Water jet based autonomous surface vehicle for coastal waters operations

Diogo Machado*, Alfredo Martins*+, José Miguel Almeida*+, Hugo Ferreira*#, Guilherme Amaral*+, Bruno Ferreira*, Anibal Matos*†, Eduardo Silva*+

*INESC TEC Institute for Systems and Computer Engineering of Porto
+ISEP - School of Engineering, Porto Polytechnic Institute, Porto, Portugal
#ESEIG - Industrial Studies and Management School, Porto Polytechnic Institute, Porto, Portugal
†Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Porto, Portugal
Email: {diogo.c.machado, alfredo.martins, jose.m.almeida, hugo.a.ferreira
gasilva, bruno.a.ferreira, anibal.matos, eduardo.silva}@inesctec.pt

***Abstract*—This paper presents the design of low cost, small autonomous surface vehicle for missions in the coastal waters and specifically for the challenging surf zone.**

The main objective of the vehicle design described in this paper is to address both the capability of operation at sea in relative challenging conditions and maintain a very low set of operational requirements (ease of deployment).

This vehicle provides a first step towards being able to perform general purpose missions (such as data gathering or patrolling) and to at least in a relatively short distances to be able to be used in rescue operations (with very low handling requirements) such as carrying support to humans on the water.

The USV is based on a commercially available fiber glass hull, it uses a directional waterjet powered by an electrical brushless motor for propulsion, thus without any protruding propeller reducing danger in rescue operations. Its small dimensions (1.5 m length) and weight allow versatility and ease of deployment.

The vehicle design is described in this paper both from a hardware and software point of view. A characterization of the vehicle in terms of energy consumption and performance is provided both from test tank and operational scenario tests. An example application in search and rescue is also presented and discussed with the integration of this vehicle in the European ICARUS (7th framework) research project addressing the development and integration of robotic tools for large scale search and rescue operations.

I. Introduction

Coastal waters environment monitoring, security and safety operations are a major area of application for autonomous surface vehicles.

This paper presents the design of the SWIFT USV (Small Waterjet Intelligent Flexible Transporter) a low cost, small autonomous surface vehicle for missions in coastal waters and specifically for the challenging surf zone.

Due to its social and economical interest, the use of robotic marine surface systems in shallow and coastal waters has been addressed both by the research community [1], [2], [3] and by industry [4]. These type of vehicles usually address either monitoring and data collection tasks [1],[3] and/or are driven by security and safety applications [4], [5]. In order to operate in the marine environment, USV sizes and configurations tend to follow standard boats according to the specific application. Some USVs have been developed specifically for coastal waters such as the ALANIS [1] but their main use is in calm waters. Very small systems [6] are used mainly in very restricted environments and for scientific applications. These have the advantage of low operational requirements and also have very limited propulsion and limitations of application at sea.

A very different approach is taken by the EMILY [7] system developed for rescue operations near shore. This system has waterjet propulsion, providing high speed and manoeuvering capabilities for operation in the swell and deliver flotation to castaways in the water.

The SWIFT USV was designed in order to address both the capability of operation at sea in relative challenging conditions and maintain a very low set of operational requirements (ease of deployment).

One area of relevance for operations is the near at, or in the surf zone. For example, in integrated shoreline morphology studies such as the ones referred in [3] there's a limitation of operation at very shallow depth. The presented vehicle aims to be able to reach part of this area and perform a diversified set of missions. Other application comes from search and rescue requirements, and in particular of being able to reach human victims on the water and eventually providing assistance [8]. This vehicle provides a first step towards being able to perform general purpose missions (such as data gathering or patrolling) and to at least in a relatively short distances to be able to be used in rescue operations (with very low handling requirements) such as carrying support to humans on the water. It was also developed taking into account the lessons learned in the search and rescue research project ICARUS with the development of an unmanned robotized survival capsule [9].

The remainder of the paper is organized as follows: in the next section the vehicle architecture and the design options taken are discussed. Following, the system navigation and control are described along with the software implementation. The particular application of the USV in the context of the ICARUS search and rescue project is discussed in section IV. Some results from propulsion tests and field missions performed during the Portuguese Navy REX2014 exercises are then presented followed by a few concluding remarks and

978-1-4799-4918-2/14/$31.00 ©2014 IEEE

perspectives of future work.

II. System design

A. Design options

The system design options were taken having in mind the identified requirements for the vehicle. The unmanned surface vehicle should be able to operate in coastal waters, with low water depth, with propulsion power to operate in the surf zone and with relatively small dimensions and weight in order to be operated with a very reduced set of means.

A main design option was chosen for waterjet based propulsion. This provides high speed capabilities with no exposed propeller. The lack of exposed rotating parts reduces risk of mechanical failure in very shallow depths operations near rocks or in other confined spaces and also is convenient for operations near persons in the water such as in rescue operations, that are one of the main applications for the system.

Waterjet propulsion however has the major drawback of lower efficiency comparing with traditional thrusters and lack of control at low speeds. The tradeoff of reduced endurance versus high speeds was taken having in mind also the advantages of this type of systems being able to operate in the surf zone capable of coping with harsh waves.

As a consequence of the propulsion solution, the hull design was based on standard scaled waterjet crafts. This allowed the use of standard available small waterjet parts and also provided a conveniently small (1.2x0.4m) yet capable of transporting the required payload. Electric propulsion was chosen for control, environmental, maintenance, reliability and cost tradeoff issues.

The base hull was designed with a open middle compartment for payload, a bow watertight compartment housing control electronics, communications and system navigation sensors and a stern watertight compartment for the waterjet impeller, motor, motor control electronics, batteries and energy distribution electronics. This compartment has a low profile, providing the vehicle with low CG and with the batteries weight displaced to the stern, providing enough draught at the water intake for the propulsion (see figure 1).

Fig. 1. USV prototype

The vehicle is buoyant when submerged without payload and an optional cover for the cargo compartment leads to an always upright behaviour (is the cargo provides and appropriately low CG position).

B. System architecture

The system architecture follows the front seat/back seat approach [10], where the vehicle control is decoupled from the autonomy/application.

From the point of view of control basic vehicle control is performed by a small embedded commercial autopilot (microcontroller based) with the possibility of use of a higher layer of vehicle control onboard of a main computer.

The autonomy/application tasks can be performed in the payload (when this provides its computational power) or in the vehicle main CPU. Payload and vehicle supervision can interact either directly to the basic low level controller or to the main computer.

A diagram of the overall hardware system architecture can be observed in figure 2.

The vehicle waterjet positioned at the vehicle stern is oriented by a direction servo in order to control heading and has a mechanical cap reversing the water flow allowing for vehicle stopping and reversal. This cap is also controlled by a servo.

The low level controller (PixHawk PX4 autopilot [11]) allows for direct radio control, providing the references for the motor speed, direction and reverse servos.

A brushless DC motor is used to drive the impeller and its controlled by a dedicated Electronic Speed Controller unit. A 1730W, 800 rpm/V motor is used to drive the 49mm impeller (Graupner Jet5).

A set of 8S LiFePo4 batteries with 8400mAh of total capacity with a 25V nominal voltage provide energy for the propulsion and electronics.

Energy is provided to the bow computational electronics and to payload from the batteries trough a energy management module containing a set of DC/DCs providing regulated 24V, 12V and 5V outputs.

Main current consumption is monitored with a hall sensor system, and is provided to the low level controller. Due to the large power consumption from the waterjet, this is mainly used for motor monitoring and estimation of available energy in the batteries.

Two GPS receivers are connected both to the autopilot and to the main computer (when installed).

The main CPU is connected wireless to the supervision control through a IEEE 802.11a client. The use of a 5GHz communication link has proved more reliable in our past experience with unmanned surface vehicles [12] [3] near water. It also has the advantage of using a commonly more clear frequency band when used in operations with multiple vehicles and systems.

Connection between the two controllers is provided through serial port with 2 links, one control link using MAVLink protocol [11] and the autopilot console. When operating the

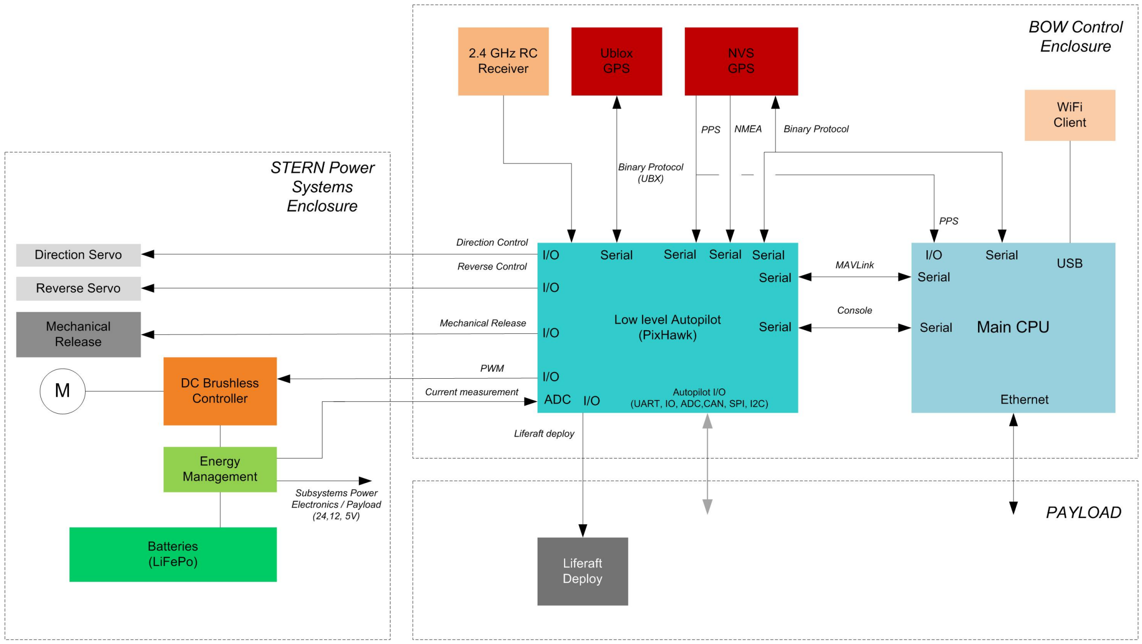

Fig. 2. System architecture

system without main computer (only with low level controller) is still possible to have remote access with a ZigBee (using a XBee module) wireless connection providing the MAVlink serial channel to a remote operator console.

Apart from the power, an ethernet link to the main computer and the low level controller I/O (comprising digital I/O, SPI, I2C and CANbus interface) are available to the payload.

III. Navigation and Control

Vehicle control is performed by the low level controller (PX4) with the option to have direct control from the main computer (trough the low level one using direct motor commands).

The PX4 autopilot is an open-source, open hardware project for an autopilot for a diverse set of autonomous vehicles. It was initially developed by under the PIXHAWK project of ETH Zurich (Swiss Federal Institute of Technology) and is currently community supported with commercial available implementations from the 3D Robotics company. It was originally designed for micro-aerial vehicles but it supports other system types.

This low level controller is based on a ARM Cortex-M4 microcontroller running NuttX. NuttX is an open source realtime operating system with a small footprint suitable for microcontrollers and embedded systems. It provides a "Linux like" development environment being capable of running both already available PX4 firmware applications (such as flight controllers or position estimators for aerial vehicles) or custom ones.

The main CPU runs a Linux based operating system and provides high level vehicle supervision and eventually computational power for mission oriented tasks.

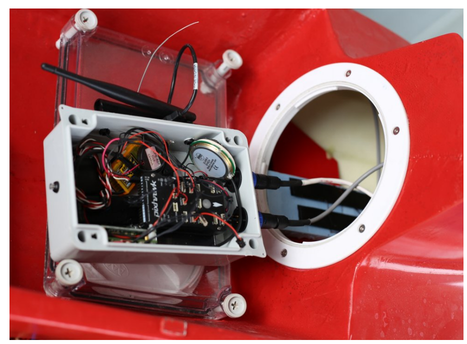

Fig. 3. PX4 low level vehicle controller and GPS receivers)

The ROS (Robotic Operating System) framework [13] is used in the main computer. ROS is an open source middleware developed for robotic applications. It has a very large user base, both in the main academic and research communities and also in industry. It provides an inter-process publish-subscribe communication mechanism, a development environment and a set of useful tools for debugging, logging and code producing. ROS has been ported to a vast number of robots and facilitates not only the development of complex robotic applications but also the integration of many currently available algorithms and solutions.

In ROS the robot application(s) is composed by a set of processes (nodes) subscribing and publishing information. This network of nodes can be altered dynamically in runtime, and each node can be contain a simple processing algorithm or a complex application. The framework is flexible not imposing a particular development structure. Another of its advantages comes from the large number of available packages, and in particular sensors and actuators drivers.

Different main CPUs can be used interchangeably according to computation power requirements and application specific needs. The common point is both the required standard hardware communication interfaces (2 serial communications to the low level controller, GPS serial connections; usually provided by a common USB interface; and wifi vehicle external communications) and the Linux + ROS software environment.

Two CPUs have been used alternatively as main computer: one an ARM based low power system (Odroid XU board with and octa core ARM Cortex A-15/A-7) and the other an Intel Atom Dual core system (Commell LP170C). Both systems have a very small footprint (10 x 7.5 cm) and relatively low power (4-6W for the ARM and 12 W for the Intel).

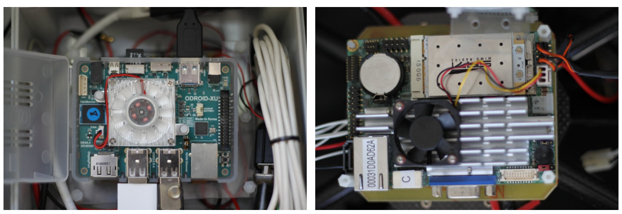

Fig. 4. Main CPU boards (left ARM based, right Intel Atom based)

The use of a standard Linux + ROS CPU box, allows to incorporate additional sensors such as cameras or acoustic sensors directly in the main CPU and take advantage de common useful libraries such as PCL or OpenCV for more advanced mission tasks.

The flexibility in the CPU choice also extends to the payload applications, since it is possible to load the vehicle with a set of sensors and their computational box and this payload can easily connect to the main CPU using the standard ROS communication mechanism.

Although not currently implemented, the possibility of having a standard Linux main CPU for vehicle control in addition to the low level controller allows for the use of the MOOS software framework [10] also common in marine robotics applications.

The vehicle uses GPS for localization. It is equipped with 2 GPS receivers: a module with a Ublox LEA-6H receiver and an additional magnetic compass incorporated and a NVS NV08C-CSM.

The first one is the standard GPS solution commonly used with the PX4. Since it has the magnetic compass, ceramic patch antenna integrated in one module is convenient both in terms of space and when necessary to keep the compass away from magnetic interferences (that is not the present case). This GPS was used initially in the development and was substituted by the NVS receiver (although it was not removed from the system providing an additional information source). The latter has higher update rate (10Hz) and supports a wider set of features and satellite constellations (SBAS, GLONASS and GALILEO).

Both GPS receivers are connected simultaneously to the low level controller and to the main computer. This allows for both only autopilot vehicle control or for control maneuvers from the main CPU.

In addition to the data connections, a PPS (Pulse Per Second) signal is provided by the NVS receiver to both the autopilot and the main computer. This signal uses de handshake pins of 2 serial connections and provides a time reference. The time reference signal is used in both cpus to allow for precise time synchronization.

The Chrony [14] time synchronization daemon runs in both CPUs and allows for the system clock to be synchronized both with GPS time and with external computers. Clock synchronization is useful for sensor data recording missions and posterior data registration and geolocation. It is also useful from a navigation point of view allowing for more precise solutions with distributed information.

The GPS precision solution varies with the fix determination method and available conditions. In standalone GPS position fixes the NVS receiver provides accuracy less than 2.5m (RMS). When there is the possibility of having a base station, a RTK differential solution is used. In this case accuracy in the order of a few decimeters is achievable in real time. The RTK solution is computed onboard taking advantage of the RTKLib software library [15].

The vehicle state (6DOF, position and orientation) is estimated in the low level controller with a EKF filter task. This filter uses information from the GPS (currently the NVS, the Ublox is used as a failsafe one), the magnetic compass and the inertial sensors integrated in the controller module. In general this state estimator solution is used for both vehicle control and localization purposes. It is available on the main computer in a ROS topic published by a ROS-MAVLink bridge node. When a higher precision is required (such as when using RTK solutions) or when different state estimators or additional navigation sources, the vehicle state is determined in a dedicated estimator node running on the main computer. In this case the vehicle control is determined on the main CPU instead of the low level controller.

Vehicle control has multiple modes of operation: direct RC control, waypoint mission control on the low level autopilot, remote teleoperation trough the main computer and mission control on the main computer. The latter can be either a waypoint define mission or a sequence of other control maneuvers. In both cases, the actuator references (speed and direction) are computed on the main CPU and the low level controller is bypassed.

The first basic operation mode corresponds to direct servo (and ESC reference) control from the RC receiver (a 2.4GHz DSM system) outputs, in this case the vehicle is controlled with a standard RC transmitter.

Mission control in the autopilot is defined by a set of waypoints sent (either by the remote console, or the main CPU) through the MAVLink connection. This allows both the mission to be defined from shore in a QGroundControl (Figure 5 left) application or locally defined (by an autonomy application running on the main CPU or from a payload CPU).

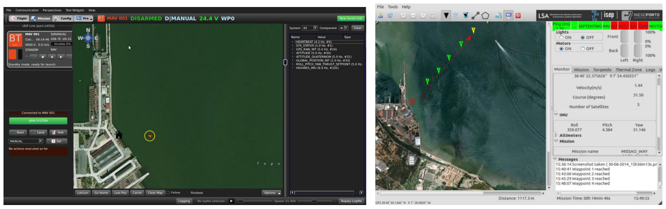

Fig. 5. QGroundControl console (left), ROAZ remote USV console (right)

Teleoperation though the main computer is usually performed remotely by a console (that needs not to be MAVLink compatible, for example our USV console [16], see Figure 5) that issues teleoperation or telecontrol commands to the vehicle main CPU. This computer then translates the received commands for the vehicle direct actuator controls. One example of this mode of operation is in the ICARUS rescue scenario where the C2C system teleoperates the vehicle using a JAUS based interoperability layer running on the robot main computer.

All these modes of operation allow for a very flexible set of application scenarios.

Vehicle guidance is provided by a simple LOS (Line of Sight) control law providing heading and speed references. This guidance scheme is similar to kinematic based guidance laws for nonholonomic vehicles [17] or path tracking ones as in [18].

The heading reference is given by a simple PID controller based on the heading error to the waypoint or a virtual reference point for path following. The speed reference is either defined as constant one or as a reference proportional to the distance error to the waypoint (for the last one). It should be noted that the vehicle has a very bad reverse motion capability (the water flow deflection cup is basically only useful for stoping providing a very low backwards control action). In view of this, under normal waypoint guidance negative speeds are not allowed and when the vehicle needs to reverse direction a turning maneuver is issued. As noted in [17] at the waypoint the control is not stable, however an acceptance radius is used to determine waypoint completion.

IV. ICARUS Search and rescue payload

A major application of the SWIFT USV is in search and rescue operations. One of the main requirements for the vehicle development was its use and evaluation as an unmanned liferaft transporter for the ICARUS project.

ICARUS [19] is as European Union research project in the area of search and rescue funded under the 7th Framework program led by the Royal Military Academy in Brussels, Belgium. It is a project with 24 partners from 10 European countries with a total budget of 17MEuro and aims to develop and integrate robotic tools for large scale disasters. The project has two example scenarios: one large scale urban search and rescue scenario and a maritime scenario.

For the maritime scenario a large scale accident is addressed with a large number of persons in the water. This scenario is led by INESC TEC and the Portuguese partners (namely the Portuguese Navy) with the participation of the NATO CMRE (Center for Marine Research and Experimentation) and the Calzoni company from Italy.

Fig. 6. ICARUS marine rescue scenario

ICARUS intends to contribute to the usage and integration of robotic solutions in the existing human based rescue framework. For the maritime scenario, a set of systems is considered. In this concept (see Figure 6) aerial and marine surface unmanned vehicles are considered. Fixed wing UAVs (Unmanned Aerial Vehicles) with relatively long endurance, provide large area coverage, situational awareness and victim detection and tracking. Small rotor wing UAVs are used for proximity tracking, inspection and localized information. Fast USVs carry assets to the accident area and deploy unmanned liferaft robotized capsules (UCAPS) that deliver flotation to near victims or groups of victims.

These UCAPS can be deployed from USVs or other manned vehicles, should cover the last hundred meters to the victims.

Multiple approaches were studied for the development of these robotic liferafts [9] from custom designed liferafts, transporter vehicles plus standard liferafts, linked or not linked solutions upon inflation, etc.

The approach to design a liferaft robotic transporter was taken for its versatility. It is thus possible to use standard liferafts reducing cost and facilitating the integration of these systems in the existing rescue infrastructure.

A system capable of transporting a 4 person SOLAS approved liferaft was already developed [9] in the project context.

The presented SWIFT USV was intended to test and validate a smaller UCAP system and to test waterjet propulsion solution. This type of propulsion is already used in rescue unmanned vehicles such as EMILY and provides high speed at the expense of efficiency and endurance. Since the UCAPS considered in the ICARUS context are to operate in the last phase of the rescue operations, their endurance and range do not need to be large.

Performing as an ICARUS UCAPS was one of the key requirements in the vehicle design and the reduced dimensions (in comparison with other previous solutions) imply smaller payload carrying capability and thus smaller liferafts.

However the vehicle is still able to carry a airplane certified liferaft for 4 persons (Figure 7).

Fig. 7. USV carrying liferaft for 4 persons

A similar deployment system is considered for this vehicle as in [9]. This is a simple ramp where each vehicle falls to the water when required. This deployment system is very simple to implement in both USVs to be used in the ICARUS project demonstration and in a large variety of vehicles.

Deployment from the USV is performed by a mechanical latch activated by the capsule itself. Thus there is no power or communication wiring on the carrying vessel apart from the possibility of transporting the deployment ramp. This implies that some energy is spent in the SWIFT USV during transport since it must be alive to detach itself upon command (either remote or preprogrammed).

V. Results

A. Propulsion analysis

A series of tests were performed with the SWIFT USV in our test tank to evaluate different motor solutions for water jet drive. In these tests, the vehicle bollard pull was measure for 4 different motors: 1200 rpm/V, 800 rpm/V, 600 rpm/V, and 385 rpm/V.

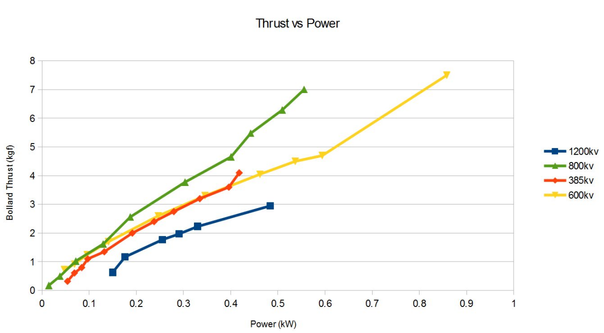

Fig. 8. Bollard pull tests for multiple motors

From the analysis in figure 8 the motor providing higher thrust was the 800 rpm/V. With this motor with an electrical power of 550W a thrust force of 7 Kgf.

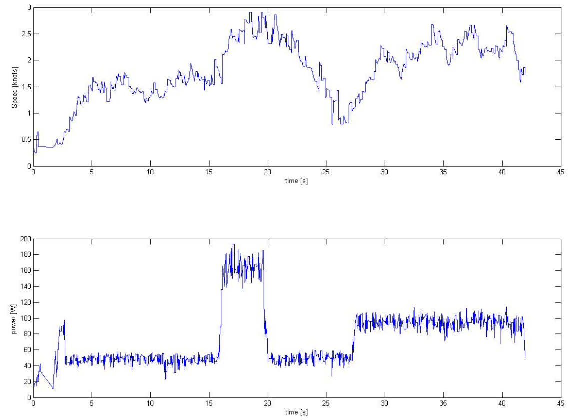

Fig. 9. Vehicle speed and power at low speed

The vehicle low speed profile power consumption was analyzed in a test mission in calm waters. This profile is relevant for data gathering applications and longer endurance. In figure 9 it can be observed that for the testing conditions (vehicle total weight of 16Kg) the power consumption for a speed of 1.5Knots is about 50W. Without additional batteries this leads to a mission endurance of about 4hr or 10Km. This endurance time can easily be doubled by adding less than 2 Kg of batteries.

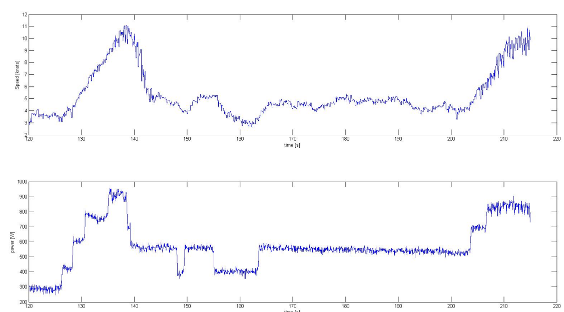

Fig. 10. Vehicle speed and power at high speed

In figure 10 the high speed regime consumption is presented. For this range of speeds one can observe that at about 11 knots the power consumption is around 900W.

B. Field Tests

The SWIFT USV was tested in field trials at the Portuguese Navy's REX2014 - Robotic Exercises 2014. These exercises were performed at the Alfeite Naval Base, near Lisbon in July 2014. The exercises objectives were the preparation and test of partial integration of systems under the ICARUS project and the assessment of new robotic technologies and their use for the Portuguese Navy.

During these tests the vehicle performed missions in the Tagus river under relatively calm waters. The main objectives were to validate the system operationally. It was also addressed the integration issues in the ICARUS C2C and the operation with multiple vehicles (namely with the ROAZ USV).

The deployment from a larger USV was tested and validated in the INESC TEC Roaz USV (see figure).

Fig. 11. Deployment from ROAZ USV

The SWIFT USV was tested in the REX2014 exercises with the rescue payload with a total wight of 16Kg (8Kg of payload plus 8Kg of vehicle) moving at a speed of 12 knots.

Fig. 12. Tests with rescue payload

On figure 13 one can observe a partial USV trajectory performed upon deployment from ROAZ.

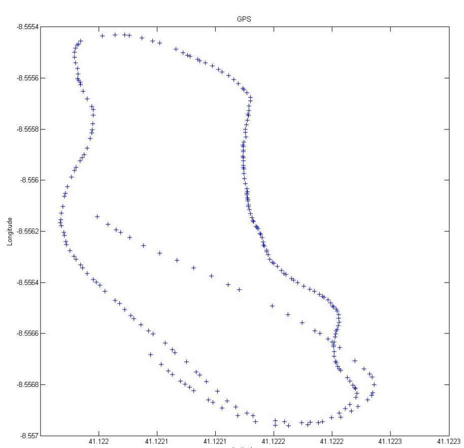

Fig. 13. SWIFT USV Trajectory in a mission test at REX2014

With the standard battery configuration (8400 mAh, 25V) and the rescue payload (liferaft with 8Kg) the vehicle is capable of operate at 10 knots for about 15min. This leads to a 4.5 Km of autonomy at high speed for rescue missions.

VI. Conclusions

In this paper a small USV for coastal operations was presented. This robotic vehicle was designed for multiple operations in very shallow water and with the capability of operating in the surf zone. This system targets a relatively new area of operation in the intertidal zone or in difficult environments.

A waterjet based design provided some cargo payload with flexibility in applications and also advantages in terms of speed and not having protruding elements. This aspects are particular relevant for rescue operations that were one of the main applications envisioned for the system.

A modular design with an embedded autopilot and optional additional main computer provides flexibility, allowing for multiple control possibilities form basic remote piloting and waypoint GPS based navigation to more complex maneuvers (such as adaptive sensing missions depending on sensing data).

The onboard main computer runs the commonly used ROS framework and is capable of integrating multiple command and control infrastructures, allowing for standard communication methods and facilitating the integration of the system with other systems.

One example of this flexibility is provided by the ICARUS project search and rescue application where the vehicle is used as unmanned capsule transporter and integrates to the ICARUS command and control framework using JAUS. The use of the USV in this application is also discussed in the present work.

Propulsion tests were discussed and the system was validated in field trials during the Portuguese Navy REX2014 exercises performed at Base Naval do Alfeite, Portugal in July of 2014. Vehicle integration to the ICARUS C2C as an unmanned survival robotic capsule was also already partially validated at these exercises.

These preliminary validation experiments provide confidence on the application of the SWIFT USV in rescue missions. Further tests will be performed in more demanding environments, namely under strong waves. Automatic deployment and inflation of liferaft will also be tested. Full integration with the ICARUS C2C will also be performed.

The vehicle application in data gathering missions such as bathymetry in the intertidal or very shallow water will be evaluated with field trials.

Acknowledgments

The authors acknowledge the helpful support from the Portuguese Navy. The project has also being supported by ISEP-IPP and INESC TEC.

This work is partially financed by European FP7 Research project ICARUS - Integrated Components for Assisted Rescue and Unmanned Search Operations - grant agreement no:285417 and is co-financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF) and by the ERDF - European Regional Development Fund through the COMPETE Programme and by National Funds through the

FCT - Portuguese Foundation for Science and Technology within project FCOMP-01-0124-FEDER-022701.

REFERENCES

[1] M. Caccia, M. Bibuli, R. Bono, G. Bruzzone, G. Bruzzone, and E. Spirandelli, "Aluminum hull USV for coastal water and seafloor monitoring," in *OCEANS 2009-EUROPE*. IEEE, May 2009, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5278309

[2] J. Curcio, P. A. McGillivary, K. Fall, A. Maffei, K. Schwehr, C. Kitts, and P. Ballou, "Self-Positioning Smart Buoys, The "Un-Buoy" Solution: Logistic Considerations using Autonomous Surface Craft Technology and Improved Communications Infrastructure," in *Proc. MTS/IEEE Oceans 2006*, Boston, Sep. 2006.

[3] H. Ferreira, C. Almeida, A. Martins, J. Almeida, N. Dias, A. Dias, and E. Silva, "Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle," in *OCEANS 2009-EUROPE*. IEEE, May 2009, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5278235

[4] "Protector Unmanned Surface Vehicle Web Page, Rafael Advanced Defense Systems Ltd." [Online]. Available: http://www.rafael.co.il/

[5] T. J. Pastore and R. T. Kessel, "Unmanned surface vessels for surface and subsurface threats in harbours : background and practical lessons," in *UDT Europe, Undersea Defence Technology Europe*, no. June, Cannes, Jun. 2009.

[6] Clearpath Robotics Inc., "Kingfisher Unamanned Surface Vessel Web Page." [Online]. Available: http://www.clearpathrobotics.com/kingfisher/

[7] "Hydronalix (EMILY developers) web page." [Online]. Available: http://www.hydronalix.com/

[8] A. Martins, A. Dias, J. Almeida, H. Ferreira, C. Almeida, G. Amaral, D. Machado, J. Sousa, P. Pereira, A. Matos, V. Lobo, and E. Silva, "Field experiments for marine casualty detection with autonomous surface vehicles," pp. 1–5, 2013.

[9] A. Matos, E. Silva, N. Cruz, J. Alves, D. Almeida, M. Pinto, A. Martins, J. Almeida, and D. Machado, "Development of an Unmanned Capsule for large-scale maritime search and rescue," in *MTSS/IEEE OCEANS 2013*, San Diego, California, USA, Sep. 2013, pp. 1–8.

[10] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles with MOOS-IvP," *Journal of Field Robotics*, vol. 27, no. 6, pp. 834–875, Nov. 2010. [Online]. Available: http://doi.wiley.com/10.1002/rob.20370

[11] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, May 2011, pp. 2992–2997. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980229

[12] A. Martins, J. M. Almeida, H. Ferreira, H. Silva, N. Dias, and E. Silva, "Autonomous surface vehicle docking maneuvre with visual information," in *Proc. ICRA07 IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 2007.

[13] M. Quigley, K. Conley, B. P. 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*, 2009.

[14] "Chrony Daemon Webpage." [Online]. Available: http://chrony.tuxfamily.org/

[15] T. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB," in *International Symposium on GPS/GNSS*, 2009, pp. 4–6.

[16] N. Dias, C. Almeida, H. Ferreira, J. Almeida, A. Martins, A. Dias, and E. Silva, "Manoeuvre based mission control system for Autonomous Surface Vehicle," in *OCEANS 2009-EUROPE*. IEEE, May 2009, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5278234

[17] A. Astolfi, "On the stabilization of nonholonomic systems," in *Proc. 33rd Conference on Decision and Control*, no. December, 1994, pp. 4–9.

[18] M. Aicardi, G. Casalino, and G. Indiveri, "A planar path following controller for underactuated marine vehicles," in *Proc. 9th Mediterranean Conference on Control and Automation*, no. June, Dubrovnik, Croatia, 2001, pp. 27–29.

[19] G. De Cubber, D. Doroftei, D. Serrano, K. Chintamani, R. Sabino, and S. Ourevitch, "The EU-ICARUS project: Developing assistive robotic tools for search and rescue operations," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, Oct. 2013, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6719323