Advances in Automated Ship Structure Inspection

Thomas Koch, Atlantec Enterprise Solutions GmbH, Hamburg/Germany,
thomas.koch@atlantec-es.com
Sankaranarayanan Natarajan, DFKI GmbH, Robotics Innovation Center, Bremen/Germany,
sankar.natarajan@dfki.de
Felix Bernhard, DFKI GmbH, Robotics Innovation Center, Bremen/Germany,
felix.bernhard@dfki.de
Alberto Ortiz, University of Balearic Islands, Palma de Mallorca/Spain,
alberto.ortiz@uib.es
Francisco Bonnin-Pascual, University of Balearic Islands, Palma de Mallorca/Spain,
xisco.bonnin@uib.es
Emilio Garcia-Fidalgo, University of Balearic Islands, Palma de Mallorca/Spain,
emilio.garcia@uib.es
Joan Jose Company Corcoles, University of Balearic Islands, Palma de Mallorca/Spain,
joanpep.company@uib.es

Abstract

Inspection of ship-board structures by humans is a time-consuming, expensive and commonly hazardous activity, creating the need to search for better solutions. Simultaneously the desire for broader and intensified data acquisition grows to provide a better basis for refined condition assessment. As the implementation of automated and increasingly autonomous robotic platforms for everyday use becomes more realistic, we have realised various approaches to robotic ship inspection, which are based on advances in multiple dimensions: light-weight robotic platform development, complex data pre-processing functions and large-scale data analytics. Results of investigations carried out as part of a project for enhanced ship safety are presented.

1. Introduction and background

Inspections on-board sea-going vessels are regular activities being initiated not only due to applicable classification and statutory regulations, but also performed for ship operators to assess the condition of a vessel's structure and to ensure its integrity. Since unexpected disruptions of vessel service availability tends to be very costly, ship operators have a vested interest to monitor the condition of the structure at a required level of detail. Unfortunately, continuous monitoring of stress and strain conditions, identification of defects or assessment of corrosion is not trivial and quite expensive. Since inspections are and will be an important source of information for structure condition assessment, it seems necessary to try to reduce the effort and cost related to these activities.

With the constant progress that can be observed in the field of robotics and automation, the introduction of automated tools for support of inspections appears to be a logical step to improve the efficiency and cost of inspections. However, some challenges must be overcome in this field of application, such as dealing with the harsh operating environment, complying with the safety requirements and managing the limited access and space. This has led to the research approach of using different robotic platforms and a wide range on sensors to determine useful combinations of technology for different operational scenarios. On the other hand, automated devices can still only cover part of the activities carried out during inspections. Critical (and usually limiting) design factors are: in-situ data processing capability, energy supply vs. payload, sensor sensitivity and reliability, signal quality, autonomy and fail-safe operation. Therefore, it is essential to combine the actual robotic devices with a carefully configured support system that provides functions for control, data acquisition and data post-processing.

The application of automated tools is intended to provide various important benefits. An important starting condition is that the overall cost of the robotic platforms is already quite low and will continue to decrease, eventually reaching more or less the commodity level. This will enable ship operators to equip vessels with a tailored set of inspection devices, which are instantly available for operation. This will lower the entry barrier considerably for ad-hoc inspections of limited scope, which are more easily arranged with less impact on operations of the vessel. As in many other industrial application areas, the use of automated tools will substantially reduce the exposure of operating staff to hazardous conditions. Due to the automated data acquisition, the hull structure coverage can be extended both in terms of area/volume as well as in number of parameters collected.

Being able to collect data and assess a structure’s condition more easily will contribute to building up a wealth of processed data. It will help to establish a more detailed time history of various condition parameters, enabling or improving the use of advanced analytics for structure condition assessment leading to new possibilities for forecasting and damage prevention.

This paper describes the automation efforts pursued within the framework of the EU-funded project INCASS, *INCASS (2013)*, in accordance to the aforementioned. In more detail: Section 2 describes two robotic systems which have been developed in the framework of the project INCASS; Section 3 outlines the procedures which implement the collection of sensor data; Section 4 overviews the analysis of images for defect detection; Section 5 details how the sensor data are collected, processed and delivered to the central system; Section 6 is for data management and big data processing; finally, Section 7 concludes the paper and suggests future work.

2. Robotic systems for on-board inspection

In the recent years, advanced technological devices are finding their way into the marine vessel’s inspection area. In this respect, the INCASS project considers the inspection of interior areas of a vessel, e.g. cargo holds, by means of robotic platforms. The two robotic devices which are available, namely an aerial platform and a magnetic crawler, are the result of the re-design of two platforms developed to accomplish the objectives of project MINOAS, *MINOAS (2009)*, in accordance to the feedback received from vessel surveyors during several field trials at the end of that project. In short, they are intended to be lightweight and fast deploying vehicles capable of reaching points several meters high above the ground. The main parameters of both platforms can be found in Table I. They both are reviewed in more detail in the following sections.

Table I: Main technical specification of the INCASS robotic platforms.

	Aerial platform (Pelican)	Magnetic crawler
Size (L x W x H)	650 mm × 650 mm × 270 mm	330 mm × 300 mm × 130 mm
Weight	1700 g	1230 g
Speed	0.5 – 2 m/sec	0.5 m/sec on vertical adhesive walls
Propulsion	• 4 × 160 W brushless motors • 10” propellers	• 2 × 12V DC geared motors with encoders
Sensors	• 3-axis IMU • Laser scanner / optical flow • Height meter • 2 Mpx still camera • 12 Mpx Full HD video camera	• Motor encoders • HD 720p USB camera • 3-degree of freedom accelerometer
Auxiliary components	• 2 × LED spotlights (3W)	• 3 x LED • Lithium-Polymer saver
Communication & interaction	• Dual 2.4 - 5GHz Wi-Fi LAN • Joystick / gamepad	
Power	11.1V, 4500mAh, 3-cell Lithium-Polymer	11.1V, 800mAh, 3-cell Lithium-Polymer

2.1 Aerial platform

Platform overview

In line with the robotic platform developed for the MINOAS project, the INCASS aerial vehicle is based on a multi-rotor design. The control software has been configured to be hosted by any of the research platforms developed by Ascending Technologies (the quadcopters Hummingbird and Pelican, and the hexacopter Firefly), although it could be adapted to other systems. The AscTec vehicles are equipped with an inertial measuring unit (IMU), which comprises a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer, and two ARM7 microcontrollers. Attitude stabilization control loops linked to the on-board IMU and thrust control run over the main ARM7 microcontroller as part of the platform firmware. The manufacturer leaves almost free an additional secondary ARM7 microcontroller which can execute on-board higher-level control loops.

All platforms are fitted with a navigation sensor suite that allow them to estimate the vehicle *state*, which comprises 3-axis speed (v_x, v_y, v_z), the flying height z and the distances to the closest obstacles in different orientations, e.g. left (d_l), right (d_r) and forward (d_f). These estimations can be performed by means of different sensor combinations leading to different weight, volume occupied and power consumption. This permits preparing for the inspection application either vehicles of low payload capacity (lower-cost platform) or vehicles able to lift a heavier sensor suite (higher-cost platform). By way of example, Fig. 1 shows a Hummingbird platform, fitted with two lightweight optical-flow sensors for speed estimation, ultrasound sensors for obstacle detection and an infrared height-meter, and a Pelican platform fitted with a laser scanner for speed estimation and obstacle detection, and a laser-based height-meter.

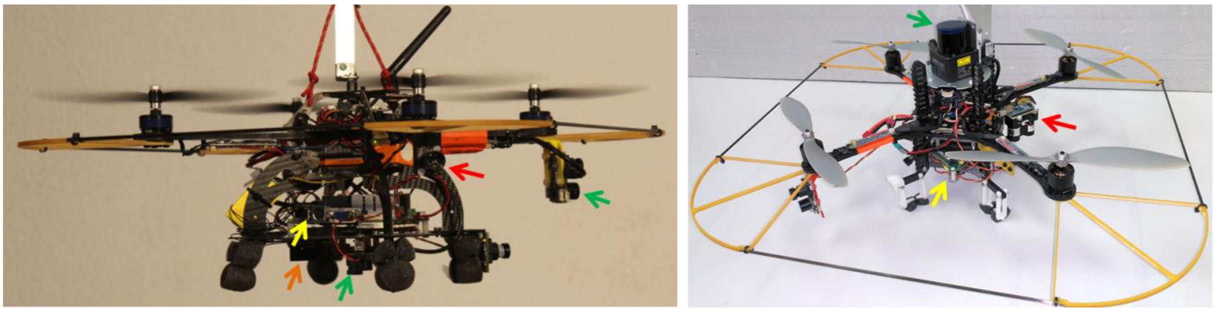


Fig. 1: [left] A Hummingbird platform featuring optical flow sensors (green), an infrared height-meter (orange), and ultrasound sensors (red). [right] A Pelican platform featuring a laser scanner (green) and a laser-based height-meter (red). The embedded PC is indicated by a yellow arrow in each case.

Besides the navigation sensor suite, all platforms carry, in accordance to their payload capacity, one or several cameras for collecting the expected visual inspection data.

To finish, apart from the two ARM7 microcontrollers integrated in the *flight control unit* of the AscTec platforms, all vehicles carry an embedded PC, which avoids sending sensor data to a base station, but process them on-board and, thus, prevent communications latency inside critical control loops. Once again, the different platforms are fitted with boards compatible with their payload limits, e.g. the Hummingbird of Figure 1 features a Commell LP-172 Pico-ITX board fitted with an Intel Atom 1.86 GHz processor and 4 GB RAM, while the Pelican carries an Intel NUC D54250WYB with an Intel Core i5-4250U 1.3 GHz processor and 4 GB RAM.

Control software

The aerial platforms integrate a control architecture that follows the *supervised autonomy* (SA) paradigm, *Cheng and Zelinsky (2001)*. This is a human-robot framework where the robot implements a number of autonomous functions, including self-preservation and other safety-related issues, which simplify the intended operations for the user, so that he/she, which is allowed to be within the general platform control loop, can focus in accomplishing the task at hand. Within this framework, the communication between the robot and the user is performed via qualitative instructions and explanations: the user prescribes high-level instructions to the platform while this provides instructive feedback. In our case, we use simple devices such as a joystick or a gamepad to introduce the qualitative commands and a *graphical user interface* (GUI) to receive the robot feedback. Joystick commands and the GUI are handled at a *base station* (BS) linked with the MAV via a Wi-Fi connection.

The control software is organized around a layered structure distributed among the available computational resources. On the one hand, the *low-level control* layer implementing attitude stabilization and direct motor control executes over the main microcontroller as the platform firmware provided by the manufacturer, *Gurdan et al. (2007)*. On the other hand, *mid-level control*, running over the secondary microcontroller, comprises height and velocity controllers which map input speed commands into roll, pitch, yaw and thrust orders. Lastly, the *high-level control* layer, which executes over the embedded PC, implements a reactive control strategy coded as a series of ROS[1] nodes running over Linux Ubuntu, which combine the user desired speed command with the available sensor data –v_x, v_y, and v_z velocities, height z and distances to the closest obstacles d_l, d_r and d_f –, to obtain a final and safe speed set-point that is sent to the speed controllers.

Speed commands are generated through a set of robot behaviours organized into a hybrid competitive-cooperative framework, *Arkin (1998)*. That is to say, on the one hand, higher priority behaviours can overwrite the output of lower priority behaviours by means of a suppression mechanism taken from the *subsumption* architectural model. On the other hand, the cooperation between behaviours with the same priority level is performed through a *motor schema*, where all the involved behaviours supply each a motion vector and the final output is their weighted summation. An additional flow control mechanism selects, according to a specific input, among the outputs provided by two or more behaviours.

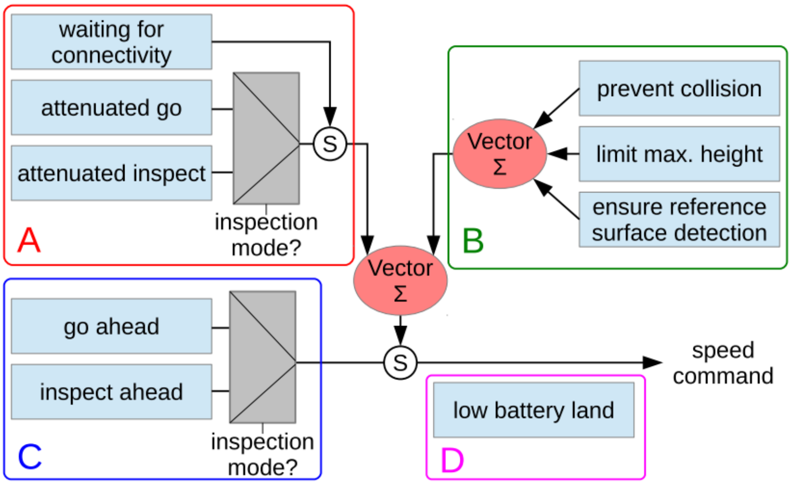


Fig. 2: Behaviour-based upper control layer.

Figure 2 details the behaviour-based architecture, grouping the different behaviours depending on its purpose. A total of four general categories have been identified for the particular case of visual inspection: (a) *behaviours to accomplish the user intention*, which propagate the user desired speed command, attenuating it towards zero in the presence of close obstacles, or keeps hovering until the Wi-Fi link is restored after an interruption; (b) *behaviours to ensure the platform safety within the environment*, which prevent the robot from colliding or getting off the safe area of operation; (c) *behaviours to increase the autonomy level*, which provide the platform with higher levels of autonomy to both simplify the operation and to introduce further assistance during inspections; and (d) *behaviours to check flight viability*, which checks whether the flight can start or progress at a

[1] Robot Operating System (http://www.ros.org)

certain moment in time. Some of the behaviours in groups (a) and (c) can operate in the so-called *inspection mode*. While in this mode, the vehicle moves at a constant and reduced speed (if it is not hovering) and user commands for longitudinal displacements or turning around the vertical axis are ignored. In this way, during an inspection, the platform keeps at constant distance/orientation with regard to the front wall, for improved image capture.

2.2 Magnetic crawler

Platform overview

The main aim for developing the magnetic crawler is to get a closer visual inspection. The use of magnets or suction pads to locomote on the inspection surface of a marine vessel is already available in research and also industry robots, CROMSCI, *Jung et al. (2010)*, MagneBike, *Tache et al. (2010)*, MARC, *Bibuli et al. (2012)*. As marine vessels are made of solid steel using magnetic wheels or magnetic tracks system offers an efficient solution for both locomotion and also for traction. In this section the hardware design and software architecture of the magnetic crawler is explained. The first design concept of this magnetic crawler was described in *Eich and Voegele (2011)*. During the development of the current magnetic crawler the surveyor's recommendations are included in the design criteria. Figure 3 shows the newly developed magnetic crawler. The main design criteria for the magnetic crawler are to be lightweight; deployment does not require any extra installation, capable of teleoperating and semi-autonomous navigation, wireless communication, tiltable camera system and easy to use for the surveyor.

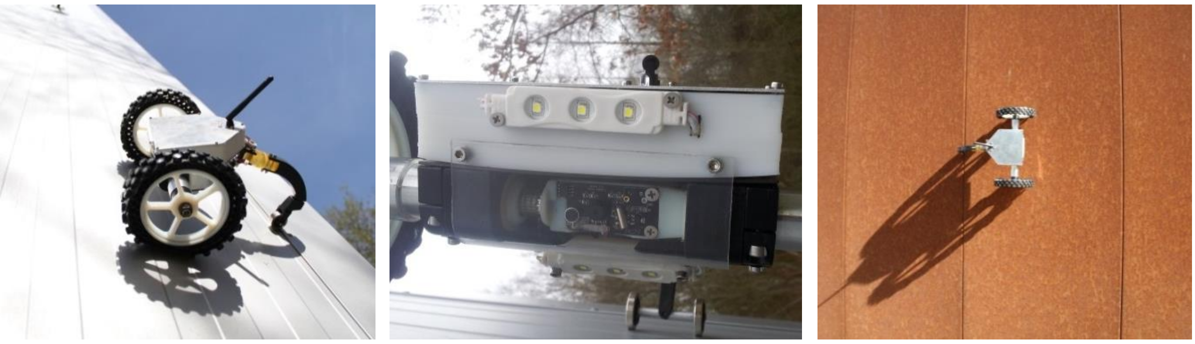


Fig. 3: The lightweight magnetic crawler. (source: DFKI GmbH)

The magnetic crawler is controlled and data are processed by an Odroid-U3 single-board computer with a quad-core processor. The computer has a 2GB RAM, 3 high speed USB port, Ethernet and GPIO/UART/I2C ports. The magnetic crawler is equipped with the sensors like accelerometer, high definition camera and two motor encoders. A custom designed printed circuit board was developed based on a PIC board housing an ADXL345 accelerometer, a DC/DDC converter, a USB-IO board. A 720p high definition USB based camera is mounted on a tilt unit and it is installed in front of the crawler. This tilt unit is controlled by a micro-servo motor. The tiltable camera is advantageous over a static camera as it will be helpful to navigate the crawler by tilting the camera in the crawler moving direction. In order to get a close-up image the camera can be tilted downwards. Adequate light is always an issue in taking inspection images in the marine vessel. It is not always possible to illuminate an entire tank or a cargo hold for acquiring image during the inspection. There might be certain areas which were not properly illuminated due to lack of light source near the target area or due to the shadow caused by the crawler. To overcome this issue the crawler is fitted with two sets of LEDs, one set is fixed in front of the crawler and the other set is fixed below the crawler. All these components are controlled through the USB-IO board. The crawler is equipped with a wireless local area network (WLAN) which is used in transmitting camera data such as images and videos and also for communicating with the crawler.

The magnetic crawler is actuated by two 12 V DC motors that drive the two magnetic wheels. In order to increase the crawler stability while climbing on corrugated metal structures, a flexible tail is attached on the rear part of the crawler through passive joints. A neodymium permanent magnet ring is fixed to the end of the tail. The magnetic wheel is one of the most challenging parts in the development of the crawler. Instead of fitting the magnets directly in to the wheel, a soft wheel-belt was designed to hold the magnet around the wheel frame. The cups-like structure in the soft wheel belt holds the magnets in place and also helps in preventing the magnets losses. The soft wheel belt was manufactured by developing a negative mould out of CNC-milling-wax and then casting a positive mould with a flexible polyurethane casting system (Figure 4). The magnets where glued with a two-component epoxide glue into the roughened polyurethane cups, their adhesion pulled them to the bottom direction (iron sheet) while drying.

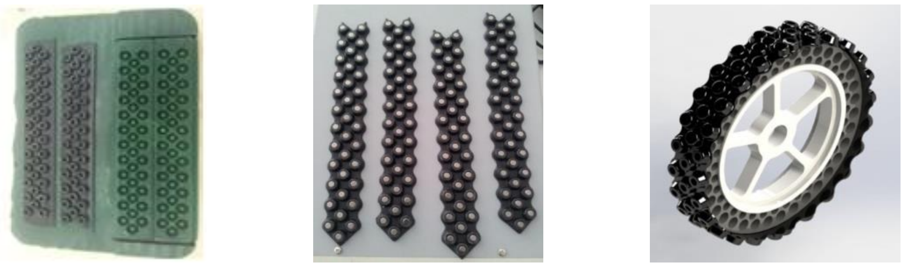


Fig. 4: Wax mould with the casted polyurethane stripes (left). Polyurethane stripes with integrated magnets (middle). Wheel-CAD model (right) (source: DFKI GmbH)

Control Software

A graphical overview of the magnetic crawler's control architecture is shown in figure 5. The crawler consists of several hardware and software components. The Robot Operating System (ROS) is used as a software framework in the crawler. To maintain the light weight, only low level software components such as motor controller, driver for camera, accelerometer, and camera servo motor are running in the crawler computer. As self-localisation is not feasible by using only crawler's on-board sensors, an external tracking unit is also used along with crawler's on-board sensors. The external tracking unit consists of a camera, a laser-based distance measurement unit and two dynamixel servos. The bright LED light which is fitted on the top of the crawler for tracking purpose will be detected by the tracking unit camera. With the help of the two servo motors the tracking unit controller is capable of tracking the moving crawler. A detailed design and the control of the tracking unit were described in *Eich and Voegele (2011)*. The crawler can be teleoperated by an operator with a help of a joystick. A graphical user interface (GUI) is developed to provide a surveyor an overview of the crawler's current status and also for sending higher level commands. The crawler's current status such as current pose, image from the camera are shown in the GUI. The surveyor can also use this GUI for controlling the crawler movement, sending the inspected image along with its pose value to the database, controlling the tilt angle of the camera and for turning ON/OFF the LED light.

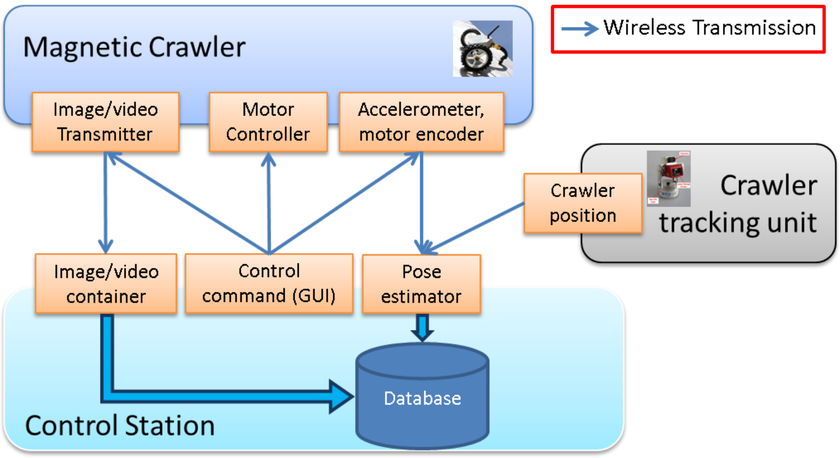


Fig. 5: Overview of the software architecture of the lightweight magnetic crawler.

3. On-site data collection

As described above, the main purpose of the two robotic platforms is to serve as remote sensors of inspection data. In this section, we describe the sort of data collected as well as any processing that is performed.

3.1 Aerial platform

During flight, any of the aerial platforms can collect pictures on demand or at a fixed rate, e.g. 10 fps, as well as log flight data. The latter includes the vehicle pose, i.e. 3D position and 3D attitude, the vehicle speeds and the distances to the closest obstacles. Of particular relevance is the vehicle pose, which permits associating a 3D position to the defects found. For this purpose, two *simultaneous and localization methods* (SLAM) have been integrated on-board the aerial platforms given their different payload capacities. One adopts a laser-based SLAM strategy while the other is a visual single-camera SLAM solution: while the first one aligns consecutive laser scans to estimate the vehicle motion from one time instant to the next, the second solution matches image features across consecutive images, projects them in 3D and determines the corresponding 3D transformation. Depending on the robot on-board computational capabilities, the latter process can run on-line or off-line, after flight. By way of example, Fig. 6 (top) shows the paths estimated by the laser-based approach for two flights, as well as illustrates the defect localization process after visual inspection through the projection, as different coloured boxes, of the bounding boxes of the defects found during a flight. Figure 6 (bottom) shows the robot during a visual inspection on board an oil tanker and one of the images captured.

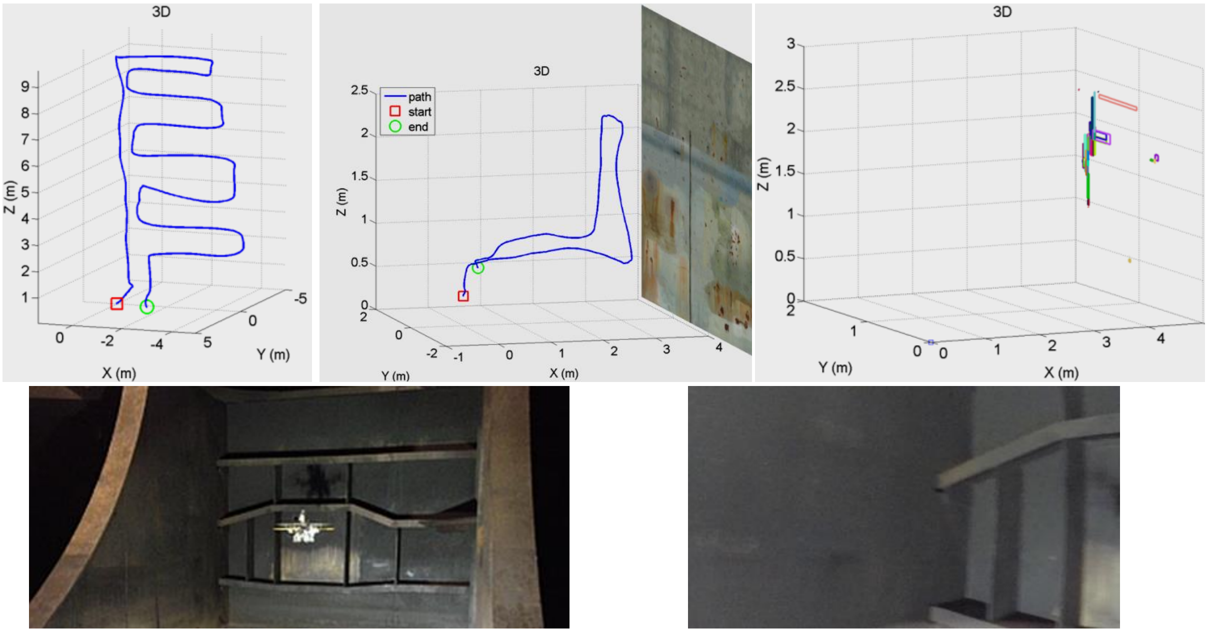
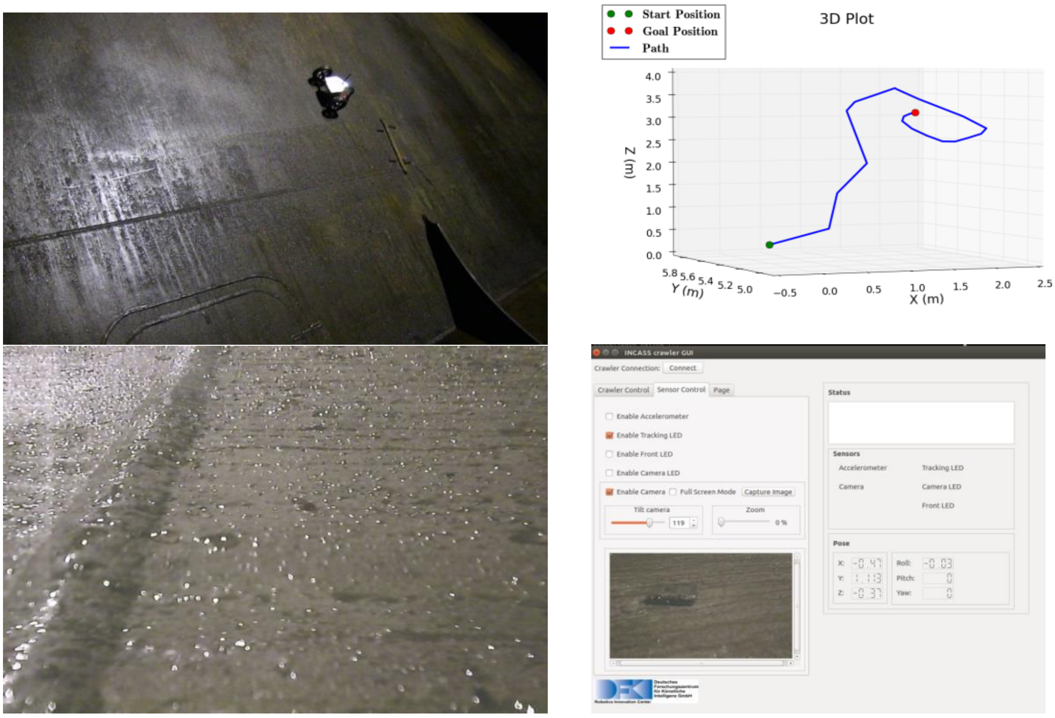


Fig. 6: Paths estimated after a flight and illustration of the defect localization process after visual inspection (top). Visual inspection inside an oil tanker and one of the images collected (bottom).

3.2 Magnetic crawler

During the visual inspection, the surveyor needs, besides the visual data, the actual position of the visual data with respect to a reference frame. A pose estimator module running on the control station is responsible for estimating the crawler's position along with its orientation with respect to a reference frame. Since the position information sent by the crawler's tracking unit and crawler's orientation and odometry information are transmitted separately, the pose estimator module needs to synchronise all the incoming data based on the timestamps before estimating the crawler's current pose. The visual data along with its corresponding pose value are stored in a database on the crawler's control station. The inspection information is all saved in a XML file. Figure 7 shows the inspection area, the path followed by the crawler, and a sample inspection image.



Crawler Connection: Connect

☐ Enable Accelerometer
☑ Enable Tracking LED
☐ Enable Front LED
☐ Enable Camera LED
☑ Enable Camera ☐ Full Screen Mode

Tilt camera: 119
Zoom: 0 %

X: -0.47 Roll: -0.03
Y: 1.13 Pitch: 0
Z: -0.37 Yaw: 0

Fig. 7: Visual inspection inside an oil tank (top, left). Path followed by the crawler (top, right). Sample image during the inspection (bottom, left). The GUI showing an image and its pose (bottom-right).

4. Image analysis and defect assessment

In this work, we consider defects as rare phenomena that may appear on a regular surface or structure. Since they are rare, the probability that an area is affected by a defect is rather low. This low probability can be used as an indicator of image saliency, and thus highlight image areas suspicious of being defective. Next sections describe saliency computation as well as the performance attained.

4.1 Bayesian Approach for Saliency Computation

Similarly to *Zhang et al. (2008)*, we make use of a Bayesian approach to compute a saliency map Σ_{ij}:

$$\Sigma_{ij} = \frac{1}{p(F=f_{ij})} p\left(F = f_{ij} \middle| T = \delta\right). \tag{1}$$

where f_{ij} is the value of the feature F found at an image location *(i, j)*, and T stands for the target class, i.e. the defect class δ in our case. Hence, equation (1) combines top-down information with bottom-up saliency to find the pointwise mutual information between the feature and the target. Using this formulation, the saliency at a given image point decreases as the probability of feature value f_{ij} gets higher, and increases as the probability of feature value f_{ij} for the defect class δ increases.

4.2 Contrast-based Saliency

As said before, we consider defects as rare phenomena that catch the visual attention of the observer during visual inspection. Following this idea, we describe defects by means of features typically used in cognitive models to predict human eye fixations. To this end, we make use of one of the most influential saliency computational models based on contrast, described in *Itti et al. (1998)*. In this model, the contrast levels in intensity, colour and orientation are computed as centre-surround differences between fine and coarse scales over image pyramids of up to 7 levels; that is to say, the difference between each pixel on a fine (or centre) scale c and its corresponding pixel in a coarse (or surrounding) scale s is calculated as $\mathrm{M}(\mathrm{c}, \mathrm{s}) = |\mathrm{M}(\mathrm{c}) \otimes \mathrm{M}(s)|$, where $\otimes$ is the centre-surround operator, $c \in \{1, 2, 3\}$ and $s = c + \lambda$, with $\lambda \in \{3, 4\}$. Given an RGB colour image, this process is performed over: ($\oplus$ denotes the across-scale addition operator)

- the intensity channel $I = (r + g + b)/3$, with r, g and b as the original red, green and blue channels, to build the *intensity conspicuity map* $IM = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} N\big(I(c,s)\big)$;
- the colour channels RG and BY defined as $RG = R - G$ and $BY = B - Y$, with $R = r - (g + b)/2$ for red, $G = g - (r + b)/2$ for green, $B = b - (r + g)/2$ for blue and $Y = (r + g)/2 - |r - g|/2 - b$ for yellow[2], to build the *colour conspicuity map* $CM = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} N(RG(c,\ s)) + N(BY(c,s))$; and
- the orientation channels $O(\theta)$, calculated by convolution between channel I and Gabor filters at orientations 0º, 45º, 90º and 135º, to build the *orientation conspicuity map* $OM = \sum_{\theta \in \{0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}\}} N\Big(\oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} N\big(O(c,s,\theta)\big)\Big)$.

The map normalization operator $N(*)$ highlights saliency peaks in maps where a small number of strong peaks of activity (conspicuous locations) are present, while globally suppressing peaks when numerous comparable peak responses are present. To this end: (1) the map is normalized to a fixed range, (2) the global maximum M is found, (3) the local maxima average m is determined, and (4) the map is multiplied by $(M - m)^2$.

Finally, the three conspicuity maps are normalized and summed into the final output:

$$\Sigma_{ij} = \frac{1}{3}(N(IM) + N(CM) + N(OM)) \tag{2}$$

4.3 Performance Assessment

Figure 8(a) shows *probability density functions* (PDFs) for contrast, i.e. $p(F = contrast_{ij})$, and contrast conditioned on the presence of defects, i.e. $p(F = contrast_{ij} | T = \delta)$, both determined by means of the *Parzen windows* method, *Theodoridis and Koutroumbas (2009)*, and a training image set comprising surfaces and structures containing cracks, coating breakdown and corrosion.

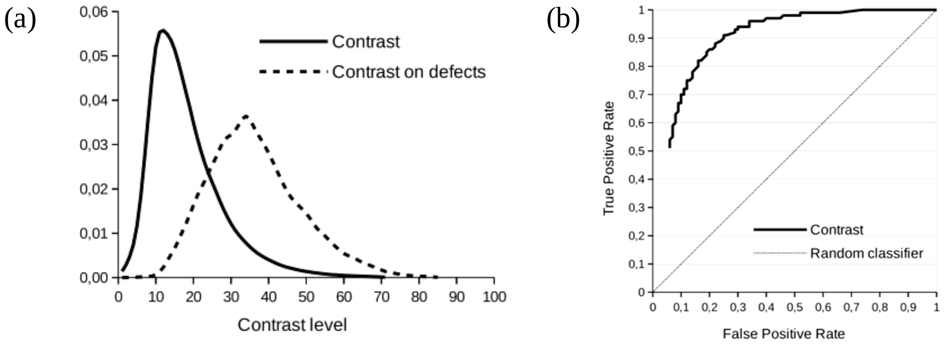


Fig. 8: (a) PDFs for contrast. (b) ROC curve for the defect detector [AUC = 0.88].

Figure 9 shows detection results for a number of images after the training performed. Moreover, we have evaluated the detection approach using *leave-one-out cross-validation*, *Duda et al. (2000)*: one image is selected from the dataset, while the rest is used to obtain the feature PDFs that make up the defect detector (training step), the selected image is next used to validate the detector, and the process

[2] negative values are set to zero for all channels

is repeated for each image in the dataset. Global performance is shown in Fig. 8(b) in the form of a ROC curve relating *true positive rate* (TPR) and *false positive rate* (FPR). (FPR, TPR) points result from thresholding the defect maps at different contrast levels and comparing the resulting binary image with a ground truth. The *area under the curve* (AUC) metric, *Fawcett (2006)*, for this detector was assessed as 0.88, which is quite above the performance of a random classifier.

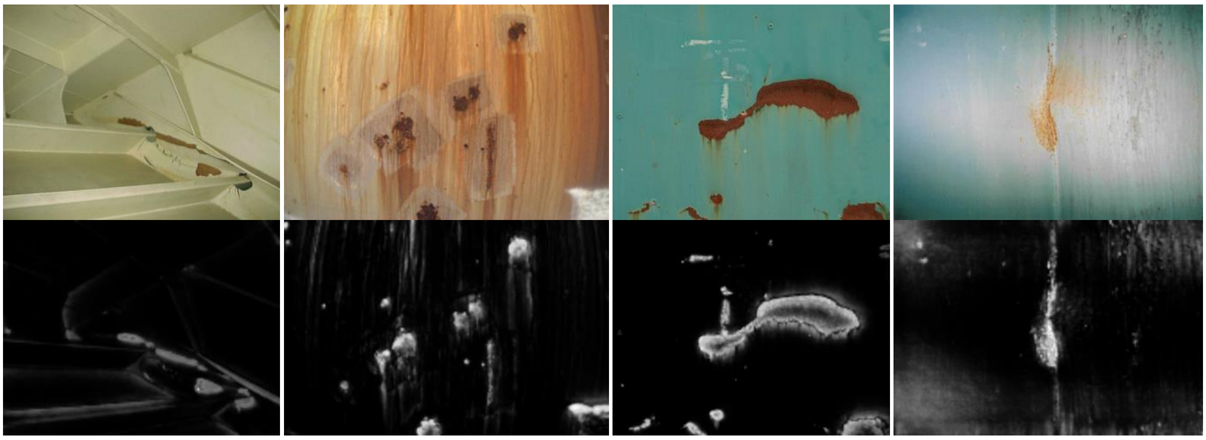


Fig. 9: Real images containing defects and corresponding defect maps. (*Whiter* means *more salient*.)

5. Data transfer

The robotic platforms can feed a number of different data streams depending on the number of sensors and image capturing devices used. Every connected data acquisition platform can output different data collections for further processing. However, the internal data structures for the data acquisition operation may vary widely. It can be expected that in future production applications a considerable range of different robotics-based data acquisition systems will be available, offering a confusing number data formats to work with. To be able to deliver this data efficiently and reliably, a commonly established data transfer method seems necessary.

The purpose of the Hull Condition Monitoring (HCM) data exchange standard, *HCM (2016)*, is to provide such a data transfer model and format for easy and yet powerful exchange of thickness measurement results from inspections. It is therefore an important means for capturing information needed to analyse and assess the structural status of a vessel. HCM defines an XML schema based data model that is originally influenced by ISO 10303-218, *ISO (2004)*, to cover description of structure models but adds a range of entities to handle inspection related data. It therefore defines a focussed and compact model suitable for description of ship structures as required for inspection purposes, e.g. as defined by *IACS (2006)*, *IACS (2009)*. Based on this, inspection results are captured by means of campaign data that can hold all relevant details about the findings. Most importantly, HCM also provides an extension mechanism which supports transport of additional data for use case specific contexts. This mechanism has been used in our implementation work to satisfy additional requirements demanded by automated inspection activities.

As illustrated above, common inspection tasks do not only involve thickness measurement readings, but may also include data sources such as ultrasonic readings or visual inspection imaging and image analysis. For automated visual inspections image and video data is essential, but after post-processing various kinds of condition assessments will also be available and need to be communicated. To satisfy these demands, three extensions for HCM have been developed:

- `VisualInspectionCampaign` – allows the transfer of imaging data directly linked to the HCM structure model
- `HullSurvey` – captures the condition assessment data such as identified cracks and common types of deformations

- CoatingCondition – provides a model for describing the coating and corrosion condition encountered during an inspection

The following excerpt from an exchange unit utilising the HullSurvey extension demonstrates the application for describing the geometry of identified crack geometry and the link to the captured image which has been used for this assessment:

```
<Extension id="hs1" status="in progress" xsi:type="HullSurvey">
   <SurveySession id="session-1" status="in progress">
      <Defects>
         <Crack length="0.16539" name=" session-1.defect1">
            <Shape>
               <Vertex x="0.91963" y="0.30105" z="1.601"/>
               <Vertex x="0.92056" y="0.27804" z="1.601"/>
               <Vertex x="0.92056" y="0.27804" z="1.4372"/>
               <Vertex x="0.91963" y="0.30105" z="1.4372"/>
            </Shape>
            <Attachments>
               <DocumentReference refId="doc0"/>
            </Attachments>
         </Crack>
   …
  </SurveySession>
   …
</Extension>
```

6. Data management and processing

During ship inspections, surveys and monitoring activities using automated devices results in a considerable amount of “raw” data being collected. This data has to be processed and stored to capture and extract as much relevant information as possible. Automation allows inspecting an increased quantity of sampling locations during available time slots using various sensors more or less concurrently, thus additionally amplifying this demand. Data volume is further multiplied as inspection and monitoring is supposed to occur more frequently during the full operational life time of the vessel. Previously acquired data must be made available for further (re-)processing and (re-)assessment. Support functions for visual inspection by human operators including thorough evaluation performed by experts or application of advanced analysis using distinct (and rapidly developing) collections of algorithms (some of which will inevitably require access to sufficiently powerful computing resources) depend on easy access and efficient querying and retrieval. With such large sets of data, a foundation is established to apply new or refined analytical methods, which are essential for automating (part of) the assessment activities (e.g. defect pre-identification) and to prepare the data in the best possible way for final judgement by human experts.

Despite the fact that inspection devices carry considerable embedded processing capabilities, immediate post-processing capacity is nevertheless limited and thus further substantial post-processing occurs after actual acquisition but before finally storing the data for long-term reference. During this process the objective is to capture as much raw information as possible during acquisition and then to reduce the data volume using closely linked post-processing steps without losing any valuable information.

Such post-processing steps may include:

- Filtering (e.g. removing noise, detecting out-of-bounds recordings, and performing other quality checks)
- Data compression (e.g. removing redundancy or combining many samples into a compact storage format)
- Data fusion (e.g. combining image data, structure model data, thickness measurements to create part status information)
- Data attribution (e.g. linking image information with location and orientation data)

- Information generation (e.g. defect identification from image processing)

As shown in the system architecture (Fig. 10), processing is distributed to different system components: operations that depend on raw data and/or are specific to individual robotic platforms are implemented as part of portable data acquisition systems, which will be closely located and linked to the robotic units. The output from these systems is suitable for compact storage and transfer using the extended HCM and VoyageLog data exchange formats. These data can be asynchronously processed by the information management system (IMS) using appropriate adapters which prepare and convert the HCM and VoyageLog data for persistent data storage. Once it is stored, the data will be available for any type of inquiry, reporting and analytic operation.

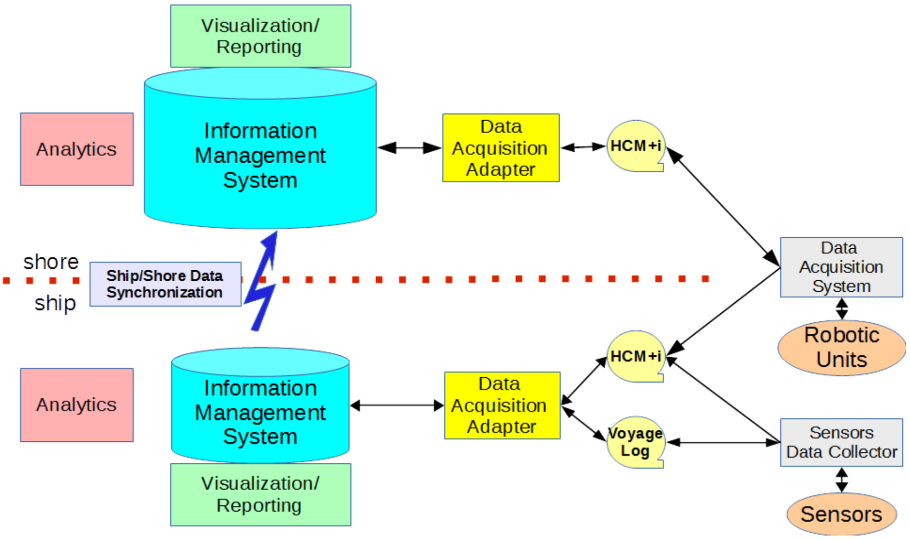


Fig. 10: System architecture for information management system

Within the information management system, data entities such as the following are being stored for processing and evaluation:

- Vessel details: general particulars, class notation, time line
- Compartmentation data
- Hull structure details down to physical part level
- Analytical models: finite element models, hydrodynamic models, reliability models
- Life cycle history: voyage tracking, loading conditions, inspection events
- Repairs: activity records and replaced parts
- Inspection data:
    - Bulk data
        - Thickness measurements
        - Images, videos
    - Post-processed data
        - Coating condition assessment
        - Corrosion assessment
        - Structure defects: cracks, buckling, general deformations
- Monitoring data: time series data for strain gauges, accelerometers, vibration and shock sensors
- Analytics results

The information management system has been implemented based on the Topgallant Information Server (TGIS), *AES (2015)*, which provides specific support for complex structured engineering type data and can operate on top of different storage engine platforms.

Considering the wide range of data management requirements in this application, it was determined that scalability of the information management system is a critical design parameter. This has been achieved by employing three different storage platforms for different usage scenarios, all of which are operated via the same single data management API available in TGIS:

- A small footprint embedded single-process engine that utilizes a configurable in-memory database cache to ensure high performance on smaller platforms and requires minimal configuration but provides full persistence and transaction capabilities. This engine allows prototyping as well as operating in moderately sized and portable support tools, e.g. as they may be used on-board vessels.
- A client-server engine that operates as a multi-user networked database layer. It is intended to cover common scenarios of multi-client operation against a storage server. This configuration is suitable for use in larger on-board configurations. However it is primarily targeting mid-range on-shore configurations.
- A distributed, highly scalable engine implementing a NOSQL storage paradigm. This engine provides "big data" processing power, capable of handling multi-Terabyte data volumes with selectable data redundancy and availability levels. Since this platform can involve a large amount of hardware (including options to operate as a cloud-based system), more advanced configuration considerations apply. It is expected that with continued advances in commodity hardware and virtualisation techniques this may well become the standard operational platform.

For complex analytics and continued (re-)processing of the data, the system supports the dynamic integration on externally provided calculation modules, as described in *Koch et al. (2015)*. Using this method, a variety of processing capabilities can be provided ranging from visualisation (e.g. image rendering) and reporting (e.g. graphs) to analytics and reporting.

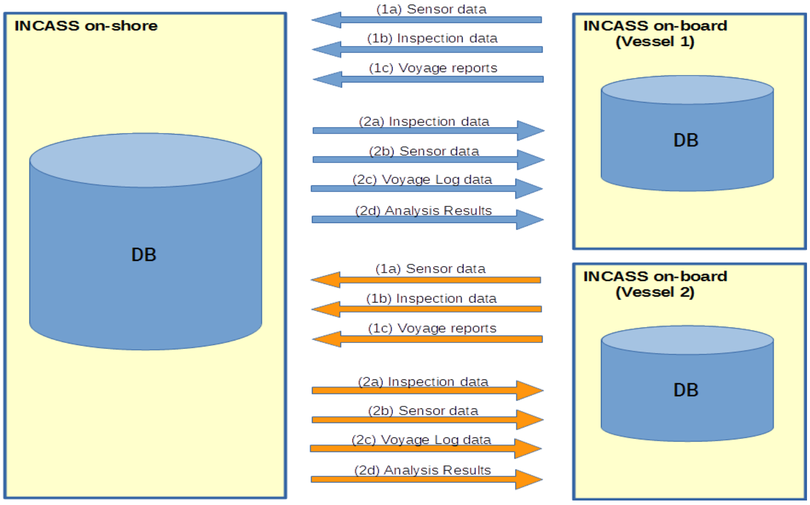


Fig 11: Ship-Shore communication data streams

Finally, some attention had to be given to the communication requirements of the overall system configuration (Fig. 11). Since the data acquisition occurs almost exclusively on board the vessel and quite a range of useful functions can be provided on board based on those data, it seems logical to have an on-board installation of the information management components which can be used as a connection point by the robotics data acquisition systems. At the same time, ship operators will want to have access to the same data on shore, and the known data about a particular vessel should always – at least eventually – be synchronised between these installations. This requires a reliable data communication path between on-board and on-shore components.

Constraints for such a communication path are well known: while technology advances rapidly in terms of available ship-shore bandwidth and reliability, it will – for a foreseeable future – lag behind pure on-shore solutions in both these parameters, and this must be taken into account. For this application, a messaging sub-system based on the MQTT protocol has been put in place, what ensures a fault-tolerant and reliable operation (Fig. 12), *OASIS (2014)*. By including an independently operating message broker, which supports guaranteed message delivery, data synchronisation can be accomplished via optimised combined message publisher/subscriber end points in the IMS installations.

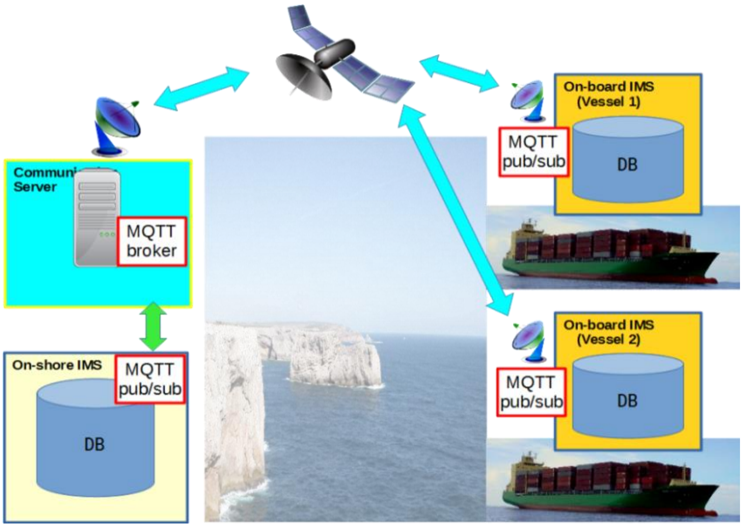


Fig 12: Reliable communication via MQTT messaging

7. Conclusions and Outlook

Due to the high cost of inspections of ship-board structures, given the recent advances in robotics and automation, better methods for these tasks seem feasible. However, the operating environment as well as the variety and complexity of the inspection processes pose a challenge. In this paper, we have discussed some developments that demonstrate how different platforms and approaches may be used and how they could operate effectively in combination and with support from appropriate software components. Results indicate that – at least for the foreseeable future – a well-tuned combination of different robotic tools may be a very promising solution. Another important aspect (just about to be fully realised) is the fact that automated data acquisition needs to be complemented by adequate post-processing and data management capabilities.

Remaining challenges that need to be addressed in the near future comprise the incorporation of further developments on industry standard data formats for inspection data transfer, as well as the development of enhanced software for embedded large volume data acquisition. Further assessment of the robotic platforms performance will also take place in the near future in order to identify limitations and ways of improvement in all hardware and software components.

Acknowledgements

Work described in this paper has been supported with funding from the 7th Framework Programme of the European Commission under Grant Agreement FP7- SST.2013.4-2, CP-605200 (INCASS - Inspection Capabilities for Enhanced Ship Safety). It reflects only the authors’ views and European Union is not liable for any use that may be made of the information contained therein. The authors also wish to thank the partners of the INCASS Project for their input and support.

References

AES (2015), Topgallant Information Server, http://www.atlantec-es.com/topgallant-product-is.html.

ARKIN, R. (1998), *Behavior-based Robotics*, MIT press.

BIBULI, M. et al. (2012), *MARC: Magnetic Autonomous Robotic Crawler Development and Exploitation in the MINOAS Project*, Int. Conf. Computer and IT Appl. Maritime Ind., pp. 62-75.

CHENG, G.; ZELINSKY, A. (2001), *Supervised Autonomy: A Framework for Human-Robot Systems Development*, Autonomous Robots 10/3, pp. 251–266.

DUDA, R.; HART, P.; STORK, D. (2000), *Pattern Classification*, Wiley-Interscience.

EICH, M; VOEGELE, T. (2011), *Design and control of a lightweight magnetic climbing robot for vessel inspection*, IEEE Mediterranean Conf. Control & Automation, pp. 1200-1205.

FAWCETT, T. (2006), *An introduction to ROC analysis*, Pattern Recogn. Letters 27/8, pp. 861-874.

GURDAN, D. et al (2007), *Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz*, IEEE Intl. Conf. Robotics & Automation, pp. 361–366.

HCM (2016), OpenHCM Consortium: Sourceforge repository, http://sourceforge.net/projects/openhcmstandard/files/Schema xsd files and corresponding descriptions/.

IACS (2006), UR Z *Survey and Certification. Requirements concerning Survey and Certification*, Intl. Association of Classification Societies, London, http://www.iacs.org.uk

IACS (2009), PR19 *Procedural Requirement for Thickness Measurements*, Intl. Association of Classification Societies, London, http://www.iacs.org.uk

INCASS (2013), *Description of Work*, FP7-TRANSPORT-2013-MOVE-1; SST.2013.4-2. Inspection capabilities for enhanced ship safety (GA 605200).ISO (2004), ISO 10303-218:2004: *Industrial automation systems and integration -- Product data representation and exchange -- Part 218: Application protocol: Ship structures*, Intl. Organization for Standardization, Geneva.

ITTI, L.; KOCH, C.; NIEBUR, E. (1998), *A model of saliency-based visual attention for rapid scene analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence 20/11, pp. 1254-1259.

JUNG, M.; SCHMIDT, D.; BERNS. K. (2010) *Behavior-Based Obstacle Detection and Avoidance System for the Omnidirectional Wall-Climbing Robot CROMSCI*, Emerging Trends in Mobile Robotics, pp. 73–80.

KOCH, T.; SMITH, M; TANNEBERGER, K. (2015), *Improving Machinery & Equipment Life Cycle Management Processes*, Int. Conf. Computer and IT Appl. Maritime Ind., pp. 101-115.

MINOAS (2009), *Description of Work*, FP7-SST-RTD-1; SST.2008.5.2.1. Marine INspection rObotic Assistant System (GA 233715).

OASIS (2014), Banks, A.; Gupta, R. (eds.), MQTT Version 3.1.1, OASIS Standard, 29 October 2014, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

TACHE, F. et al. (2010); MagneBike: Compact Magnetic Wheeled Robot for Power Plant Inspection, Intl. Conf. Applied Robotics for the Power Industry, pp. 1-2.THEODORIDIS, S.; KOUTROUMBAS, K. (2008), *Pattern Recognition*, Academic Press.

ZHANG, L. et al. (2008), *SUN: A Bayesian framework for saliency using natural statistics*, Journal of Vision 8/7, pp. 1-20.