Acta Scientific COMPUTER SCIENCES

Volume 4 Issue 6 June 2022

Review Article

Continuous and Discrete-time Models of Surface Watercraft Nonlinear Dynamics

Nikolaos I Xiros[1]* and Eleftherios K Loghis[2]

[1]The university of New Orleans, B. Bollinger School of Naval Architecture and Marine Engineering, New Orleans, Louisiana, United States

[2]National Technical University of Athens, School of Electrical and Computer Engineering, Athens, Greece

***Corresponding Author:** Nikolaos I Xiros, The university of New Orleans, B. Bollinger School of Naval Architecture and Marine Engineering, New Orleans, Louisiana, United States.

Received: December 04,2021

Published: May 13, 2022

Abstract

The scale model of a surface watercraft unit with electric propulsion using a dc motor and waterjet has been developed. A dynamic model capable to adequately describe the motion of the vehicle under a variety of conditions is also developed by combining basic principles with data series obtained through a series of field experiments. The aim is to minimize the number and cost of sensors needed in this end, without unacceptably compromising accuracy, by employing knowledge of vehicle dynamics in order to form a customized gray-box modeling approach. A set of nonlinear differential equations are derived, that can be used to describe the behavior of the marine vehicle at hand. This dynamic model will form the basis for applying physicomimetic approaches to control and navigation of a standalone or swarm of similar vehicles. In the physicomimetic control law synthesis approach, the control problem is tackled by the concept of virtual forces acting on the vehicle and in result generating motion patterns that are desired in a certain application, e.g. avoid obstacles and collisions. To achieve physicomimetic control it is required to effectively cancel the actual dynamics or physics to which each craft unit's motion complies with and then impose the desired dynamics through virtual forces. In the present work, as first step, a series of open loop experiments allow us developing the actual dynamics of vehicle motion.

Keywords: Dynamics; Scale Model; Unmanned Surface Vehicles (USVs)

Introduction

Unmanned Surface Vehicles (USVs) are self-contained unmanned untethered vessels that can transit on the surface of the water autonomously or through remote control. Unlike conventional manned surface vessels that are usually large and costly to build and operate, USVs are typically smaller in size and lower cost resulting from the reduced payload requirement due to being unmanned. In manned vessels, much of the volume and weight is necessary to support the activities (such as control, navigation, maintenance, and mission related tasks), and sustainment (such as berthing, feeding, and entertainment) of the human occupants that recursively increases the size, volume, and power requirements. USVs have no such requirements and therefore are typically many times smaller and more efficient than manned surface vessels.

In the last two decades significant effort has been invested in the development of Unmanned Underwater Vehicles (UUVs), while only a small effort has focused on Unmanned Surface Vessels/Autonomous Surface Vessels (USVs/ASVs). The major efforts in the design of USVs have focused in two areas: platforms for hydrographic data acquisition [1-3], and signal relay platforms that

provide positioning and communications capabilities through the air-sea interface for UUVs [3-5].

The work presented here is part of a larger project that aims to develop a new concept for guidance and control of swarms of marine, or more general, vehicles based on a novel concept termed as robust probabilistic control. In particular, a low-cost solution to tackle the problem is being currently under development that addresses the problem of navigation with minimal information exchange requirements.

In this sense, a multi-purpose Autonomous Surface Vehicle (ASV) that is a low cost mobile surface platform (Figure 1a) has been designed and developed. The system is integrated with a motion measurement package to aid in navigation, control, and to enhance dynamic performance. Control techniques that improve the maneuvering of this type of vehicle are designed, to exploit and expand its operational capabilities [6-13]. This single ASV can also be outfitted with acoustic communication systems to provide position updates and allow underwater vehicles, like in figure 1b or agents to communicate while in transit and surveying. It is also possible to interact with the underwater vehicle to change the mission through an operator communicating with the USV via an RF uplink from shore or a distant vessel [14] or even a hovering aircraft like e.g. the extremely low-cost rotorcraft shown in figure 1 c.

Since more than one units of the ASV in figure 1a or additional underwater or aerial ones (Figures 1b and 1c respectively) need to operate in a concerted manner in order to perform a mission, a possibly heterogeneous swarm guidance and control strategy is needed. Such strategy needs to be distributed, reconfigurable, flexible and fault-tolerant as well as straightforward to implement. This is why we are in train of developing the Robust Probabilistic Control concept, a brief presentation of which is given hereafter in a sea mine detection framework. Then, an outline of the ASV developed is presented along with some preliminary field trial results and associated comments.

Figure 1: (a) The surface vehicle without most of her outfit (top) (b) Unmanned underwater vehicle (bottom left) and (c) Rotorcraft (bottom right).

Problem statement

The majority of tactical scenarios envisioned for performing autonomous underwater cueing for targets or objects (like proud mine-like objects, especially in large littoral areas), involve swarms (flotillas) of unmanned marine vehicles, underwater and possibly surface or even aerial. In specific, the proposed strategy for countermine operations relies on heterogeneous swarms in the sense that they consist of at least two types (tiers) of vehicles as shown in figure 2a: (a) A Tier-1 surface vehicle which can be an greater-scale version of the boat in figure 1a. This type of vehicle will carry radar, hull-mounted sonar, compass, magnetometer(s) and proximity sensors, cameras (infrared or optical), GPS receiver(s), AIS and ECDIS as well as have the capacity to carry and deploy the platform of various sonar types which are unmanned underwater vehicles similar to the one in figure 1b or towed non-self-propelled platforms ("fish"). Further, the Tier-1 vehicles will carry considerable computing and communication assets necessary for the mission. (b) Tier-2 vehicles may be low-cost aerial units like the one shown in figure 1c outfitted with low-cost sensors like camera, GPS and compass as well as appropriate actuation for steering and throttle control.

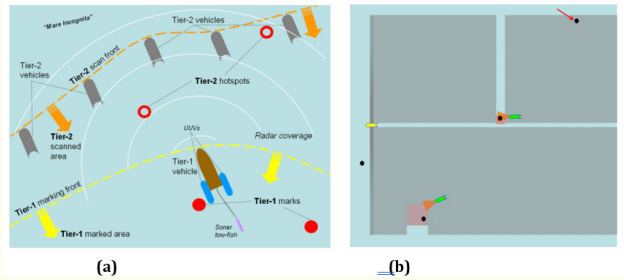


Figure 2: An approach to mine detection and marking (a) 2-tier swarm structure, up and (b) 2D field of operations, down.

Various strategies have been proposed concerning the level of cooperation between the various vehicles as well as diversification of the flotilla in the sense of the number and types of vehicles used. However, up to now there is no general framework to comparatively assess these strategies and schemes. In this end, a unified performance index (or metric) is needed encompassing various quantities, reflecting mission accomplishment as well as the associated economic costs of system acquisition and operation.

Although, economic costs are more or less straightforward to define, mission accomplishment assessment needs in most cases some additional work. On the other hand, a generic performance index that comes to mind when military operations, financial or strategic decisions are of interest is probabilities associated with various events. Returning to the object cueing scenario and limiting our investigation here to a 2D rectangular area (or field) as shown in figure 2b, without significant loss of generality, the following spatial function (map) is introduced for any point (x,y) in our 2D area

f(x,y) = 0 if target is not present at (x,y);

1 if target is present at (x,y).

Furthermore, we assume that f is time-invariant. Therefore, as time goes on any object cueing system covers an increasingly larger portion of the field and assigns 0 or 1 to each one of the points 'visited'; however, for each decision made there is an associated probability of error $p_e(x,y; t)$. It is noted here that probability of error is time dependent because each point in the field may be 'revisited' as many times needed and by any type of vehicles. Therefore, a reasonable setup of the object cueing problem is minimize t_0 for which it holds

$p_e(x,y; t) < p_{e0}$ for $t > t_0$ and all (x,y) in the field.

If the economic costs of system acquisition and operation are included in the minimization problem above, then the overall problem can be approached by multi-objective optimization methods.

Solving the problem stated above requires the determination of various system parameters including but not limited to number of vehicles and vehicle types, vehicle speed, number of sensors and sensor types, level of cooperation, field coverage scheme etc. Evidently, additional constraints will be introduced to the problem due to the physics and engineering involved as e.g. vehicle speed ceiling, coverage scheme limitations due to maneuverability limitations etc.

A simple model for taking into account the main features of a possible configuration is shown in figure 2b. In this simplified version, there are two types of vehicles regardless of their implementation: Search vehicles (SV, shown in yellow) and Classification vehicles (CV, shown in green). Also, search sensors relying on the echoing principle (e.g. sonar) are considered.

SVs are equipped with non-classifying, long-range, side-scanning sensors. They provide with field scans of resolution decreasing with respect to the distance from the vehicle. As shown in figure 2b, the area marked as a result of an echo signal is increasing with the round trip time of the signal. Furthermore, this area is in any case larger than the area covered by the target itself as there are a number of uncertainties involved, including the positioning uncertainty of the SV. By assuming a 'line-of-sight' (LOS) mode of operation for the sensors, any target detected causes obstruction of view; therefore, the area 'behind' a detected target cannot be scanned unless the SV approaches the obstructed area from another angle or during another pass.

The probability of error for any point inside the region scanned by a specific SV, v, up to time t is given by the following relationship.

$$p_e(x,y; v@t) = f(x,y) \times p_{miss}(d<(x,y),traj\{v@t\}>) +$$

$$+ [1 - f(x,y)] \times \sum \{ [1 - p_{miss}(d<(x_k,y_k),traj\{v;t\}>)] \times$$

$$\times p_{1|0}(x,y; x_k,y_k; traj\{v@t\}) \}$$

In the above, d<(x,y),traj{v@t}> stands for the Euclidean distance of a point (x,y) from the trajectory, traj{v@t}, of the specific SV, v, up to time t; if not defined is set to Inf.

Distribution $p_{miss}(d<(x,y),traj\{v@t\}>)$ refers to the probability density function to miss a target in the manner indicated by the red arrow in figure 2, provided that point (x,y) lies in the area encompassed by a target; $p_{miss}(d<(x,y),traj\{v@t\}>)$ is expected to be an increasing function of d<(x,y),traj{v@t}> and actually to saturate to unity if this distance lies above a threshold. Also, p_{miss} depends on the capabilities of the sensors mounted on the SV.

Distribution $p_{1|0}$ stands for the probability density function that a point (x,y) belongs to the marking area assigned to the k-th target, if detected by SV v until time t. It is reasonable to assume that $p_{1|0}$ is a decreasing function of the distance of point (x,y) from the center location of the k-th target (the larger the distance between the two points the less probable to be included in the same 'patch') and an increasing function of $d<(x_k,y_k),traj\{v;t\}>$ because of the 'spreading' caused by reduced sensor resolution in the far field.

$$p_{1|0}(x,y; x_k,y_k; traj\{v@t\}) =$$

$$= p_{1|0}(d<(x,y),(x_k,y_k)>, d<(x_k,y_k),traj\{v;t\}>)$$

In the above, $d<(x_1,y_1),(x_2,y_2)>$ stands for the Euclidean distance between two points on the plane. It would be

The summation in the relationship for $p_e(x,y; v@t)$ spans all the targets in the field of interest. If (x,y) does not lie in the region already scanned by the specific SV, v, until time t, it is easy to verify that $p_e(x,y; v@t) = 1$ if $f(x,y) = 1$ and $p_e(x,y; v@t) = 0$ if $f(x,y) = 0$ and no target in some vicinity of (x,y) has been scanned yet. Although, it could have been defined differently, these facts make intuitive sense.

The relationship for $p_e(x,y; v@t)$ is valid provided some underlying assumptions hold. The targets must be distributed reasonably sparse so that the patches marked for two or more neighboring objects by the SVs do not overlap. Furthermore, the probability of 'spontaneous triggering' of the sensors, in the sense that they detect a target at a point where none is present in the vicinity, is considered negligible. Although these assumptions simplify the expression for $p_e(x,y; v@t)$, they can be dropped, if needed, by appropriate extensions.

A way to tackle such requirements expressed in a probabilistic framework is to consider techniques Robust Stochastic Control. Indeed, stochastic control deals with systems where the objective is to minimize or maximize the probability of an event related to a dynamic system or random signal by manipulating one or more of a driving signal. In conventional stochastic control, e.g. [15,16], the full dynamics of the process or system under control is considered. Given recent advances in self-organization, however, an alternative path could be followed. Specifically, the swarm's vehicles will be driven by local, decentralized and in effect distributed deterministic control laws that require the knowledge by each vehicle of its own motion or other variables as well as those of the immediately neighboring units, but in no case the state variables of the entire formation. This scheme is usually implemented as a set of "virtual forces" exerted between neighboring vehicles, resembling the forces arising between the molecules or particles of a solid, liquid or gaseous substance; thereof the term physicomimetics or artificial physics. A critical step needed towards the direction of applying this technique, which is referred to as Robust Probabilistic Control is the migration from the concept of trajectories employed to describe the time evolution of the phase (state) space of a dynamical system to that of the phase space probability distribution function (or measure), ρ(q,p;t) of the dynamical system. Consider a dynamical system with canonical coordinates q_i and conjugate momenta p_i, where i = 1,…,n. Then the phase space distribution ρ(q,p) determines the probability $\rho(q,p)d^nqd^np$ that the system will be found in the infinitesimal phase space volume d^nqd^np. The concept and its applications will be investigated in full detail in a follow-up work.

Probabilistic measures of mission success like the ones introduced earlier can in turn be linked to individual vehicle dynamics through Liouville's Theorem. This theoretical tool depicts the dynamics of the collective behavior of an autonomous swarm on the basis of the phase-space probability distribution measure concept, instead of some more conventional approach based on individual vehicle trajectories. Specifically, Liouville's theorem governs the evolution of the distribution function ρ(q,p;t) in time according to the following partial differential equation:

$$\frac{d\rho}{dt} = \frac{\partial\rho}{\partial t} + \sum_{i=1}^{n}\left(\frac{\partial\rho}{\partial q_i}\dot{q}_i + \frac{\partial\rho}{\partial p_i}\dot{p}_i\right) = 0$$

As a direct consequence of Liouville's theorem, the following equation can be obtained for the distribution function of a swarm's phase space.

$$\frac{\partial\rho}{\partial t} = [H,\rho]$$

Where H is the Hamiltonian and the Poisson bracket between two functions of the canonical coordinates is defined by the following.

$$[f,g] = \sum_{i=1}^{n}\left(\frac{\partial f}{\partial q_i}\cdot\frac{\partial g}{\partial p_i} - \frac{\partial f}{\partial p_i}\cdot\frac{\partial g}{\partial q_i}\right)$$

The Newtonian equation of motion for each swarm vehicle will, in effect, assume the following form.

$$\vec{\mathbf{a}}_i = \frac{1}{m_{i,\text{virtual}}} \cdot \sum_{\substack{j \\ i \neq j}} \vec{\mathbf{F}}_{ij,\text{virtual}}$$

In the above, a_i is the acceleration of the i-th vehicle, $m_{i,virtual}$ the virtual mass assigned to the vehicle by the artificial physics, and $F_{ij,virtual}$ the virtual force exerted on the i-th vehicle by the j-th neighbor per the physicomimetic interaction law governing the swarm dynamics.

The challenges related with swarm vehicle navigation and guidance are complex and can only be tackled using a comprehensive and non-trivial framework for the design of the local-loop controller that can guarantee compliance of the vehicle to artificial physics dynamics. The actual equation of motion of an arbitrary vehicle, assuming a material point model, is as follows.

$$\vec{\mathbf{a}}_i = \frac{1}{m_i} \cdot \left[\sum \vec{\mathbf{F}}_{\text{exogenous}} + \sum \vec{\mathbf{F}}_{\text{control}} \right]$$

In the above, m_i is the actual mass of the i-th vehicle, $\Sigma F_{exogenous}$ the resultant exogenous force exerted on the i-th vehicle by the environment and $\Sigma F_{control}$ the resultant control force exerted on the i-th vehicle by thrusters, control surfaces etc. To make vehicle dynamics comply to artificial physics, as defined in the previous task, the resultant control force has to be as follows.

$$\sum \vec{\mathbf{F}}_{\text{control}} = \frac{m_i}{m_{i,\text{virtual}}} \cdot \sum_{\substack{j \\ i \neq j}} \vec{\mathbf{F}}_{ij,\text{virtual}} - \sum \vec{\mathbf{F}}_{\text{exogenous}}$$

The above clearly shows the issues that may arise if the physicomimetic control law generates excessively high forces or the virtual mass value is small compared to the actual mass value of the vehicle. This can lead to excessive control force requirements that can make the artificial physics scheme infeasible. This is why the local-loop controller design is a non-trivial aspect and needs to be investigated at least for one type or class of practical vehicles.

As in other similar cases, two or more subsystems either linear or nonlinear and either lumped or distributed, are coupled through processes which commonly are highly nonlinear. In our case an appropriate identification method is employed, as explained later on, which comes as generalization of the identification methodology for linear systems in phase-plane. Since the physicomimetic governing control laws may need to be continuously reconfigured, the option to adapt accordingly the gains of the local-loop controllers will be investigated in order to implement them in a follow-up effort. Such approach will further increase system responsiveness and adaptability. Such adaptability enables for broad real-time reconfiguration of an autonomous swarm which in turn could convert routinely multi-sortie missions, due to the need of off-line reconfiguration, to single-sortie ones not needing real-time oversight. The above work clearly demonstrates the need of vehicle dynamics identification for local-loop controller tuning and adaptation. The system identification and state-space model derived is given in the section following after vehicle description.

vehicle development

The ultimate objective of this project is to develop an automated guidance system for ferryboats as well as landing craft. A landing craft scale model with electric propulsion outfit and running on two identical waterjets is shown in figure 1. The guidance algorithm to be implemented will be based on the Robust Probabilistic Control framework presented in the previous section. In this end, the following steps are envisioned: a) Development of hardware and software for field tests, data acquisition and algorithm validation; b) Data acquisition campaign, system identification and vehicle modeling; c) Development of a control algorithm with validated behavior in autopilot tasks; and d) To enhance the autopilot function of the shipboard controller with automated guidance capability based on distance sensing and other sensory instrumentation as required in the Robust Probabilistic Control framework.

Landing craft scale model

The landing craft to be used is using two water-cooled Permanent-Magnet DC motors for propulsion. Motor speed and torque output is controlled through Pulse-Width-Modulation converter.

Two waterjets are employed for propulsion. Power supply is through two battery packs connected in parallel, with nominal output voltage of 13.2V. Each set includes 11 Ni-Cd batteries of 3000mAh each, connected in series to achieve the required voltage output.

main processing development board

An in-house, custom-made microprocessor board has been developed to be used as the hardware platform for the development

of the Robust Probabilistic Control scheme implementing the automated guidance functionality. The actual board developed is shown in figure 3, along with the architecture layout.

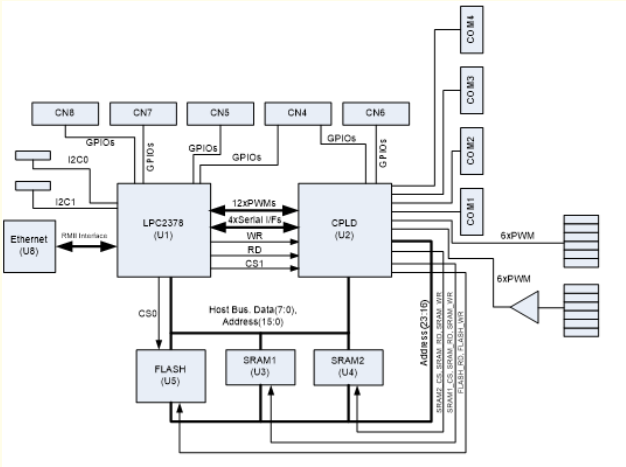


Figure 3: The main microprocessor board (top) and its architecture (bottom).

The system software implemented includes drivers for a variety of peripheral devices including a GPS receiver with active antenna, magnetometer-type compass, 6 DOF Inertial Measurement Unit used as tri-axial accelerometer and motion sensor, two optical sensors used as rpm counters and a PC wireless interface for command and data transmission in real time. To integrate all these devices the system software employs hardware features on the development board like the four serial interfaces, the six PWM output ports, the six PWM input ports, 1MB SRAM and 8MB flash memory that have been intentionally included in the design.

Another important system software function is the system monitor that records the current status of the boat. Status updates are generated several times per sec and at different sampling rates for each sensor. The data collected by the system monitor function is stored in real time at the SRAM or the flash memory module to be retrieved later by a computer.

For human-machine-interface (HMI) purposes a Command Line Interface (CLI) has been developed with a rich command set. The CLI can be extended at a later to time to a GUI (Graphical User Interface) running in real-time at the remote control workstation. The currently available CLI allows commanding the vehicle and performing field tests. Debugging functionality separate for each individual sensor has been included in the CLI.

The entire system software platform has been based on the FreeRTOS real-time operating system.

auxiliary hardware

The development board is complemented by a CPLD (Complex Programmable Logic Device) intended to provide with fast I/O (input-output) and interfacing capabilities.

The following functional modules have been incorporated on the CPLD: a) Parallel interface to communicate with the microcontroller unit; b) Two RPM counter modules; c) Two PWM reader modules used to check for consistency and record the propulsion motor PWM settings received from the remote control workstation over wireless; d) Auxiliary circuitry to download code to the microprocessor of the development board; e) Output multiplexers allowing running the boat on manual remote control or automatic mode; f) Watchdog to monitor proper execution of the microcontroller program, by resetting it in case of timeout; g) Reset signal generator for the development board microcontroller.

The functional modules above have been programmed onboard the CPLD as VHDL hardware description language modules. The VHDL code extends in 16 files.

system identification of craft dynamics

dynamic tests and experimentation

A set of preliminary experimental tests have been performed on the watercraft. These tests had a dual purpose: to calibrate and fine-tune the vehicle systems and to build a data set to be used in upcoming modeling and system identification activities.

A first subset of tests was performed on a specifically built proving tank and is described in detail in a previous work [17].

Then, a subsequent test subset was performed ashore with all systems integrated on the vehicle to functionally check and verify, fine tune and calibrate them.

Finally, a data collection subset of tests was performed at an open water basin. A representative chunk of the data series ob-

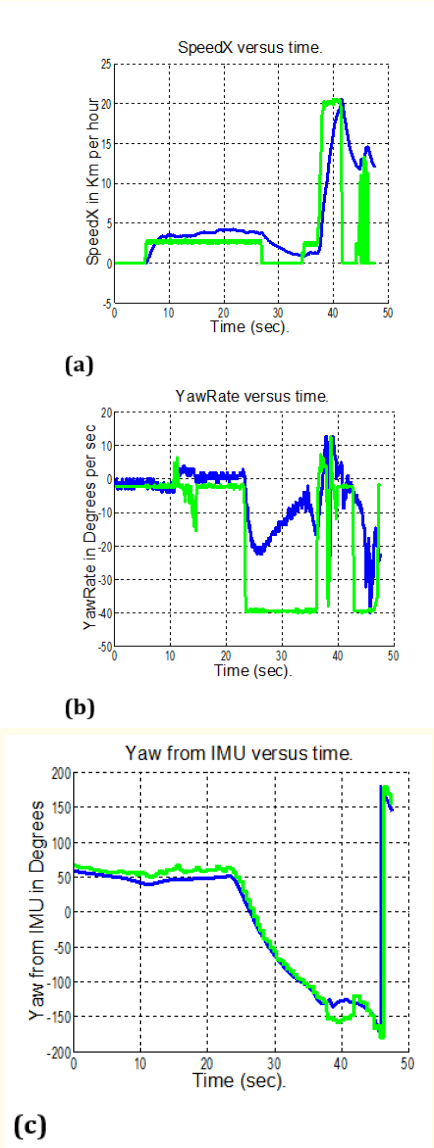
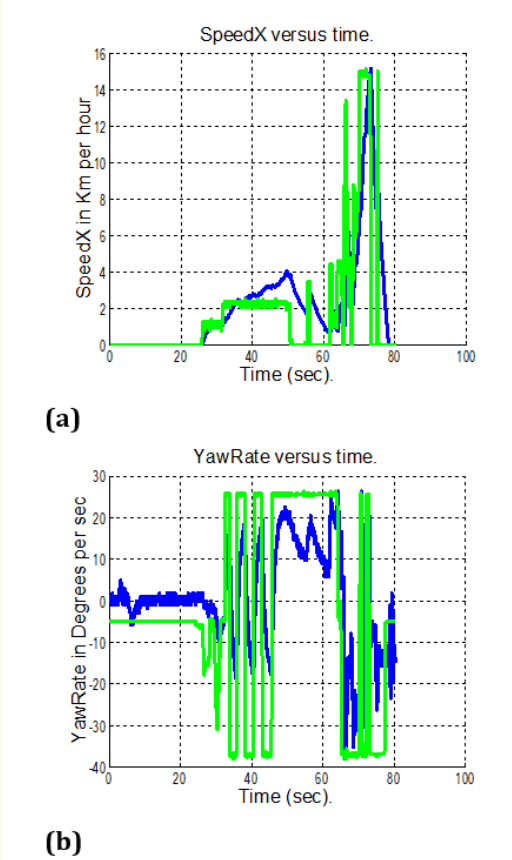


Figure 4: (a) Speed at X-axis versus time. The plot embeds in green the thrust handle values normalized to speed maximum (in order to reveal the time dependence) (b) Yaw rate versus time (blue curve). The plot embeds the steering handle values (green curve) normalized to yaw rate maximum, to show the time dependence. Negative values both at yaw rate and steering imply right turn. (c) Yaw angle resulting from integration of yaw rate sensor data versus time (blue curve). The green curve is the angle reported by the magnetometer. All angles are normalized to [-180°, 180°].

tained during these tests is shown in figure 4. For the experiment in figure 4, the boat performs a straight-line run (incl. acceleration) then a U-turn and finally a straight-line run in the opposite direction. Figure 4a shows the speed on the x-axis versus time. The plot embeds in green the thrust handle values normalized to speed maximum in order to reveal the time dependence. Figure 4b shows the yaw rate versus time (blue curve). The plot embeds the steering handle values (green curve) normalized to yaw rate maximum, to show the time dependence. Negative values both at yaw rate and steering imply right turn. Finally, figure 4c shows the yaw angle resulting from integration of yaw rate sensor data versus time (blue curve). The green curve is the angle reported by the magnetometer. All angles are normalized to [-180°, 180°]. Similar data series are presented in figure 5 for a zigzag maneuvering test.



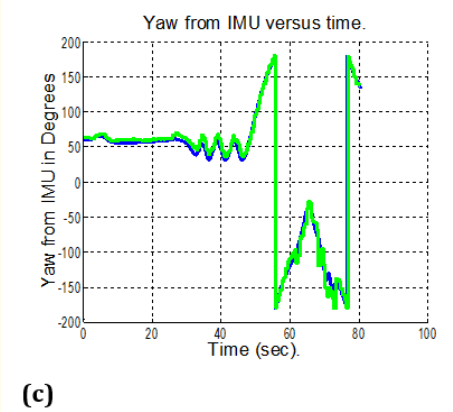


Figure 5: (a) Speed at X-axis versus time. The plot embeds in green the thrust handle values normalized to speed maximum (in order to reveal the time dependence) (b) Yaw rate versus time (blue curve). The plot embeds the steering handle values (green curve) normalized to yaw rate maximum, to show the time dependence. Negative values both at yaw rate and steering imply right turn. (c) Yaw angle resulting from integration of yaw rate sensor data versus time (blue curve). The green curve is the angle reported by the magnetometer. All angles are normalized to [-180°, 180°].

State equations for craft dynamics

Based on the measurement data series, we concluded that the craft's maneuvering dynamics can be sufficiently described by a 2nd-order lumped dynamical model with two state variables: forward velocity (speed of advance) and yaw rate (angular velocity). Using the same data series, two types of state-space models are derived to describe the vehicle: (a) a continuous-time one consisting of two 1st-order ordinary differential equations one for forward acceleration and one for yaw acceleration; (b) a discrete-time NARMA (Nonlinear Autoregressive Moving Average) one consisting of two difference equations, one for each state variable. The implementation of model (a) in Matlab/Simulink© is shown in figure 7 while that of model (b) in figure 8. The results in terms of forward velocity and yaw rate obtained by some initial runs of the continuous-time state-space model are shown in figure 8 while those of the discrete-time one in figure 9. Note that in the same plot they are validated against the experimental data series obtained by measurement. The equations defining both models are given in the next sections.

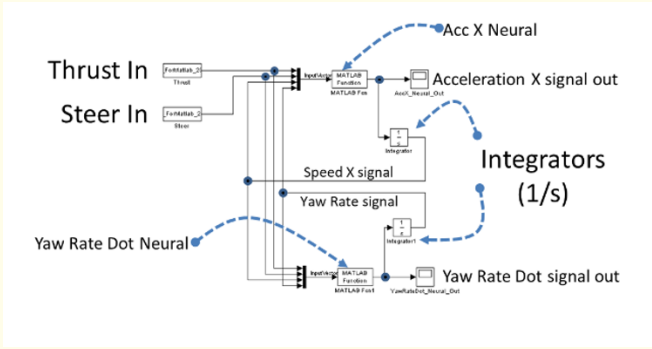


Figure 6: Continuous-time state equations for craft dynamics in Matlab/Simulink©.

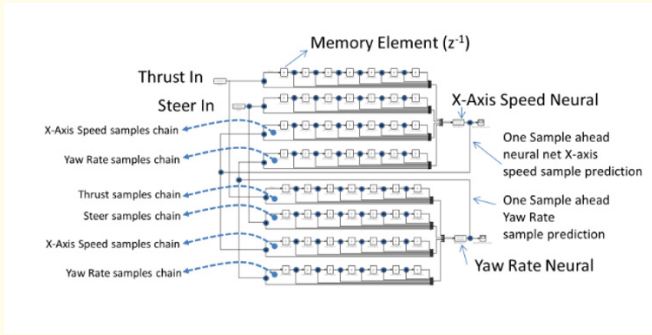


Figure 7: Discrete-time state equations for craft dynamics in Matlab/Simulink©.

State equations model in continuous time

In this setup, the model obtains the standard form of first-order nonlinear state equations as follows:

$$\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}}{dt} = \mathbf{f}_c(\mathbf{x}(t); \xi_{\#}(t)), \dot{\mathbf{x}}(t) = \begin{bmatrix} a_t(t) & a_a(t) \end{bmatrix}^{\mathrm{T}}$$

Evidently the state vector x includes the translational (rectilinear) velocity and the yaw rate (angular speed) of the vessel, while its time derivative includes translational acceleration a_t and angular acceleration a_a. Vector $\xi_{\#}$ packs the control inputs to the vessel that include the thrust throttle setting y_T and the steering command δ_R or their normalized counterparts as defined below:

$$y_{\%}(t)=\frac{y_T(t)}{256},\delta_{\%}(t)=\frac{\delta_R(t)-121}{128}$$

Note that normalized throttle lies between zero and one while normalized steer lies within –1 (full stroke starboard, i.e. clockwise) and +1 (full stroke port, i.e. counterclockwise).

The somewhat abstract function f_c is defined as a vector function; each one of its components is a neural net yielding a scalar output and accepting four scalar inputs packed in vector ξ_1 as follows:

$$\hat{\mathbf{i}}_0=\mathbf{f}_{\mathbf{C}}(\hat{\mathbf{i}}_1)=\left[f_{<1>}(\hat{\mathbf{i}}_1)\quad f_{<2>}(\hat{\mathbf{i}}_1)\right]^{\mathrm{T}},$$

$$\hat{\mathbf{i}}_0=\left[\frac{a_t(t)}{3.4\mathrm{ms}^{-2}}\quad\frac{a_a(t)}{130^{\mathrm{O}}\mathrm{s}^{-2}}\right]^{\mathrm{T}},$$

$$\hat{\mathbf{i}}_1=\left[y_{\%}(t)\quad\delta_{\%}(t)\quad\frac{u(t)}{30\frac{\mathrm{km}}{\mathrm{h}}}\quad\frac{\omega(t)}{50^{\mathrm{O}}\mathrm{s}^{-1}}\right]^{\mathrm{T}}$$

Each neural net has three layers (one hidden, one output and one input). The hidden layer of both nets has nine neurons (nodes) since 9 = 2 X 4 +1. Therefore each net obtains the generic form below. The numeric values of the weight matrices and vectors as well as biases are omitted here for the sake of compactness but will be included in full in a future more extensive work.

$$f_{<1>}(\hat{\mathbf{i}}_1)=\mathbf{w}_{0,<1>}\cdot\ddot{\mathbf{O}}_{<1>}\left(\mathbf{W}_{2,<1>}\hat{\mathbf{i}}_1+\hat{\mathbf{i}}_{2\mathrm{B},<1>}\right)+\xi_{0\mathrm{B},<1>}$$

$$f_{<2>}(\hat{\mathbf{i}}_1)=\mathbf{w}_{0,<2>}\cdot\ddot{\mathbf{O}}_{<2>}\left(\mathbf{W}_{2,<2>}\hat{\mathbf{i}}_1+\hat{\mathbf{i}}_{2\mathrm{B},<2>}\right)+\xi_{0\mathrm{B},<2>}$$

$$\ddot{\mathbf{O}}_{<1,2>}(\div)=\ddot{\mathbf{O}}_{<1,2>}\left(\begin{bmatrix}\chi_1\\\vdots\\\chi_9\end{bmatrix}\right)=\begin{bmatrix}\Phi_{<1,2>}(\chi_1)\\\vdots\\\Phi_{<1,2>}(\chi_9)\end{bmatrix}$$

$$\Phi_{<1>}(\chi)=\Phi_{<2>}(\chi)=\tanh(\chi)=\frac{2}{1+e^{-2\chi}}-1$$

Difference equations model (discrete time)

In this setup, the model obtains the standard form of nonlinear state equations for the state signal vector sample one time step ahead, as follows:

$$\hat{\mathbf{x}}(n+1)=\mathbf{f}_d\left(\mathbf{x}\begin{pmatrix}n,\\n-1,\\\ldots,\\n-7\end{pmatrix};\ {}^{\#}\begin{pmatrix}n,\\n-1,\\\ldots,\\n-7\end{pmatrix}\right)$$

Note that a time window (memory) of seven past samples for the state and control input vectors, which are defined as previously, has been used. The time step Δt is 0.9 s and should not be confused with the sampling period T_S = 30 ms = Δt/30 used for data series acquisition. All time instants are identified as integer multiples of time step Δt in this model.

The somewhat abstract function f_d is defined as a vector function; each one of its components is a neural net yielding a scalar output and accepting 32 scalar inputs packed in vector ξ_1 as follows:

$$\hat{\mathbf{i}}_0=\mathbf{f}_{\mathbf{D}}(\hat{\mathbf{i}}_1)=\left[f_{<1>}(\hat{\mathbf{i}}_1)\quad f_{<2>}(\hat{\mathbf{i}}_1)\right]^{\mathrm{T}},$$

$$\hat{\mathbf{i}}_0=\left[\frac{u(n+1)}{30\frac{\mathrm{km}}{\mathrm{h}}}\quad\frac{\omega(n+1)}{50^{\mathrm{O}}\mathrm{s}^{-1}}\right]^{\mathrm{T}},$$

$$\hat{\mathbf{i}}_1=\begin{bmatrix}\hat{\mathbf{i}}_y\\\hat{\mathbf{i}}_\delta\\\hat{\mathbf{i}}_u\\\hat{\mathbf{i}}_\omega\end{bmatrix},\quad\hat{\mathbf{i}}_y=\left[y_{\%}(n-7)\quad\cdots\quad y_{\%}(n)\right]^{\mathrm{T}}$$

$$\hat{\mathbf{i}}_\delta=\left[\delta_{\%}(n-7)\quad\cdots\quad\delta_{\%}(n)\right]^{\mathrm{T}}$$

$$\hat{\mathbf{i}}_u=\frac{\left[u(n-7)\quad\cdots\quad u(n)\right]^{\mathrm{T}}}{30\frac{\mathrm{km}}{\mathrm{h}}}$$

$$\hat{\mathbf{i}}_\omega=\frac{\left[\omega(n-7)\quad\cdots\quad\omega(n)\right]^{\mathrm{T}}}{50^{\mathrm{O}}\mathrm{s}^{-1}}$$

Like previously, each neural net has three layers. However, both nets are now remarkably larger since they accept more inputs. The hidden layer of both nets has now 65 = 2 X 32 +1 neurons. Therefore each net obtains the generic form below. The numeric values of the weight matrices and vectors as well as biases will be included in full in a future more extensive work.

$$f_{<1>}(\hat{\mathbf{i}}_1)=\mathbf{w}_{0,<1>}\cdot\ddot{\mathbf{O}}_{<1>}\left(\mathbf{W}_{2,<1>}\hat{\mathbf{i}}_1+\hat{\mathbf{i}}_{2\mathrm{B},<1>}\right)+\xi_{0\mathrm{B},<1>}$$

$$f_{<2>}(\hat{\mathbf{i}}_1)=\mathbf{w}_{0,<2>}\cdot\ddot{\mathbf{O}}_{<2>}\left(\mathbf{W}_{2,<2>}\hat{\mathbf{i}}_1+\hat{\mathbf{i}}_{2\mathrm{B},<2>}\right)+\xi_{0\mathrm{B},<2>}$$

$$\ddot{\mathbf{O}}_{<1,2>}(\div)=\ddot{\mathbf{O}}_{<1,2>}\left(\begin{bmatrix}\chi_1\\\vdots\\\chi_{65}\end{bmatrix}\right)=\begin{bmatrix}\Phi_{<1,2>}(\chi_1)\\\vdots\\\Phi_{<1,2>}(\chi_{65})\end{bmatrix}$$

$$\Phi_{<1>}(\chi)=\left(1+\exp(-\chi)\right)^{-1},\Phi_{<2>}(\chi)=\tanh(\chi)$$

Signal analysis

For both the continuous and the discrete model, signal analysis and processing was required in order to be able to use the experi-

mental data series obtained by measurement for training of the neural nets appearing in the model core.

For the continuous-time model it was required to generate a smooth yaw acceleration signal since the instrumentation did not provide such output directly. The signal pre-processing used to obtain that is shown in figure 8.

Figure 8: Signal processing to generate smooth yaw acceleration signal for the neural nets in the continuous-time state model.

For the discrete-time model we needed to keep the number of inputs to the core neural nets to a minimum. In this end, decimation of the recoded data series to the time scales occurring in the system was required. Note that, following standard engineering practice, the sampling rate of the measurements was rather high to ensure fidelity. Data series decimation was performed by propagating the recorded data signals through the filtering architecture shown in figure 9.

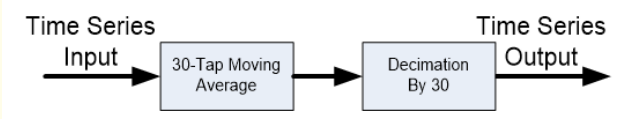


Figure 9: Decimation filtering applied to minimize number of inputs of the neural nets in the discrete-time state model.

Simulation results and model assessment

Based on the field trials for the landing craft scale model in figure 1 two models were developed with fundamentally different features. The continuous time (CT) model which is the most common in system identification and control engineering literature could not be readily constructed due to a major technical weakness of the low-cost sensor instrumentation employed. Specifically, the lack of data for yaw acceleration directly obtained through measurement. Indeed, the IMU employed was providing with translational acceleration data series for each run but not with yaw angular acceleration data.

To overcome this problem a two-fold approach was employed. An elaborate filtering scheme along with detailed preprocessing of the yaw rate angular speed much more complicated than what applied for the discrete time (DT model. Also, a DT model was developed to cross-check the results of the CT model.

As can be seen in figure 10, the results obtained from the DT model for both forward speed and yaw rate of the vessel were very close to the measurements. Even though there is no integration, numerical or otherwise, involved the major parameters that needed to be identified for the DT model was the memory (time window size) for each scalar state variable and control input as well as the most suitable decimation factor for the high-rate sampled measurement series. As was demonstrated in practice, a suitable decimation factor choice led to significantly reduced care for the memory size. This combined with the fact that neural nets can perform well even with partial or redundant data series allowed to use the same memory size for all states and control inputs with very good results as observed in figure 10.

In contrast, even though much more effort was invested in the development of the CT model, the outcome was significantly poorer as seen in figure 11 for both forward speed and especially yaw rate since no measured data for yaw acceleration were available from the IMU used. To make comparisons easier the same run was tested for both the CT and DT model.

One way to improve the fidelity of the CT model is to employ the DT model to generate data series which can be used in turn to train the neural nets of the CT model especially the one for yaw acceleration. In this scenario, an appropriate data fusion scheme is needed to compromise the smoothening filtering employed for the speed signals and the corresponding outputs of the DT model for which acceleration is not used or needed to develop. Such a data fusion scheme would generate speed signals closer to the measurements but smooth enough to be differentiated with respect to time yielding improved accelerations.

Evidently, another way is to use a different IMU that directly outputs yaw acceleration and not just yaw rate. In this case the DT

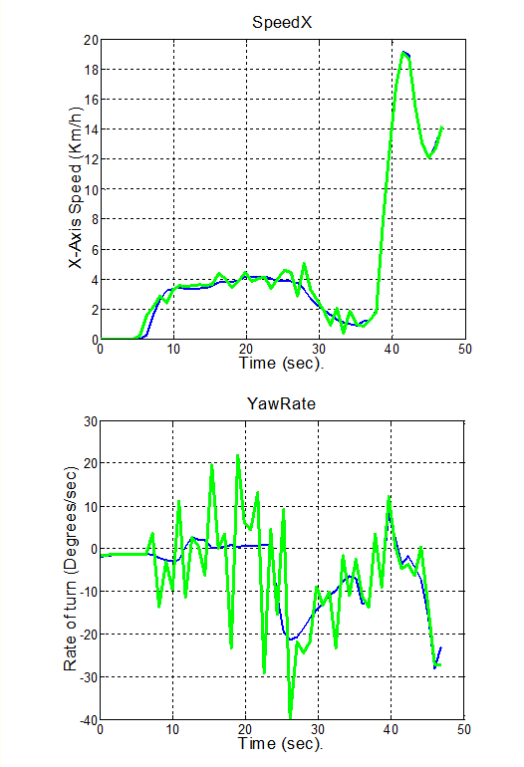


Figure 10: Discrete-time state-space model output (green) vs. measurements (blue): forward velocity (up) and yaw rate (down).

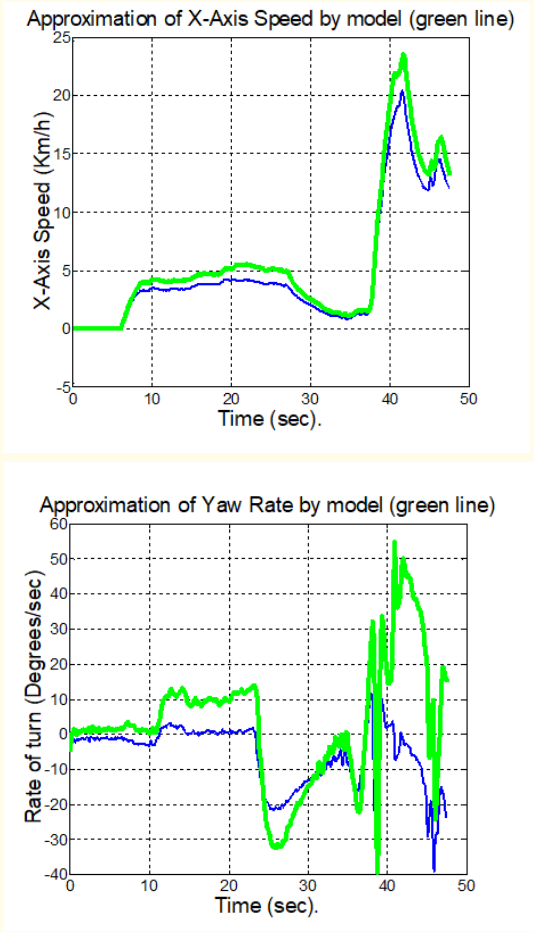


Figure 11: Continuous-time state-space model output (green) vs. measurements (blue): forward velocity (left) and yaw rate (right).

model can still be of value because it can be employed in a signal processing scheme to remove the noise of from the acceleration measurements; indeed, as can be seen even from the forward speed and acceleration signals there is significant disagreement between the two which is attributed to sensor noise and distortion.

In any case, either the CT or the DT model can be employed in the framework of Robust Probabilistic Control outlined earlier. Since this framework is one of Stochastic Control, therefore intrinsically designed to work with uncertain signals, and works with state distribution functions rather than exact trajectories any modeling errors can be incorporated in a probabilistic model similar to the one outlined.

Conclusions

In this work, the problem statement of automated guidance of an autonomous marine vehicle is formulated in a Robust Probabilistic Control framework. Then, the developments toward a landing craft scale model with appropriate control hardware and system software is presented. The boat will be used to implement Robust Probabilistic Control concepts in automated guidance. The next steps in this direction require some more work on fine tuning of the onboard instrumentation, control algorithm design and controller development and finally validation of the control system through extensive simulation and field tests.

The feedback control law to be investigated includes robust control concepts implemented by numerical, computational techniques involving artificial intelligence and neural networks. Additionally, a supervisory controller implementing a task, e.g. homing to a focal point from arbitrary initial position away and arbitrary initial velocity. For both the feedback control law and the supervisory controller the continuous-time and the discrete time model of the craft will be valuable and used during both the design and development phase as well as for system validation and verification. For example, the continuous-time model can be used to derive linearized local models that can be in turn used to design and tune the feedback control law; the discrete-time model can be used to verify the supervisory control scheme as well as the overall control system. The results will be presented as part of future works.

Acknowledgments

The authors would like to thank Dr. Tryfon Koussiouris who served as Professor of Electrical and Computer Engineering at the National Technical University of Athens and Mr. Dimitris Loghis for their valuable help and ongoing support to this project. Nik. Xiros would also like to express his gratitude to the National Science Foundation (NSF) and, in particular, the Energy, Power, Control and Networks (EPCN) program for their continued support of this research under grant ECCS-1809182, titled "Collaborative Research: Design and Control of Networked Offshore Hydrokinetic Power-Plants with Energy Storage".

Bibliography

1. M. Chaumet-Lagrange., *et al*. "Design of an Autonomous Surface Vehicle (ASV)". University of Bordeaux I, France, Automatic and production Laboratory, IEEE (1994).

2. JE Manley. "Development of the Autonomous Surface Craft 'ACES'". Massachusetts Institute of Technology , Department of Ocean Engineering, Sea Grant College Program, Cambridge MA 02139, IEEE (1997).

3. "CARAVELA Development of a Long-Range Autonomous Oceanographic Vessel". Dynamic Systems and Ocean Robotics lab (DSOR), 1998-2000.

4. P Oliveira., *et al*. "The DELFIM Autonomous Surface Craft". Report December (1999).

5. P Oliveira., *et al*. "A Nonlinear Vision Based Tracking System for Coordinated Control of Marine Vehicles". IST/DEEC, Lisbon, Portugal, Naval Postgraduate School, Monterey USA (2002).

6. Wang L., *et al*. "Design and Comparison of H∞/H2 Controllers for Frigate Rudder Roll Stabilization". *Journal of Marine Science and Application* 18 (2019): 492-509.

7. Xiros NI., *et al*. "Modeling and Simulation of Planing-Hull Watercraft Outfitted with an Electric Motor Drive and a Surface-Piercing Propeller". *Journal of Marine Science and Engineering* 7.2 (2019): 49.

8. Taunton DJ., *et al*. "Characteristics of a Series of High Speed Hard Chine Planing Hulls -Part II: Performance in Waves". *International Journal of Small Craft Technology* (2011): 153.

9. Claus D Simonsen., *et al*. "Maneuvering predictions in the early design phase using CFD generated PMM data". 29th Symposium on Naval Hydrodynamics, Gothenburg, Sweden, 26-31 August (2012).

10. Toru Katayama., *et al*. "Development of Maneuvering Simulation Method for High Speed Craft Using Hydrodynamic Forces Obtained from Model Tests". 10th International Conference on Fast Sea Transportation, FAST 2009, Athens, Greece, October (2009).

11. Hajizadeh Sajad., *et al*. "Evaluation of planing craft maneuverability using mathematical modeling". *Scientia Iranica* 67 (2016): 85-100.

12. Tavakoli Sasan and Dashtimanesh Abbas. "A six-DOF theoretical model for steady turning maneuver of a planing hull". *Ocean Engineering* 189 (2019): 106328.

13. Sung Li and Fabien. "Control Oriented Maneuver Model of Prismatic Planing Hull". *Brian* (2021).

14. A Leonessa., *et al*. "Development of a small, multi-purpose, autonomous surface vessel". Florida Atlantic University, Department of Ocean Engineering (2002).

15. T Fossen. "Guidance and Control of Ocean Vehicles". Baffins Lane, England: John Wiley and Sons, Ins, (1994).

16. Xiros NI., *et al*. "An Automatic Steering System for Robust Disturbance Rejection". *IASME Transactions* 1.2 (2004).

17. Xiros NI., *et al*. "Theoretical and Experimental Investigation of Unmanned Boat Electric Propulsion System with PMDC Motor and Waterjet". *IMAREST Journal of Marine Engineering and Technology* A14 (2009): 27-44.