational safety and security, as well as longer operation times, have led to a demand for research and development of innovative autonomous systems. These autonomous systems include Unmanned Surface Vehicles (USVs) and Autonomous Underwater Vehicles (AUVs) as main offshore vehicles. USVs can be studied for numerous potential applications in an advantageous way, such as scientific research, environmental missions, or ocean resource exploration.

To accomplish all these offshore applications, the availability of an adequately accurate USV model is imperative for simulation study purposes, controller design, and development. USV models are commonly reduced-order for horizontal plane control (surge, sway, and yaw motions). These models have been used in numerous studies, such as the 3 Degrees-Of-Freedom (DOFs) horizontal plane models for maneuvering based on the rigid-body-kinetics [1], or the model representation of Nomoto [2] for heading autopilot among others. Furthermore, System Identification (SI) can be included to obtain an accurate model for simulation studies using field-test data [3]. SI using the Nomoto model, SI based on Particle-Swarm-Optimization (PSO), and SI for the 3-DOFs ship maneuvering model are presented for USV approaches in [4], [5], and [6], respectively. Moreover, [7] proposes a nonlinear modeling scheme for a waterjet propulsion USV system. However, both the Nomoto model and [7] use the rudder angle as model input.

For controller design and development, safe and efficient control of USVs depends heavily on proper Guidance, Navigation, and Control (GNC) systems with sensing, state estimation, and situational awareness capabilities. A path-following is adopted as a guidance system in this work, as it is easier to implement than trajectory tracking, and is closer to practical engineering. For path-following in offshore operations, most of the studies have been done in a free obstacle path scheme using Line-Of-Sight (LOS) algorithm [8] or a guidance-based algorithm [9]. Recent progress on path-following is focused more on dealing with external disturbances to improve the control performance [10].

To perform missions in real-world environments, USVs are required to have the ability to detect obstacles, recognize and track targets, and map environments. To obtain situational awareness of the USV, passive (e.g., stereo cameras) and active (LiDAR or radar) perception methods have been used in numerous studies. However, the majority of the obstacle detection techniques rely on depth measurements, in which LiDAR sensors are the most robust method of obtaining depth data. In [11], a 3-D scanning LiDAR performance was focused on the marine environment for a USV. However, 2-D or 3-D representation of LiDAR can suffer the clutter phenomena of a marine environment [12]. In this paper, SICK MRS1000 LiDAR [13] solves this effect, as it has 4 spread-out scan planes and a multi-echo analysis that avoids the noise produced by fog, rain, or dust. Also, the working range of this device is from 0.2 to 64 meters with a 275° aperture angle, being completely suitable for small USVs in harbor operations.

An approach to combine both path-following and obstacle avoidance capabilities can be the use of safety boundary boxes around a static or moving obstacle. In [14], the use of safety boundary boxes was studied for collision avoidance, where a corresponding collision boundary box is associated with each obstacle. The goal of this study is to find the optimal path while avoiding any collision boundary boxes. In addition, [15] includes a multi-layer obstacle avoidance based on a single LiDAR and presents an efficient solution to USV path planning in the case of sensor errors and collision risks, defining a safety box for obstacle recognition.

Besides to the safe and efficient control, USVs operating in populated waterways may require compliance with existing rules. These rules can be the collision regulations defined by the convention on the international regulations for preventing

The authors are with the Mechatronics Research Group (MRG), Tampere University (TAU), 33720, Tampere, Finland (e-mail: jose.villa@tuni.fi; jussi.aaltonen@tuni.fi, kari.koskinen@tuni.fi).

collisions at sea (COLREGs) [16]. Regarding COLREGs in USVs, [17] reports preliminary research results of a novel automatic obstacle avoidance approach. Furthermore, [18] studies an under-actuated USV for collision avoidance, ensuring path-following while abiding by the COLREGs. However, neither of these studies have been implemented in field-tests.

Once the GNC algorithm has been tested in a simulation environment, it can be implemented in a field-test. This implementation uses Robot Operating System (ROS), which is a flexible framework for writing robot software [19]. ROS provides the necessary tools to access sensors' data, process it, and generate a response for the vehicle actuators. To link all USV sensors and actuators in the same ROS network, rosserial provides a tool for connecting embedded computers with the rest of the system [20]. As an example of this system connectivity, [21] designs and implements a low-cost, high performance, and generic multi-layer ROS-based architecture for autonomous systems. In addition, there are several examples of state-of-the-art autonomous vehicles using ROS, such as self-driving taxis [22].

In this study, the USV model is obtained with a different approach than using the rudder angle as model input. The same model combines the waterjet and USV dynamics by using the joystick commands as model input. For obstacle avoidance, a safety boundary box is selected, providing fast decision-making capabilities due to its simplicity, low data transfer, and modular approach. In addition, a LOS algorithm is implemented without compensation of environmental elements, to focus further on the obstacle avoidance. This obstacle avoidance for a path-following algorithm uses a modular ROS architecture to provide a simple, computationally cheap, and easy implementation. Hence, obstacle avoidance capabilities in field-tests are the main focus of this research, to allow the application of COLREGs in the future.

In this work, a model-based GNC architecture for a USV is proposed for path-following with obstacle avoidance using a LiDAR as a perception method. In Section II, the USV modeling and simulation are presented using SI as the tool to define the maneuvering model. Then, in Section III, the control methodology design is included using the LOS-based guidance system for control. Obstacle avoidance capabilities are involved in defining a safety boundary box around the detected object and using its corners for the new waypoints of the path-following. Finally, in Section IV, the implementation of a GNC architecture is described as modular and multilayer, allowing the fast check for the optimum operability of the vehicle capabilities. Control scenarios in both simulation and field-test are shown to validate the proposed GNC architecture.

II. USV MODELING AND SIMULATION

A. Overview of under-actuated USV

The USV used in this study is an aluminum hull with thrust vectoring waterjet propulsion. The USV has optimal maneuverability using the twin waterjet configuration that facilitates the movement in all directions without bow and stern thrusters (see Fig. 1). The motion of the USV is simplified from six to three DOFs. These 3-DOFs are surge, sway, and yaw motions, while ignoring roll, pitch, and heave.

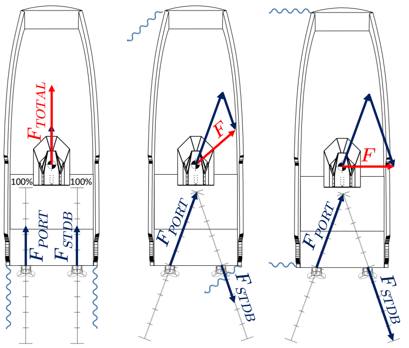


Fig. 1. Twin waterjet USV maneuvering.

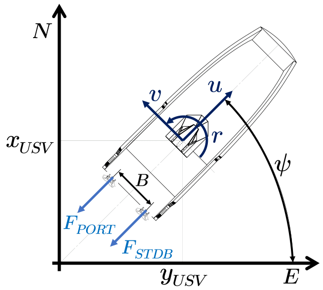


Fig. 2. Simplified model of the considered vehicle.

Fig. 2 shows a simplified model of the twin waterjet propelled USV. The waterjet propelled USV uses the Port and Starboard (STDB) waterjets to provide thrust forces when moving either forward, backward, sideways, or performing turns. Also, Fig. 2 illustrates the position and orientation of the USV. Due to the use of GPS-Compass as the navigation system, a coordinate transformation is applied to obtain the absolute position of the USV in the planar coordinate system. This transformation is between World Geodetic System 84 (WGS84), which provides longitude and latitude [l, μ] of the USV, and ETRS-TM35FIN [23], which shows the North-East-Down (NED) (x_{USV}, y_{USV}) position. The USV heading ψ is described using the attitude (Euler angles). Furthermore, the body-fixed reference frame is used for relative positioning, with linear [u, v] and angular [r] velocities.

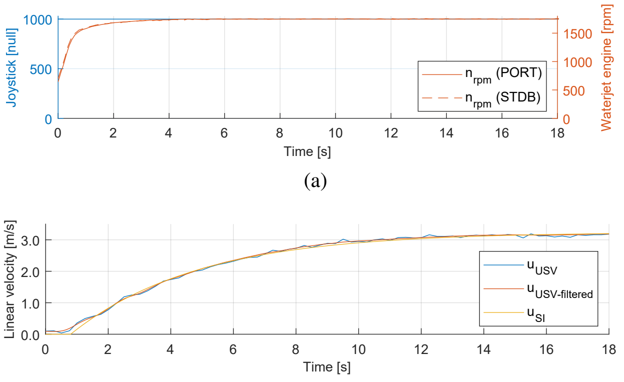
B. Vehicle modeling

The development of an effective maneuvering model will facilitate the GNC algorithms design and simulation. The guidance system is defined as a path-following control, where the USV moves forward with reference speed u at the same time as minimizes the cross-track error e to the predefined path. Several heading controllers for marine crafts are based on the model representation of Nomoto [2]. The Nomoto autopilot model can be derived from the linearized maneuvering model of the USV, and Nomoto's second-order model refers to

$$\frac{r}{\delta}(s) = \frac{K_p(1+T_z s)}{(1+T_{p1}s)(1+T_{p2}s)}. \tag{1}$$

where r is the angular velocity, and δ is the rudder angle.

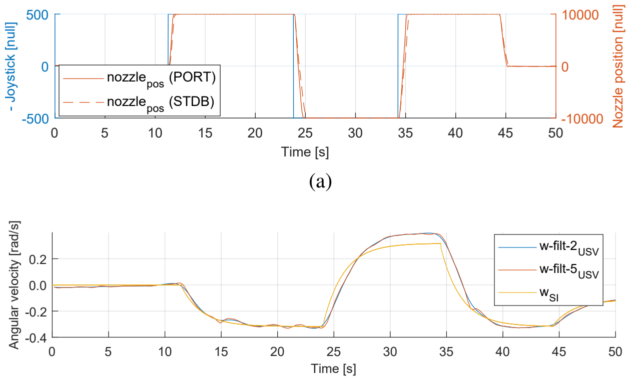
The model representation of Nomoto has the main advantage of its simplicity. Its parameters can be defined directly



(a)

(b)

Fig. 3. USV in forward motion (surge): (a) Waterjet engine response for a constant Joy_{surge} input, (b) Model-validation using SI for surge velocity.



(a)

(b)

Fig. 4. USV in yaw motion: (a) $nozzle_{pos}$ response for a zigzag Joy_{yaw} input, (b) Model-validation using SI for angular velocity.

from the field-testing data, and thus, the hydrodynamic derivatives do not need to be computed explicitly.

C. Model-validation using System Identification

Propulsive thrust in the water jet propulsion unit is created by the reaction force, which is caused by the kinetic energy of the mass flow generated by the propulsion unit impeller. The nozzle position $nozzle_{pos}$ changes the direction of the jet flow coming out of the nozzle, which creates the force needed for turning. Hence, the total thrust force (F_{Total} in Fig. 1) is a combination of the engine rpm of the waterjet n_{rpm} and $nozzle_{pos}$. n_{rpm} is directly obtained from the waterjet engine, and $nozzle_{pos}$ is a variable from -10.000 to 10.000, with 0 as the neutral position (forward motion). Fig. 3a and 4a show the settling time effect for joystick input in both n_{rpm} and $nozzle_{pos}$, which are related to the waterjet dynamics. A combination of a 1D lookup table with second-order transfer functions is proposed to simplify the USV modeling, including the effect of both waterjet and USV dynamics. The parameters for the 1D Lookup table are obtained from field-testing data for the forward and right-turn motions and are presented in Table I. These parameters are used to map the input to output values, approximating a mathematical function.

TABLE I
1D Lookup Table parameters

Joy_{surge}	400	500	600	700	800	900	1000
n_{rpm}	690	920	1110	1300	1480	1650	1820
Joy_{yaw}	0	50	150	200	250	300	400
$nozzle_{pos}$	0	1175	3500	4665	5830	7000	9325

TABLE II
Surge and Yaw motion parameters

Motion	T_z	T_{p1}	T_{p2}	K_p	T_d
Surge	0.17563	4.08900	0.17299	2.930×10^{-3}	0.8
Yaw	0.09835	1.81108	0.00144	-3.177×10^{-5}	0.0

The mathematical model of the USV includes a constant surge velocity with a variable heading of the USV. As it is presented in [4], the behavior obtained with MATLAB SI tool [24] is close to the Least Squares Support Vector Machines (LS-SVM) approach. Thus, the MATLAB SI tool has been chosen for simplicity to obtain both surge and yaw models.

After studying several SI models, the surge motion is defined in (2) as a transfer function with one zero (T_z), two poles (T_{p1}, T_{p2}), a process gain (K_p), and an Input/Output delay (T_d). The engine rpm of the waterjet n_{rpm} is declared as input, while surge velocity u is the output of the transfer function. Fig. 3b shows both SI and field-tests (raw and filtered USV linear velocity) step response for a constant joystick Joy_{surge} input value.

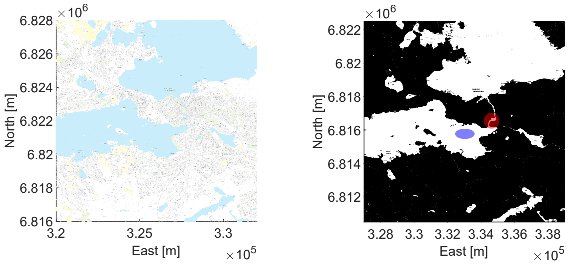
$$\frac{u}{n_{rpm}}(s) = \exp(-T_d s)\frac{K_p(1+T_z s)}{(1+T_{p1}s)(1+T_{p2}s)}. \tag{2}$$

The yaw motion is defined using the same Nomoto's second-order model declared in (1). However, the model input is the nozzle position of the waterjet $nozzle_{pos}$ instead of the original rudder angle δ used in the Nomoto's model. The nozzle position generates different angular velocity for a constant engine rpm. The output is the angular velocity r for yaw motion, obtaining the yaw angle ψ from its integration. Fig. 4b shows both SI and field-tests (two filtered plots from the yaw angular velocity) step response for a constant Joy_{surge} input value, and a zigzag in Joy_{yaw}. The variance between the left and right turns is produced by the misalignment of the center of mass and center of buoyancy in the USV (inclination of 1.5° to the right).

Table II includes the identified model parameters for the transfer functions of surge and yaw motions included in the mathematical model. The variables u and r are obtained according to their 1D Lookup tables and transfer functions (from (2) and (1) respectively). Thus, there is a different set of dynamics comparing to a USV with a propeller and rudder, a different set of parameters, but the approach still works.

D. Simulation environment

A simulation experiment is essential to verify whether the GNC algorithm is valid in the USV workspace. The map of the control scenario must be processed, requiring the definition of two separate matrices. The first matrix includes RGB data per pixel, and the second one contains the planar coordinate



(a) (b) BW map

Fig. 5. Map processing: (a) RGB map with ETRS-TM35FIN planar coordinate system, (b) BW map after map processing where red area refers to Control Scenario I and blue area to Control Scenario II in Section IV.

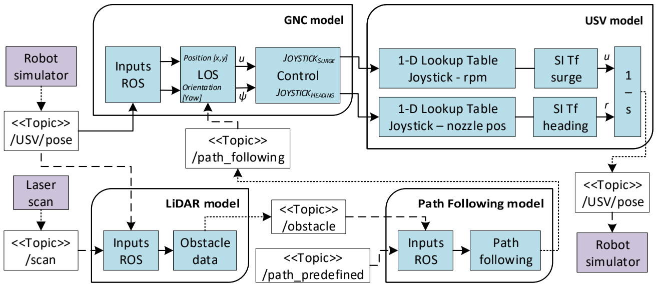


Fig. 6. Block-diagram representation of the simulated USV model.

system, which is added by reading the world file. Then, the RBG image, shown in Fig. 5a, is converted to Black and White (BW), and any noisy points are removed using standard image processing software (see Fig. 5b). Finally, the BW image is converted into a binary occupancy grid in MATLAB software, which is also the simulation environment for this study. A binary occupancy grid uses true values to represent the occupied workspace (black), such as obstacles, and false values for the free workspace (white). This map is used as a simulation scenario for the GNC system, allowing the inclusion of external objects in a specific position, and the location of the USV at a certain point on the map. In addition, this map provides the possibility for future path planning research, in which a binary occupancy grid is essential.

The GNC controller for a simulated USV is created and computes the control commands to follow the desired path, which is a set of waypoints defined explicitly. The simulated vehicle has kinematic equations for the motion of an under-actuated USV, obtained from the SI process mentioned in this Section. Furthermore, situational awareness capabilities are included using a laser scan object. The laser scan message is processed to extract scan ranges and angles, and compute these variables to obtain the position of the detected obstacle. This is a similar approach to the one obtained from the USV LiDAR. The controller receives the vehicle pose and laser scan data from the simulated vehicle and sends joystick commands to drive the vehicle on the given path. The schematic of the simulated USV model is shown in Fig. 6. The main goal of the mathematical model of the USV is to test the obstacle avoidance capabilities. Hence, the hypothesis of independence between surge and yaw dynamics is acceptable even though it does not include drift or environmental disturbances.

III. GNC system with Line-Of-Sight based model

A. Line-of-sight guidance system

A path-following algorithm aims to reach every waypoint of a predefined path independent of time. A commonly used method for path-following is the named LOS guidance, which is adopted as a reference trajectory for the USV in this study. A LOS vector from the surface vehicle to the next waypoint or a point on the path between two waypoints can be used for heading control, similar to [1].

For lookahead-based steering, the course angle is separated into two parts, defined as

$$\chi_d(e) = \chi_p + \chi_r(e). \tag{3}$$

where $\chi_p = \alpha_k$ is the path-tangential angle defined in (4), while χ_r is a velocity-path relative angle, which ensures that the velocity has the direction towards a point on the path that is in a lookahead distance $\Delta(t) > 0$ along of the direct projection [25].

$$\alpha_k = \operatorname{atan2}(y_{k+1} - y_k, x_{k+1} - x_k). \tag{4}$$

The steering law can be interpreted as a saturating control law

$$\chi_r(e) := \arctan(-K_p e - K_i \int_0^t e(\tau) d\tau). \tag{5}$$

where the proportional gain is $K_p = 1/\Delta(t) > 0$, and $K_i > 0$ represents the integral gain. This lookahead-based steering law is equivalent to saturated proportional control law and $e(t)$ is the cross-track error given by

$$e(t) = -[x_{USV}(t) - x_k]\sin(\alpha_k) + [y_{USV}(t) - y_k]\cos(\alpha_k). \tag{6}$$

The lookahead-based steering can be implemented related to the heading controller applying the transformation shown in (7). The variable sideslip (drift) angle β [1] has been avoided to simplify the steering law.

$$\psi_d = \chi_d - \beta = \chi_p + \chi_r - \beta. \tag{7}$$

The switching mechanism is defined as a circle of acceptance for surface vehicles [1], which selects the next waypoint as a lookahead point if the position of the USV lies within a circle with radius R around (x_{k+1}, y_{k+1}). This circle of acceptance is defined as

$$[x_{USV}(t) - x_{k+1}]^2 + [y_{USV}(t) - y_{k+1}]^2 \leq R_{k+1}^2. \tag{8}$$

where, if the time surface vehicle position $(x_{USV}(t), y_{USV}(t))$ satisfies (8), the next waypoint (x_{k+1}, y_{k+1}) needs to be selected. Radius R is equal to two USV lengths ($R = 2L$). LOS guidance system and circle of acceptance are shown in Fig. 7.

B. Obstacle avoidance using LiDAR

Obstacle avoidance capabilities are an essential role of the GNC algorithm, as it provides the safety feature for the autonomous vehicle operation. In this study, LiDAR has been used as the main perception sensor. This sensor provides obstacle information in Cartesian coordinates for each beam with a 275° aperture angle. This position gives the possibility

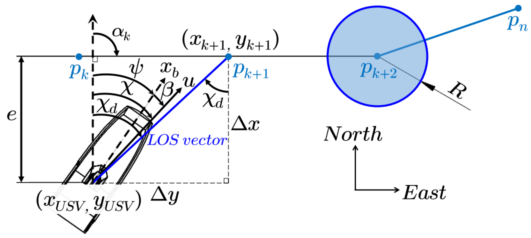


Fig. 7. LOS guidance system and circle of acceptance.

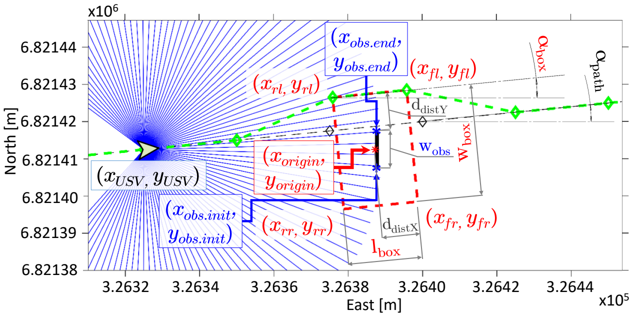


Fig. 8. Safety boundary box for obstacle recognition using LiDAR: Safety boundary box (dotted red line) is generated to create the new path for the USV (dotted green line).

to locate the obstacle in the relative map on the vehicle, or absolute coordinates. Also, as the LiDAR used in this study has four spread-out scan planes, if there is an object on a slope like the shore of an island, the closer detected point is always used for the obstacle avoidance.

Once that the obstacle has been located relative to the USV, a safety boundary box is defined around this object, providing additional information for the GNC algorithm. The implementation of this boundary box is shown in Fig. 8. This boundary box is defined in (9) as a rectangle-shaped box with width w_{box} and length l_{box} parameters. Both dimensions are determined with the obstacle's width w_{obs} and length l_{obs}, and by a predefined fixed parameter for safety distance d_{safe_X} and d_{safe_Y} in X and Y axis, respectively. The boundary box origin (x_{origin}, y_{origin}) is defined according to the absolute USV position and the LiDAR data defined in (10).

$$\begin{cases} w_{box} = d_{safe_Y} + \frac{w_{obs}}{2} \\ l_{box} = d_{safe_X} + \frac{l_{obs}}{2} \end{cases}. \tag{9}$$

$$\begin{bmatrix} x_{origin} \\ y_{origin} \end{bmatrix} = \begin{bmatrix} x_{USV} \\ y_{USV} \end{bmatrix} + \begin{bmatrix} \frac{x_{obs,init}+x_{obs,end}}{2} \\ \frac{y_{obs,init}+y_{obs,end}}{2} \end{bmatrix}. \tag{10}$$

As the obstacle avoidance is designed for harbor operations, the shape of this boundary box is selected according to the distance from the left to the right side walls of the harbor route. Also, the shape can be related to the vehicle position according to the GPS (once it enters a predefined harbor area). The safety boundary box has the same angle α_{box} as the slope of the predefined path α_{path}, allowing a continuous path operation. The corners of the boundary box are used as new waypoints of the path-following in the GNC algorithm, continuing with the LOS control approach. The obstacle avoidance algorithm also selects the new waypoint depending on which side is a closer trajectory to the vehicle. The first waypoint can be rear left (x_{rl}, y_{rl}) or rear right (x_{rr}, y_{rr}), while the next one is the correspondent side of the front corner. Equations (11) and (12) define all boundary box corners, where a and b refers to rear/front or left/right respectively, and i and j to the sign for the second element. $R(\alpha_{path})$ is the rotation matrix in the XY-plane counterclockwise through α_{path}.

$$\begin{bmatrix} x_{ab} \\ y_{ab} \end{bmatrix} = \begin{bmatrix} x_{origin} \\ y_{origin} \end{bmatrix} + R(\alpha_{path}) \begin{bmatrix} i\frac{w_{box}}{2} \\ j\frac{l_{box}}{2} \end{bmatrix}. \tag{11}$$

$$\begin{bmatrix} x_{ab} \\ y_{ab} \end{bmatrix} = \begin{cases} i = 1 & \text{if } a = f \\ i = -1 & \text{if } a = r \\ j = 1 & \text{if } b = l \\ i = -1 & \text{if } b = r \end{cases}. \tag{12}$$

The new waypoints are selected for the LOS algorithm to perform a smooth USV trajectory, including additional waypoints to the predefined path. If there is an obstacle outside the USV predefined path, the boundary box is created, but it does not affect the next waypoint. In addition, the algorithm continuously updates the safety boundary box position, being suitable for the avoidance of stationary and slow-motion objects. Once that the obstacle has been avoided, the USV follows the predefined path until it recognizes another object in its trajectory or it reaches the ending waypoint.

C. Surge and Yaw controllers

A LOS path-following controller can be designed for a USV by representing the desired path by waypoints (x_n, y_n). This controller sends heading commands to the yaw controller to match the predefined path. The main control system is based on two separate PID controllers for surge and yaw control, and their parameters are obtained by using Rapid Control Prototyping (RCP) during the field-tests. The surge controller keeps the USV at a predefined constant speed.

The feedback loop incorporates low-pass and notch filters to reduce motions induced by waves [1]. A first-order low-pass filter with a time constant T_f can be designed according to

$$h_{lp}(s) = \frac{1}{1 + T_f s}. \tag{13}$$

This filter is used to suppress forces over the frequency $1/T_f$. Although this criterion is hard to specify for USVs, it has been defined as T_f = 0.1 s time constant after simulation and implementation results.

The bandwidth of the controller ω_b can be close to or within the range $\omega_{min} < \omega_e < \omega_{max}$ of the wave spectrum for small USVs. Adding a low-pass filter in cascade with a notch filter handles this problem. However, the estimation of the notch frequency ω_n might not be accurate. Therefore, a filter structure formed by three cascaded notch filters with fixed center frequencies has been added in the feedback loop

$$h_n(s) = \prod_{i=1}^{3} \frac{s^2 + 2\xi\omega_i s + \omega_i^2}{(s + \omega_i)^2}. \tag{14}$$

The center frequencies of the notch filters have been assumed as ω_1 = 0.1 rad/s, ω_2 = 0.2 rad/s, and ω_3 = 0.4 rad/s

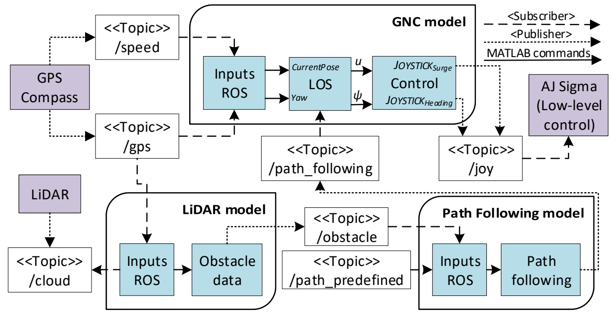


Fig. 9. Schematic of the modular GNC system: The three models are the modules included in the USV platform, which are run in separate ROS nodes.

because of the lack of big waves in the field-test conducted in harbor conditions. Hence, small wave effects have been suppressed by getting steady outputs in both surge and yaw controllers.

D. Modular system for the path planning algorithm

The path-following algorithm uses a modular approach to include obstacle detection, path-following, and control in the USV platform. Each of these modules runs a separate ROS node in the system, allowing the use of remotely operated applications. Hence, the obstacles can be positioned while the USV is operating. This approach has been previously studied in [26], implementing a path-following algorithm with a straight line. However, the algorithm in [26] did not include any collision avoidance capabilities.

All these modules use ROS messages to communicate between them, being simple to check if each module is operating correctly. Fig. 9 illustrates this modular architecture with all topics involved, defining the subscribers and publishers of each topic with the dashed or continuous line respectively. The obstacle detection module (LiDAR model) processes the data acquired from the LiDAR and the GPS-Compass and allows the GNC algorithm to obtain the position of the obstacle in absolute coordinates. Once the obstacle has been detected, its origin position, length, and width are sent to the path-following model, allowing low data transfer. This model checks if the obstacle is within the predefined path. Therefore, the path-following topic is generated depending on the interference of the obstacle in the path. This path-following topic includes the waypoints for the GNC algorithm in the control module. The GNC guidance algorithm generates the required USV heading command, sending this parameter to the controller. The controller generates the required Joystick parameters for surge Joy_{surge} and yaw Joy_{yaw} to reach the LOS values. These joystick parameters are sent to the low-level control for the USV operation, using the same commands as a manual 3-axis joystick (surge, sway, and yaw motion).

IV. Experimental validation

A. System implementation

The USV incorporates multiple mechatronics systems (see Fig. 10) to sense the surrounding environment, plan a path to a destination, and control steering and speed of the vehicle.

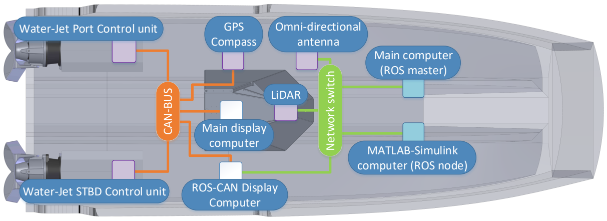

Water-Jet Port Control unit
GPS Compass
Omni-directional antenna
Main computer (ROS master)
CAN-BUS
LiDAR
Network switch
Main display computer
MATLAB-Simulink computer (ROS node)
Water-Jet STBD Control unit
ROS-CAN Display Computer

Fig. 10. System overview of the USV platform with high-level (blue boxes), intermediate-level (white boxes), and low-level control (purple boxes).

Hence, the system can be described as high-level control (computers with ROS), which performs complex computations, and low-level control (sensors and waterjet control units), which is used as an interface for basic vehicle operations. In addition, intermediate-level control (display computers) is the main link between low-level data acquisition and high-level logic operations. A LiDAR provides collision avoidance capabilities for the USV, along with the 3D map construction of the environment. All this instrumentation installed in the USV employs ROS as a framework. Thus, this framework provides necessary tools and packages to access sensor data, process it, and generate an appropriate response for different actuators.

The USV used in this study contains a Linux computer (ROS Master), which is connected to the rest of the instrumentation by a network switch via Ethernet. This Linux computer has ROS installed to send and receive the necessary commands for the USV operation. The MATLAB-Simulink computer (ROS node) is only used for testing purposes, and it includes a stand-alone ROS-node that permits a rapid prototyping procedure while testing [27]. This stand-alone ROS-node provides a solution for the labor process of C++ programming, skipping several steps to implement successfully the desired algorithm.

Two CCPilot VC display computers are used for the intermediate-level control [28]. These display computers are freely programmable, contain two CAN interfaces, Ethernet port, and they have IP66 class, being suitable for offshore environments. The ROS-CAN display computer is used to translate between CAN bus and ROS messages received from low-level and high-level systems. It uses rosserial to be connected with the rest of the system, which allows utilizing ROS with embedded systems. This display computer receives the CAN bus message from the GPS-Compass, and it can create the necessary ROS messages for the control unit (GPS position, heading, and speed of the USV) without requiring any additional converter (e.g., USB to CAN adaptor). Also, this display computer communicates via CAN bus with the main display computer, and it sends the joystick commands obtained from the high-level ROS computers, connecting the two waterjet control units with the rest of the system. The main display computer is in charge of sending joystick commands to the waterjet control units based upon priority levels. Fig. 11 shows the USV priority control level, where the ROS $/joy$ node acts as a virtual joystick with control commands Joy_{surge} and Joy_{yaw}. The steering wheel and 3-axis joystick, both forming the manual control of the USV, provides the safety feature in the autonomous algorithm.

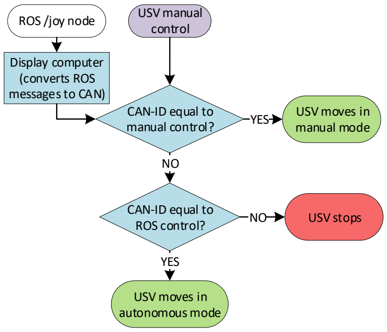

ROS /joy node
USV manual control
Display computer (converts ROS messages to CAN)
CAN-ID equal to manual control?
YES
USV moves in manual mode
NO
CAN-ID equal to ROS control?
NO
USV stops
YES
USV moves in autonomous mode

Fig. 11. Stateflow diagram for priority control level using CAN-ID: The manual joystick operation is defined as the highest priority because of safety.

B. Experimental results: Control Scenario I

The first test simulates the USV in Tammerkoski, which is a river in Tampere, Finland. This Control Scenario focuses on the obstacle avoidance implementation, not including environmental disturbances in the system. A series of GPS waypoints form the predefined path, which has a stationary obstacle in the middle of its path. Fig. 12 shows the experimental results for this Control Scenario, where the USV follows a predefined path avoiding a stationary obstacle. First, the blocking obstacles are detected by the LiDAR model, defining their origin, width, and length for the Path-following model. Then, the GNC algorithm creates a new path based on the safety boundary box, and the USV completes its trajectory.

This scenario uses the mathematical model developed in Section II, and the GNC architecture introduced in Section III. Since the harbor has a narrow channel, the safety boundary box is declared with small w_{box} and l_{box} dimensions. This test validates the GNC algorithm, generating precisely the safety boundary box, and reaching the last waypoint of the USV path.

C. Experimental results: Control Scenario II

The second test is implemented in the USV at the Pyhäjärvi lake in Tampere, Finland (see Fig. 13). The implementation consists of a path-following with collision avoidance capabilities of a static obstacle (buoy). This control scenario shows the GNC architecture capabilities for harbor conditions, but it is demonstrated in a clear obstacle area in the middle of the lake due to safety conditions.

As shown in Fig. 14, it is confirmed that the path-following with obstacle avoidance experiments were satisfactorily performed with the proposed GNC approach, whereby the black dashed, green dashed, and blue solid lines are the predefined, GNC, and actual paths respectively. The path is predefined with a series of GPS waypoints, and the USV initial position is defined as random. This randomness demonstrates the capability of the USV to reach the path from a distant initial position. Once that the blocking obstacle is detected, the GNC algorithm creates a new path based on the safety boundary box. Corresponding LOS cross-track error $e(t)$ is shown in Fig. 15, which demonstrates the correct performance of the designed GNC algorithm. In addition, Fig. 16 shows the input control values (surge and yaw parameters). The scenario has a constant surge velocity with a variable heading of the USV, obtained from the LOS guidance system.

V. Conclusion and Future Work

This article was concerned with the path-following with obstacle avoidance using a LiDAR of a USV in harbor conditions. The simulated USV model was presented to verify the designed GNC architecture. This model was based on SI methods using field-test data for surge and yaw motions of the USV. Once the USV model was validated, the GNC system with LOS based model was developed with collision avoidance capabilities. This GNC system uses a modular approach to include obstacle detection, path-following, and control in the USV platform. After designing the GNC architecture, a system implementation of the modular approach was included in the USV with three control levels (high, intermediate, and low). The experimental results show two control scenarios in both simulation and field-test, presenting the capabilities and the adequate performance of the designed GNC architecture.

Future work will include the implementation of the GNC algorithm with high-speed moving obstacles and with multiple stationary/moving obstacles, calculating a projected safety boundary box for each moving obstacle. In addition, future work will study the use of other perception sensors rather than LiDAR for long-range obstacle avoidance, such as radar.

Acknowledgment

This paper is based on the aColor project funded by Technology Industries of Finland Centennial and Jane & Aatos Erkko Foundations. The authors gratefully acknowledge the contributions of Sauli Virta from Alamarin-Jet Oy.

References

[1] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, Apr 2011.

[2] K. Nomoto, K. Taguchi, K. Honda, and S. Hirano, "On the steering qualities of ships," *Journal of Zosen Kiokai*, vol. 1956, no. 99, pp. 75–82, 1956.

[3] L. Ljung, "System identification," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–19, 1999.

[4] D. Moreno-Salinas, D. Chaos, J. M. de la Cruz, and J. Aranda, "Identification of a surface marine vessel using ls-svm," *Journal of Applied Mathematics*, vol. 2013, pp. 1–11, 2013.

[5] J. Shin, D. J. Kwak, and Y.-i. Lee, "Adaptive path-following control for an unmanned surface vessel using an identified dynamic model," *IEEE/ASME transactions on mechatronics*, vol. 22, no. 3, pp. 1143–1153, 2017.

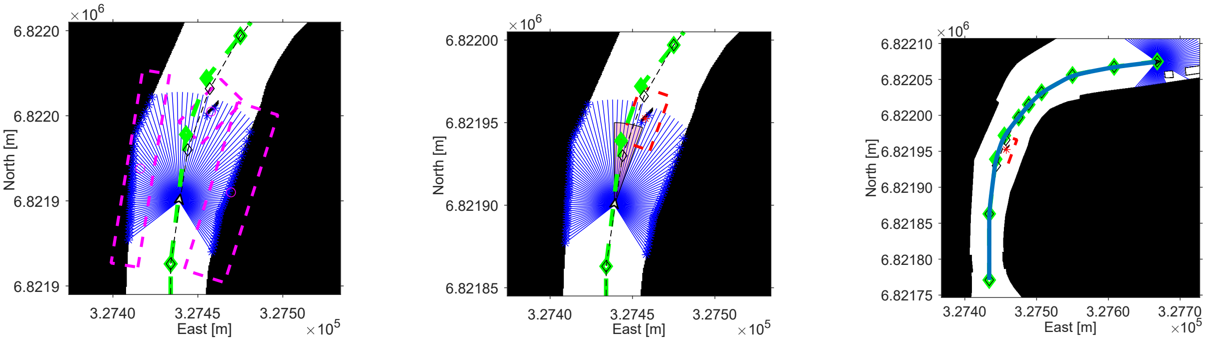
[6] S.-R. Oh, J. Sun, Z. Li, E. A. Celkis, and D. Parsons, "System identification of a model ship using a mechatronic system," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 2, pp. 316–320, 2009.

[7] J. Han, J. Xiong, Y. He, F. Gu, and D. Li, "Nonlinear modeling for a water-jet propulsion usv: An experimental study," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3348–3358, Apr 2017.

[8] D. Mu, G. Wang, Y. Fan, Y. Bai, and Y. Zhao, "Path following for podded propulsion unmanned surface vehicle: Theory, simulation and experiment," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 6, pp. 911–923, Jun 2018.

[9] M. Breivik and T. I. Fossen, "Path following for marine surface vessels," in *Oceans' 04 MTS/IEEE Techno-Ocean'04 (IEEE Cat. No. 04CH37600)*, vol. 4. IEEE, 2004, pp. 2282–2289.

[10] N. Wang and X. Pan, "Path following of autonomous underactuated ships: A translation-rotation cascade control approach," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2583–2593, Dec 2019.



(a) (b) (c)

Fig. 12. Control Scenario I: (a) LiDAR data acquisition and processing for object recognition: all safety boundary box for the LiDAR detected obstacles (dotted purple lines) are created to select the one/s to be avoided. (b) LiDAR data processing for blocking obstacle: the safety boundary box of the blocking obstacle (dotted red line) is created with the slope of the path at the obstacle position. (c) Simulated USV trajectory.

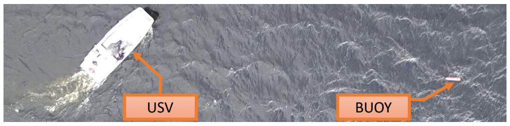


Fig. 13. Control Scenario II: USV during field-test in Tampere, Finland.

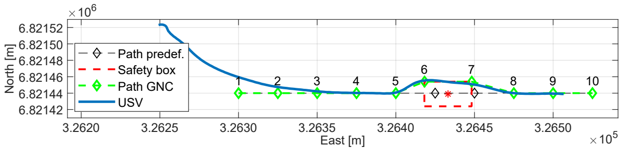


Fig. 14. Control Scenario II: USV path-following with obstacle avoidance of a stationary object. The path generated from the GNC algorithm is marked with a green line, and it uses the left-side corners of the safety boundary box (dotted red line). Waypoints are marked with their respective order number.

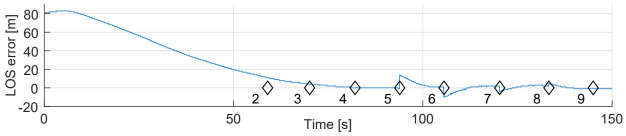
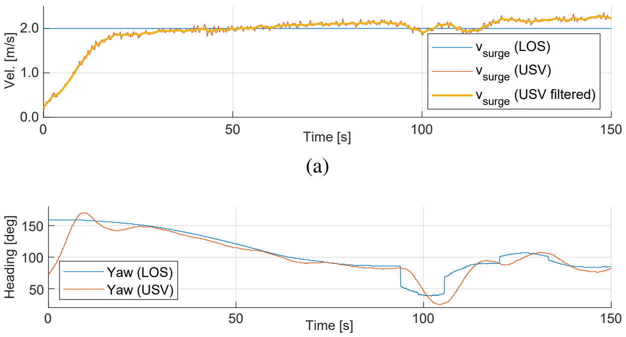


Fig. 15. Control Scenario II: Line-Of-Sight cross-track error $e(t)$ for the lookahead-based steering law defined in (6).



(a)

(b)

Fig. 16. Control Scenario II: (a) Comparison of constant surge velocity of 2 m/s with field-test data from the USV, (b) Comparison of heading angle from the LOS guidance system with field-test data from the USV.

[11] R. Halterman and M. Bruch, “Velodyne hdl-64e lidar for unmanned surface vehicle obstacle detection,” in *Unmanned Systems Technology XII*, ser. SPIE Proceedings, vol. 7692. SPIE, Apr 2010, p. 76920D.

[12] O. Gal and E. Zeitouni, “Tracking objects using phd filter for usv autonomous capabilities,” in *Robotic sailing 2012*. Springer, 2013, pp. 3–12.

[13] *MRS1000: Operating instructions*, SICK AG, Erwin-Sick-Str. 1, 79183, Waldkirch, Germany, Apr 2019, 8020494/12FY/2019-04-02.

[14] E. Simetti, S. Torelli, G. Casalino, and A. Turetta, “Experimental results on obstacle avoidance for high speed unmanned surface vehicles,” in *2014 Oceans-St. John’s*. IEEE, 2014, pp. 1–6.

[15] P. Wu, S. Xie, H. Liu, M. Li, H. Li, Y. Peng, X. Li, and J. Luo, “Autonomous obstacle avoidance of an unmanned surface vehicle based on cooperative manoeuvring,” *Industrial Robot An International Journal*, vol. 44, no. 1, pp. 64–74, Jan 2017.

[16] J. A. Curcio, *Rules of the Road for Unmanned Marine Vehicles*. Springer International Publishing, 2016, ch. chapter 23, pp. 517–526.

[17] Y. Wang, X. Yu, X. Liang, and B. Li, “A colregs-based obstacle avoidance approach for unmanned surface vehicles,” *Ocean Engineering*, vol. 169, pp. 110–124, Dec 2018.

[18] S. Moe and K. Y. Pettersen, “Set-based line-of-sight (los) path following with collision avoidance for underactuated unmanned surface vessel,” in *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, Jun 2016, pp. 402–409.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[20] P. Bouchier, “Embedded ros [ros topics],” *IEEE Robotics & Automation Magazine*, vol. 20, no. 2, pp. 17–19, Jun 2013.

[21] M. Alberri, S. Hegazy, M. Badra, M. Nasr, O. M. Shehata, and E. I. Morgan, “Generic ros-based architecture for heterogeneous multi-autonomous systems development,” in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2018.

[22] M. Aeberhard, T. Kühbeck, B. Seidl, M. Friedl, J. Thomas, and O. Scheickl, “Automated driving with ros at bmw,” *ROSCon 2015 Hamburg, Germany*, 2015.

[23] M. Ollikainen and M. Ollikainen, “The finnish coordinate reference systems,” *Finnish Geodetic Institute and the National Land Survey.*, 2004.

[24] *System Identification Toolbox User’s Guide*, The MathWorks, Inc., Natick, MA, 1988, release 2019a.

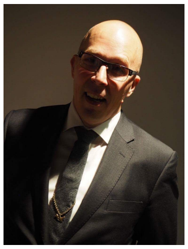
[25] F. A. Papoulias, “Bifurcation analysis of line of sight vehicle guidance using sliding modes,” *Int. J. Bifurcation Chaos*, vol. 1, pp. 849–865, 1991.

[26] J. Villa, J. Aaltonen, and K. T. Koskinen, “Model-based path planning and obstacle avoidance architecture for a twin jet unmanned surface vessel,” in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, Feb 2019, pp. 427–428.

[27] *ROS Toolbox User’s Guide*, The MathWorks, Inc., Natick, MA, 2019, release 2019a.

[28] *CCPilot VC Technical manual*, CrossControl, Mar 2019, product revision: 2.4.

Jose Villa received a B.Sc. and M.Sc. degrees in Industrial Engineering from the University of Zaragoza, Zaragoza, Spain. He is currently a Doctoral student at the Mechatronics Research Group (MRG) at Tampere University (TAU), Tampere, Finland. His research interests include offshore vehicle applications (focusing on Unmanned Surface Vehicles (USVs) and Autonomous Underwater Vehicles (AUVs)), mechanical design, and design and modeling for low-level and advanced control systems.



Jussi Aaltonen (D.Sc Eng.) is the research manager in Mechatronics Research Group (MRG) of Tampere University (TAU) in Finland. He has been working in university research since early 2000's but has also strong industrial background. His research interests are aircraft engineering, autonomous aerial and naval systems and general mechanical, and mechatronics engineering.

Kari T. Koskinen (D.Sc Eng.) is Professor of Fluid Power and responsible professor of Mechanical Engineering study program at Tampere University (TAU) in Finland. He has been working as a professor since 1998. His research interests are in design and analysis of mechatronic systems, aircraft systems, aerial and underwater robotics, lifecycle management of machine and production systems, and future intelligent machines. He is leading Mechatronics Research Group (MRG) at TAU.