Robotics and Autonomous Systems 131 (2020) 103589

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Monocular vision-based gripping of objects

Bent Oddvar Arnesen Haugaløkken *, Martin Breivik Skaldebø, Ingrid Schjølberg

*Department of Marine Technology, NTNU Trondheim, Norway*

ARTICLE INFO

*Article history:*
Received 12 November 2019
Received in revised form 28 May 2020
Accepted 3 June 2020
Available online 12 June 2020

*Keywords:*
Underwater robotics
Object detection
Autonomy
Dynamic positioning
Manipulator

ABSTRACT

Optics-based systems may provide high spatial and temporal resolution for close range object detection in underwater environments. By using a monocular camera on a low cost underwater vehicle manipulator system, objects can be tracked by the vehicle and handled by the manipulator. In this paper, a monocular camera is used to detect an object of interest through object detection. Spatial features of the object are extracted, and a dynamic positioning system is designed for the underwater vehicle in order for it to maintain a desired position relative to the object. A manipulator mounted under the vehicle is used to retrieve the object through a developed kinematic control system. Experimental tests verify the proposed methodology. A stability analysis proves asymptotic stability properties for the chosen sliding mode controller and exponential stability for the task error.



## 1. Introduction

The need for subsea inspection, maintenance, and repair (IMR) operations in the ocean industries is high, and is expected to increase further in the coming years. A great deal of IMR operations can be carried out by an underwater vehicle equipped with one or more manipulator arms, often referred to as an underwater vehicle manipulator system (UVMS). This emphasizes the necessity for research and further development of this type of technology. Fully autonomous, semi-autonomous, and tele-operated UVMSs are of interest to the maritime industry, as they may increase safety and reduce operational costs significantly [1].

In many cases, it is desired to combine tele-operation with semi-autonomy, especially where fully autonomous robots seem unreliable or are too costly to be developed or utilized efficiently. The vehicle should provide visual and sensory feedback to the operator, who may assess the situation, make decisions, and execute high level tasks, while the robot carries out the lower level tasks. Whether the system is fully autonomous, semi-autonomous, or tele-operated, several functionalities need to be developed. One of the most important sensors for situational awareness and understanding how to perform certain operations is the camera system, while various types of acoustic sonars also has seen considerable use [2–4]. The camera system quality and software become increasingly more important as we move towards higher levels of autonomy. Recent technological advances within camera systems and image processing techniques prove that the camera is the preferred sensor type for short range navigation, as they deliver information with high spatial and temporal resolution [3,5]. Low cost cameras and state-of-the-art graphical processing units have now made its way into the commercial market. Together with several open source object detection frameworks, they provide quick and reliable methods for developing and performing object detection, classification, perception, and situational awareness in underwater operations. Simultaneously, developments within commercial underwater vehicle products have surfaced, such as the BlueROV2 by Blue Robotics [6], which simplifies integration of optics-based detection with underwater vehicles. When low cost underwater vehicles, optics-based solutions, and manipulators are combined, the barrier for conducting research and experimental testing on such systems is lowered. This may have important consequences for how IMR operations within the ocean industries will be performed in the future.

A UVMS consists of the vehicle body and one or more manipulator arms. Usually, it has a tether for power and/or topside communication, and is tele-operated while featuring some basic autonomous-like functionality, e.g., automatic depth or heading control. The vehicle has 6 degrees of freedom (6 DOF), while the manipulator arm has $n$ DOF, depending on the number of joints that can move. This means that the UVMS has $6 + n$ DOF and is therefore a kinematically redundant system, which means that the system has more DOF than needed to accomplish a single task. Typically, such a kinematically redundant system is managed using kinematic control [7]. This control method utilizes the kinematic relations of the system through its Jacobian, which enables control of position, velocity, and acceleration of the manipulator to some desired state based on, e.g., velocity references and a low level controller. Kinematic control paired with

* Corresponding author.
*E-mail addresses:* bent.o.arnesen@ntnu.no (B.O.A. Haugaløkken), martin.b.skaldebo@ntnu.no (M.B. Skaldebø), ingrid.schjolberg@ntnu.no (I. Schjølberg).

https://doi.org/10.1016/j.robot.2020.103589

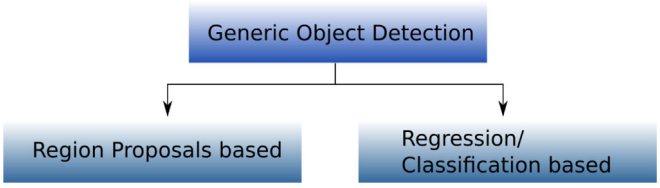


**Fig. 1.** Object detection methodologies.

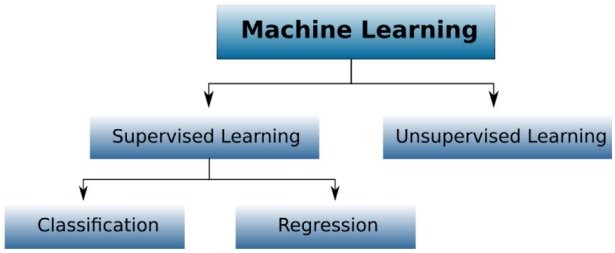


**Fig. 2.** Machine learning methodologies.

object detection for the UVMS is one way to enable autonomous gripping functionality.

Currently, underwater object detection based on optics is a very interesting field due to its wide array of applications, e.g. within research, various subsea industries, and for hobbyists. The most popular vision based techniques depend on monocular (2D) or stereo (3D) vision, while some 2.5D methods have been proposed as well, which mainly involve projecting 2D images to reconstruct 3D environment features [8]. Object detection methods range from edge [9,10] or color [11] detection to smarter solutions such as optical flow [12] and machine learning approaches, e.g. classification [13], salient feature detection [14], and object detection [15,16]. Within machine learning there are several detection methodologies, which can be distinguished into two main categories — Region proposal networks (RPNs) and regression/classification (see Fig. 1). These two methodologies both arrive from the supervised learning branch of machine learning (see Fig. 2), where known datasets are used for training in order to make predictions in new datasets, i.e. the goal is to learn the mapping from an input $x$ to an output $y$. The governing difference between these two categories is that classification approaches try to learn the mapping to a discrete or categorical output (e.g. whether an object belongs to a certain class), while regression aims to learn the mapping to a continuous or numerical output. A collection of the most essential methods for RPN and regression/classification approaches can be found in [13].

The first category, namely RPN-based methods, follows the traditional object detection procedure where region proposals are identified and classified into object categories. Such methods behave similarly to the methodologies that are used by the human brain, utilizing an initial scan of the entire scene before it is separated into regions of interest. Some of the most popular methods for the RPN are region proposals convolution neural network (R-CNN), Fast R-CNN, and Faster R-CNN. According to [17], both R-CNN and Fast R-CNN use a selective search algorithm, which is a time-consuming process. This may render the CNN methods a bit too slow for real-time object detection tasks, but it depends heavily on the available hardware. The difficulty of achieving real-time object detection with R-CNN and Fast R-CNN with the available hardware (also with respect to cost) was the main motivation for further research on RPN-based methods, that eventually led to the development of Faster R-CNN. Faster R-CNN instead uses a separate network to predict the region proposals and yields a significantly faster network, which makes it better suited for real-time object detection compared to R-CNN and Fast R-CNN.

The second machine learning category contains classification/regression approaches, where some of the methods are Single Shot Detector (SSD) and You Only Look Once (YOLO) versions 1, 2, and 3. SSD uses an approach based on classification/regression that does not necessitate object proposals and encapsulates all computation in a single network [18]. This method is fast, reliable, and achieves accurate object detection in real-time. The YOLO object detection method was first presented in [19], where object detection was conducted as a regression problem instead of classification. In this article YOLOv3 [20] has been used, which combines three neural networks into a network with 53 convolution layers called Darknet-53. This network predicts bounding boxes and class probabilities, considering and evaluating the whole image once. The process that follows is a bounding box prediction using dimension clusters as the anchor boxes, where four coordinates are calculated for each bounding box. Training is performed by summing the squared error loss, and the class prediction loss is calculated using binary cross-entropy loss. For each bounding box an objectness score is calculated, which is a measure that describes the detectors ability to identify the locations and classes of objects [21]. The system predicts the classes that may be contained within each box through multi-label classification. This multi-label classification method is chosen instead of a softmax function, which enables detection of objects with overlapping bounding boxes. Overall, YOLOv3 achieves an average precision (AP) score that is close to other SSD methods, but is approximately three times as fast [19].

Kinematic control of manipulator arms has been researched thoroughly [22], but an UVMS allows for additional manipulation operations due to the mobile base of the manipulator, and some of the largest contributions are given in the following. Most of the work within the UVMS operation community has been conducted in large projects, as equipment tend to be expensive and the integration of this with smart software solutions require interdisciplinary collaboration. One of the first autonomous manipulation operations was carried out by the SAUVIM project [23]. The work presented a recovery operation where the mission was to grasp a spherical object using a UVMS with a camera mounted on the arm's wrist. The object was detected by combining image filtering to reduce noise, Canny edge and color detection, and a method for circle detection. The TRIDENT project [24] demonstrated an object recovery operation using a stereo camera solution and task-priority with activation functions for managing several tasks at once, exploiting the redundancy of the system. In the MARIS project [25] a pipe grasping mission was conducted through three campaigns for day and night light conditions in calm waters. All of the experiments were conducted in pools, where the use of a Doppler velocity logger improved vehicle and end-effector control, leading to an increase in the grasping success rate of the object from around 30% to 70%. A camera in the gripper and an optoelectronic sensor in the wrist were used to determine when to grasp the object. They also integrated a task-priority framework with activation functions that defined the current active task, and reported a considerable improvement in robustness compared to the TRIDENT project. The GIRONA 500 I-AUV (Intervention autonomous underwater vehicle) is one of the first lightweight (approximately 150 kg) AUVs with intervention capabilities. Reportedly, it has performed intervention panel manipulation using visual servoing and panel detection, valve turning and connector plugging/unplugging, docking, optical surveying, target tracking, and multiple vehicle cooperation for large object transportation, to mention some if its accomplishments [26,27]. Furthermore, additional autonomous features are

**Table 1**
BlueROV2 and SeaArm specifications.

| Parameter | Value |
|---|---|
| BlueROV2 | |
| L × H × W | 457 × 254 × 575 [mm] |
| Weight in air | 11.5 [kg] |
| Thrusters | T-200 |
| Battery | 14.8 [V], 10 [Ah] |
| Depth rating | 100 [m] |
| Camera | Raspberry Pi Camera V2.1 |
| On-board Computer | Raspberry Pi 3B and Navio2 |
| SeaArm | |
| Degrees of freedom | 3 |
| Weight (air) | 2.4 [kg] |
| Weight (submerged) | 0 [kg] |
| Max reach (end-effector) | 580.7 [mm] |
| Servos | 5 × electric servos |
| Stall torque at 14.8 V | 10 [Nm] |
| Depth rating | 100 [m] |

still heavily researched for this vehicle. One of the most recent works in autonomous solutions for UVMS intervention operations is the DexROV project [28], which focuses on reducing the gap between autonomy and tele-operation when controlling ROVs in underwater manipulation operations. This project has utilized several technologies for fine manipulation of objects in the water column, such as stereo camera solutions, inertial navigation system, set based task priority control, obstacle avoidance, and a high-end gripper with force/contact sensors [29,30].

This paper studies and develops grasping of a known object using a monocular camera through machine learning and a small UVMS. One of the main goals is to provide an effective solution for object retrieval mission for a small, low cost UVMS ($\leq$ $ 20.000 USD). This paper follows from the work of [31], which presented a large image dataset of the object of interest, an automatic labeling procedure of the images, training of the model and the object detection procedure. Furthermore, this paper utilizes the state-of-the-art object detection framework YOLOv3 [20] to find a known object. The object is detected in a laboratory basin with a monocular camera inside a BlueROV2 underwater vehicle [6], and the spatial features of the object relative to the vehicle are estimated. The object is then grasped with the SeaArm manipulator arm [32,33] that is mounted on the vehicle. This procedure removes the need for a local positioning system and works seamlessly with tele-operations, while still incorporating autonomous functionality. A dynamic positioning (DP) system is designed to maintain the vehicle's desired position and velocity relative to the object, and a kinematic control system is developed for the manipulator in order to retrieve the object.

The main contributions of this paper are summarized below:

(1) Design of a navigation, guidance, and control system for the vehicle to maintain a desired position relative to an object detected through monocular vision and machine learning
(2) A stability proof that ensures exponential convergence of the task errors and asymptotic convergence to the sliding mode controller's sliding surface
(3) Experimental testing that proves the effectiveness of the proposed solution for grasping the object with a low cost underwater vehicle manipulator system

The paper is structured as follows. Section 2 provides brief specifications for the underwater vehicle, the manipulator arm, the camera system, and the object detection system used in the experimental work. Section 3 presents the kinematic control system, Section 4 describes task definitions and the control system and Section 5 presents the stability analysis for the sliding mode controller and the task error. The experimental testing procedure and results are presented in Section 6 and a discussion of the proposed methodology and experimental findings is given in Section 7. Conclusions and suggestions for further work are provided in Section 8.

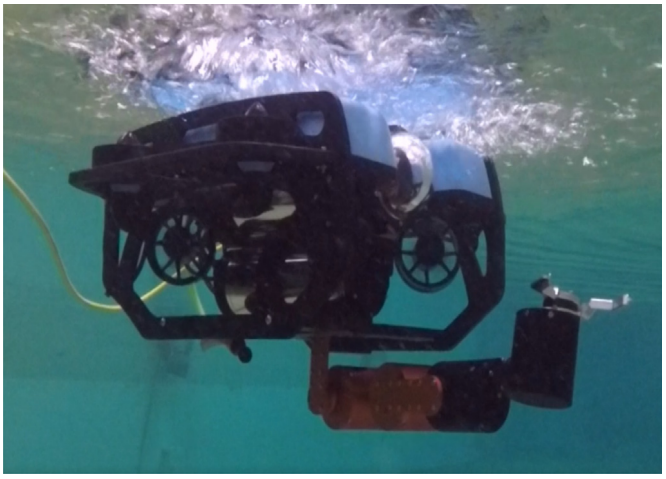

**Fig. 3.** The BlueROV2 underwater vehicle equipped with the SeaArm manipulator arm.

**Table 2**
Denavit–Hartenberg parameters.

| i | $\alpha_{i-1}$ [rad] | $a_{i-1}$ [mm] | $d_i$ [mm] | $\theta_i$ [rad] |
|---|---|---|---|---|
| 1 | 0 | 0 | 55.3 | $\theta_1$ |
| 2 | $-\pi/2$ | 0 | 0 | $-\pi/2$ |
| 3 | 0 | 142.4 | 42.1 | $\theta_2 - \pi/2$ |
| 4 | $\pi/2$ | 142.4 | 0 | $-\pi/2$ |
| 5 | $-\pi/2$ | 0 | 13 | $\theta_3 + \pi$ |
| 6 | $\pi/2$ | 0 | 42.1 | $\pi/2$ |
| 7 | 0 | 0 | −139.6 | $\theta_4 - \pi/2$ |
| 8 | 0 | 101 | −59.6 | 0 |

## 2. Specifications

This section briefly describes the specifications of the UVMS, the camera system, and the approach for detecting the object of interest. The experiments have been conducted in the Marine Cybernetics Laboratory (MC-lab) at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway [34].

### 2.1. The BlueROV2 and SeaArm manipulator arm

The BlueROV2 is a 6 DOF, slightly positively buoyant, small-sized ROV, and the SeaArm is a fully electric and neutrally buoyant 3 DOF manipulator arm. In this work, the SeaArm is mounted on the bottom left side of the vehicle as can be seen in Fig. 3. The main features of the UVMS can be found in Table 1, and the Denavit–Hartenberg (DH) parameters of the manipulator arm are presented in Table 2. SeaArm has an in-built velocity controller based on damping least squares for singularity avoidance, and uses reference positions to estimate the joint velocities.

### 2.2. The camera system and computer vision framework

The camera used in this paper is mounted inside the BlueROV2 and is a standard Raspberry Pi Camera Module V2.1. Table 3 shows some of the most important specifications related to the camera and the object's position within the image. The machine learning approach used for object detection is presented

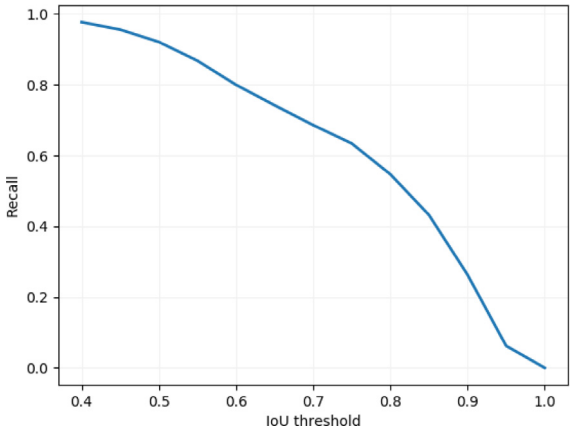


**Fig. 4.** Recall vs. IoU threshold values.

**Table 3**
Specifications of the Raspberry Pi Camera V2.1.

| Parameter | Definition | Value |
|---|---|---|
| $FOV_w$ | Field of view — width (horizontal) | 62.6 [deg] |
| $FOV_h$ | Field of view — height (vertical) | 48.8 [deg] |
| $C_w$ | Total pixel width of camera frame | 640 [px] |
| $C_h$ | Total pixel height of camera frame | 480 [px] |
| $P_{obj,w}$ | Object pixel position in width direction | 0 - 640 [px] |
| $P_{obj,h}$ | Object pixel position in height direction | 0 - 480 [px] |

in [31] and is briefly summarized here. The method utilizes an image dataset of 7071 images retrieved by splitting long video sequences into images. An automatic labeling procedure was used to find the object within the image based on color contours (color detection). An interactive user interface was used to quickly verify each labeled image and to remove incorrect labels. The object of interest was characterized by a cylindrical shape and a distinct orange color. All of the images were converted from the RGB space to hue-saturation-brightness values to create a color map that was easier to analyze and to make sure the object was apparent in the images.

The object detector that has been used here is YOLOv3 [20]. The algorithm in YOLOv3 was trained for 5000 iterations with a batch size of 64 and subdivision set to 16. One full batch is considered between every weights update in the neural network. The model was built and validated using the Darknet framework, which provided a model that has achieved an average precision (AP) of 97.7% in the pool [31]. This AP was provided by the built-in validation method in Darknet where the intersection over union (IoU) threshold value was set to 0.5. An IoU threshold value sets a boundary for successful detection by requiring minimum overlapping ratios between the suggested bounding box from the detector and the ground truth. The AP value is very high, and should be considered with some constraints. The algorithm was validated on a subset of the complete dataset, which means that the validation images embodies almost identical features as the training images. The high value may also imply an over-fitted model. However, the object detection system is able to detect the object of interest in the pool accurately. In addition to AP, the recall rate was recorded and with different threshold values for IoU. The recorded recall rates for the various IoU threshold values can be seen in Fig. 4.

The detector was set to run on an HP Laptop with Intel Core i7-7700HQ with 16GB RAM and an NVIDIA GeForce GTX 1060 (6GB GDDR5 dedicated) GPU. The system managed to analyze 30–40 frames per second for 1080 × 720 resolution video, which provided real-time compatibility.

## 3. Equations of motion

The model of the UVMS applied here has been based on [35], where the states of the UVMS base are described by the position $\boldsymbol{\eta} = [\boldsymbol{p}^T \ \boldsymbol{\theta}^T]^T$ and velocity $\boldsymbol{\nu} = [\boldsymbol{v}^T \ \boldsymbol{\omega}^T]^T$ vectors. The vector $\boldsymbol{p} = [x, y, z]^T$ is the position and $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$ is the orientation in Euler angles of the camera frame expressed relative to the object, i.e. the object-relative (OR) frame. Furthermore, $\boldsymbol{v} = [u, v, w]^T$ represents the linear velocity and $\boldsymbol{\omega} = [p, q, r]^T$ denotes the angular velocity of the camera frame expressed in the OR frame. The states of the manipulator are described by the joint angles $\boldsymbol{q} = [q_1, q_2, q_3]^T$ and joint angular rates $\dot{\boldsymbol{q}} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$. The pose (position and orientation) of the manipulator's end-effector is defined relative to the camera frame, as $\boldsymbol{\eta}_{ee} = [\boldsymbol{p}_{ee}^T, \boldsymbol{\Theta}_{ee}^T]^T$, where $\boldsymbol{p}_{ee} = [x_{ee}, y_{ee}, z_{ee}]^T$ and $\boldsymbol{\Theta}_{ee} = [\phi_{ee}, \theta_{ee}, \psi_{ee}]^T$ is the end-effector position and orientation, respectively. The combined system is then written as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_R(\boldsymbol{\eta})\boldsymbol{\nu} \tag{1}$$

$$\dot{\boldsymbol{\eta}}_{ee} = \boldsymbol{J}_e(\boldsymbol{q}, \boldsymbol{\eta})\boldsymbol{\zeta}\ , \tag{2}$$

where $\boldsymbol{J}_e = \begin{bmatrix}\boldsymbol{J}_1(\boldsymbol{q}) & \boldsymbol{J}_2(\boldsymbol{\eta})\end{bmatrix}$, and $\boldsymbol{\zeta} = [\dot{\boldsymbol{q}}^T \ \boldsymbol{\nu}^T]^T$ represents the velocity of both the UVMS body and the manipulator. $\boldsymbol{J}_R$ denotes the Jacobian for the UVMS and $\boldsymbol{J}_e$ is the Jacobian relating the end-effector time derivative to $\boldsymbol{\zeta}$, allowing for velocity based control of the end-effector position through inverse kinematics. The position data for the vehicle is only valid when the object is detected because of the nature of monocular cameras and the lack of an external positioning system in the proposed setup. The position of the UVMS is defined as

$$x = S_A \tag{3}$$

$$y = x\sin(\psi) \tag{4}$$

$$z = x\sin(\theta). \tag{5}$$

where $\psi$ is the heading angle and $\theta$ is the pitch angle relative to the object, and where $S_A$ is a scaling function that is used to estimate the camera's distance to the object. Data used to generate the scaling function was gathered manually by measuring and relating the camera's distance to the object and the corresponding pixel area of the object in the image frame. A piece-wise cubic Hermite interpolation polynomials (PCHIP) function was applied to scale an arbitrary pixel area value to a specific distance based on the gathered data, and a graphical illustration of $S_A$ is presented in Fig. 5. This distance is defined as the vehicle's x-position. The y- and z-position are obtained by exploiting the geometrical relations through (3)–(5) and the heading and pitch angles relative to the center of the object in the image frame, where the latter two are defined as

$$\psi = \frac{FOV_w}{C_w/2} \cdot P_{obj,w} - FOV_w \tag{6}$$

$$\theta = \frac{FOV_h}{C_h/2} \cdot P_{obj,h} - FOV_h\ . \tag{7}$$

In (6)–(7), $FOV_w$ and $FOV_h$ are the camera's field of view (FOV), $P_{obj} = [P_{obj,w},\ P_{obj,h}]$ refers to the pixel position, and $C_w$ and $C_h$ relate to the pixel position that the angles should be measured relative to in width (horizontal) and height (vertical) direction, respectively. With this setup, the heading angle $\psi$ and pitch angle $\theta$ both represent a zero angle in the center of the camera image. By the assumption that angles are small for the majority of the object detection procedure, a standard Kalman filter was implemented to estimate the vehicle's velocity $\boldsymbol{\nu}$ based on the relative position and angle estimates in (3)–(7).

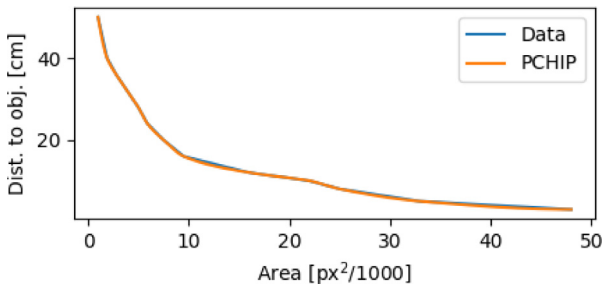


**Fig. 5.** Visual representation of the scaling function. The object area [px$^2$/1000] is plotted along the $x$-axis with the corresponding distance to the object [cm] on the $y$-axis. The original measured data is shown in blue and the area scaling function based on the PCHIP is visualized in orange [31]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## 4. Control system

This section describes the task definition, reference velocity generation, and the control system for the UVMS. The objective is to keep a fixed position and heading angle relative to an object of interest using an underwater vehicle, and then grip the object using the manipulator. It is important to note that the position is only defined during successful detection of the object. A simple Kalman filter is implemented for estimating the vehicle's velocity and a sliding mode controller based on the velocity estimates is responsible for controlling the vehicle. The Kalman filter incorporates the process noise matrix $\boldsymbol{Q} = diag(3^2, 4^2, 5^2)$ and measurement noise matrix $\boldsymbol{R} = diag(0.5, 0.5, 0.5)$, in which the input to the Kalman filter is estimated position and orientation data in [cm] and [deg]. Automatic pitch and roll proportional–integral–derivative controllers stabilize the vehicle in roll and pitch based on data from an internal measurement unit. Consequently, roll and pitch motions are handled by a lower level inertial navigation system, and are not discussed further. Instead, the assumption that the vehicle has a zero roll and pitch angle is employed. The manipulator has an integrated proportional controller based on desired position in task space to control joint velocities and uses damping least squares for avoiding singularities.

### 4.1. Task definition and kinematic control

The guidance system generates appropriate velocity references for the vehicle and desired end-effector positions for the manipulator through pre-defined tasks. Velocity references are derived by a simple technique similar to what was done in [33], while manipulator joint angular rates are computed and tracked by its in-built controller with the task variable as input. In general, an arbitrary task $\chi$ can be defined through a generic variable as

$$\boldsymbol{\sigma}_\chi = \boldsymbol{\sigma}_\chi(\boldsymbol{\eta}, \boldsymbol{q}) \tag{8}$$

and the task-specific Jacobian

$$\dot{\boldsymbol{\sigma}}_\chi = \boldsymbol{J}_\chi(\boldsymbol{\eta})\boldsymbol{\zeta} \ , \tag{9}$$

where the value $\chi$ refer to the task number or the priority of the task in a task priority hierarchy. The tasks considered in this paper are UVMS base x- and z-position and heading $\psi$, in addition to end-effector x-, y- and z-position. The vehicle's position and heading are based on the object's estimated position, which leads to the following task variable and Jacobian

$$\boldsymbol{\sigma}_1 = \begin{bmatrix} x & z & \psi \end{bmatrix}^T \tag{10}$$

$$\boldsymbol{J}_1 = \boldsymbol{I}_{3\times 3} \ , \tag{11}$$

where the simple design of $\boldsymbol{J}_1$ is based on the assumption that orientation angles are small. End-effector control can be described by a task variable as follows

$$\boldsymbol{\sigma}_2 = \begin{bmatrix} x_{ee} & y_{ee} & z_{ee} \end{bmatrix}^T \ , \tag{12}$$

where the task Jacobian $\boldsymbol{J}_2$ for the manipulator is calculated based on the DH-parameters (Table 2) and the homogeneous transformation matrix. The reference velocities are calculated using the pseudo-inverse

$$\boldsymbol{\zeta}_r = \boldsymbol{J}_\chi^\dagger(\boldsymbol{\eta})\dot{\boldsymbol{\sigma}}_{\chi,r} \ . \tag{13}$$

The pseudo-inverse is given as

$$\boldsymbol{J}_\chi^\dagger = \boldsymbol{J}_\chi^T(\boldsymbol{J}_\chi\boldsymbol{J}_\chi^T)^{-1} \ , \tag{14}$$

Note that the notation for $(\boldsymbol{\eta}, \boldsymbol{q})$ is now omitted for enhanced readability. The parameter $\dot{\boldsymbol{\sigma}}_{\chi,r}$ corresponds to the reference task velocity. This is used as a feedback to increase convergence towards the desired position and heading values of the vehicle, and according to [7], it can be chosen as

$$\dot{\boldsymbol{\sigma}}_{\chi,r} = -\boldsymbol{K}_\chi\tilde{\boldsymbol{\sigma}}_\chi \ , \tag{15}$$

where $\boldsymbol{K}_\chi$ is a gain matrix. Furthermore, the task error is chosen as $\tilde{\boldsymbol{\sigma}}_\chi = \boldsymbol{\sigma}_{\chi,d} - \boldsymbol{\sigma}_\chi$, where $\boldsymbol{\sigma}_{\chi,d}$ represents the desired values for task $\chi$. The part of $\boldsymbol{\sigma}_{\chi,d}$ that is concerned with end-effector desired position is rotated from the manipulator base frame to the camera frame, such that the end-effector moves to the object's location. As previously mentioned, the manipulator has an internal control system that moves the end-effector to a desired task position based on a proportional controller and damping least squares, which is incorporated in the Jacobian of the manipulator $\boldsymbol{J}_2$ and calculated based on the DH-parameters presented in Table 2, similar to [32]. In this way, the pseudo-inverse in (14) becomes

$$\boldsymbol{J}_\chi^\dagger = \boldsymbol{J}_\chi^T(\boldsymbol{J}_\chi\boldsymbol{J}_\chi^T + \lambda)^{-1} \ , \tag{16}$$

where $\lambda$ is the damping factor. This prohibits the manipulator from entering a singularity, and instead slows down and stops the manipulator movement, where a low value for $\lambda$ allows the manipulator to move closer to a singularity before stopping. Avoiding singularities is crucial in order to maintain motion capabilities of the manipulator and provide feasible velocity commands. Another possible control approach for avoiding singularities is to utilize task-priority related techniques, e.g. the task priority redundancy resolution technique [7,33]. The reference velocity $\boldsymbol{\zeta}_r$ for the vehicle is given as

$$\boldsymbol{\zeta}_r = -\boldsymbol{J}_1^\dagger K_1\tilde{\boldsymbol{\sigma}}_1 \ , \tag{17}$$

where $K_1$ is a gain matrix. Furthermore, the combined vehicle and manipulator system can in this case be considered a decoupled kinematic system, where all end-effector motions are carried out solely by the manipulator, meaning that vehicle motions are treated as a disturbance. A method that incorporates vehicle velocity tracking errors into end-effector control has been presented in simulations [36] and in experiments [33], but is not considered here due to the absence of currents and other environmental forces. It is a general assumption that the environment is calm and that the main mission is combined control of both vehicle and manipulator to grip the desired object, only using a monocular camera and a labeled and trained image dataset.

### 4.2. Sliding mode controller

The control system for the vehicle consists of a sliding mode controller (SMC), which makes the states of the vehicle converge to the desired values. This control law is highly applicable for

underwater vehicles, where motions tend to be slowly varying and where hydrodynamic parameters do not need to be known in advance. Furthermore, knowing these parameters accurately is either difficult or impossible [7,37]. The way the SMC works is by controlling the states of the vehicle to a sliding manifold, which has been chosen as

$$\boldsymbol{s} = (\boldsymbol{\nu}_r - \boldsymbol{\nu}) + \boldsymbol{\Lambda} \int_0^t (\boldsymbol{\nu}_r - \boldsymbol{\nu})\, d\tau \ , \tag{18}$$

where $\boldsymbol{\nu}_r$ is the reference velocities and $\boldsymbol{\Lambda}$ is an integral gain matrix. Finally, the control law [7] is then given as

$$\boldsymbol{\tau} = \boldsymbol{K}_D \boldsymbol{s} + \hat{\boldsymbol{g}}(\boldsymbol{\Theta}) + \boldsymbol{K}_S sat(\boldsymbol{s}, \epsilon). \tag{19}$$

In (19), $\boldsymbol{K}_D$ and $\boldsymbol{K}_S$ are positive definite gain matrices. By assuming that the vehicle is neutrally buoyant and that velocities will be small, restoring forces and moments represented by $\hat{\boldsymbol{g}}(\boldsymbol{\Theta})$ can be omitted. Furthermore, the function $sat(\boldsymbol{s}, \epsilon)$ refers to a saturation function of $\boldsymbol{s}$ with lower and upper bound of $\pm\epsilon$, and replaces the signum function to avoid chattering [33,38].

## 5. Stability analysis

This section studies the stability properties of the sliding mode controller and the task error in the sense of Lyapunov stability. A stability proof for a sliding mode controller for a non-holonomic mobile robot based on kinematic position control and for an underwater vehicle controlled through both kinematics and kinetics was derived in [39] and [40], respectively. The stability properties of the proposed sliding mode controller share similarities with the controller in [40], but here contains integral action as well. The proof holds for both vehicle and manipulator, and is based on the object-relative navigation system that is to be expected when performing navigation, guidance and control through a monocular camera. Furthermore, the proof holds for sliding mode control based on velocity control both with and without integral action. The assumption that desired velocities are tracked perfectly has been used, as is common in closed loop inverse kinematics [41]. This assumption is typically not valid for underwater vehicles, which have slow dynamics, but may still hold true for such vehicles for low velocities and good tracking capabilities.

In order to study the stability properties of the proposed sliding mode controller, consider the following Lyapunov candidate function (CLF)

$$V = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s} > 0 \ , \quad \forall \boldsymbol{s} \neq 0 \ , \tag{20}$$

The dynamics of (20) can now be studied by differentiating w.r.t. time, inserting for $\boldsymbol{s}$ using (18) and letting $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu}_r - \boldsymbol{\nu}$ as follows

$$\dot{V} = \boldsymbol{s}^T\dot{\boldsymbol{s}} \tag{21}$$

For increased readability the calculation of $\boldsymbol{s}$ and $\dot{\boldsymbol{s}}$ are done separately. $\boldsymbol{s}$ is found by recognizing that $\tilde{\boldsymbol{\nu}} = \boldsymbol{J}^\dagger\dot{\tilde{\boldsymbol{\sigma}}}$ :

$$\boldsymbol{s} = \tilde{\boldsymbol{\nu}} + \boldsymbol{\Lambda}\int_0^t \tilde{\boldsymbol{\nu}} d\tau \tag{22}$$

$$= \boldsymbol{J}^\dagger\dot{\tilde{\boldsymbol{\sigma}}} + \boldsymbol{\Lambda}\boldsymbol{J}^\dagger\tilde{\boldsymbol{\sigma}} d\tau \ . \tag{23}$$

By applying $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$ and assuming perfect velocity tracking, it is possible to find an expression for perfect velocity tracking by inserting (15) into $\dot{\boldsymbol{\sigma}} = \boldsymbol{J}\boldsymbol{\nu}_r$. This leads to the equation $\dot{\boldsymbol{\sigma}} = -\boldsymbol{K}\tilde{\boldsymbol{\sigma}}$, and (23) is reduced to

$$\boldsymbol{s} = \boldsymbol{J}^\dagger(\boldsymbol{K} + \boldsymbol{\Lambda})\tilde{\boldsymbol{\sigma}} \ . \tag{24}$$

Here it is assumed that $\lambda$ in (16) is chosen small s.t. $\boldsymbol{J}\boldsymbol{J}^\dagger \approx \boldsymbol{I}$. In the next step the behavior of $\dot{\boldsymbol{s}}$ is studied, which can be written as

$$\dot{\boldsymbol{s}} = \dot{\tilde{\boldsymbol{\nu}}} + \frac{d}{dt}\left(\boldsymbol{\Lambda}\int_0^t \tilde{\boldsymbol{\nu}} d\tau\right) \ . \tag{25}$$

The Eqs. (9), (13), (15), and $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu}_r - \boldsymbol{\nu}$ are now inserted into (25) in order to obtain

$$\dot{\boldsymbol{s}} = -\boldsymbol{J}^\dagger\boldsymbol{K}\dot{\tilde{\boldsymbol{\sigma}}} - \boldsymbol{J}^\dagger\ddot{\boldsymbol{\sigma}} + \boldsymbol{\Lambda}(-\boldsymbol{J}^\dagger\boldsymbol{K}\tilde{\boldsymbol{\sigma}} - \boldsymbol{J}^\dagger\dot{\boldsymbol{\sigma}}) \ . \tag{26}$$

By assuming slowly changing velocities it follows that $\ddot{\boldsymbol{\sigma}} \approx 0$, and with $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$ where $\boldsymbol{\sigma}_d$ is constant, (26) is further reduced to

$$\dot{\boldsymbol{s}} = \boldsymbol{J}^\dagger\boldsymbol{K}\dot{\boldsymbol{\sigma}} - \boldsymbol{\Lambda}\boldsymbol{J}^\dagger\boldsymbol{K}\tilde{\boldsymbol{\sigma}} - \boldsymbol{\Lambda}\boldsymbol{J}^\dagger\dot{\boldsymbol{\sigma}} \ . \tag{27}$$

Furthermore, assuming perfect velocity tracking and applying (15) to (27) yields

$$\dot{\boldsymbol{s}} = -\boldsymbol{J}^\dagger\boldsymbol{K}^2\tilde{\boldsymbol{\sigma}} \ . \tag{28}$$

Finally, (21) is now computed by combining the reduced equations for $\boldsymbol{s}$ in (24) and $\dot{\boldsymbol{s}}$ in (28), which yields

$$\dot{V} = \boldsymbol{s}^T\dot{\boldsymbol{s}} \tag{29}$$

$$= \left(\boldsymbol{J}^\dagger(\boldsymbol{K} + \boldsymbol{\Lambda})\tilde{\boldsymbol{\sigma}}\right)^T\left(-\boldsymbol{J}^\dagger\boldsymbol{K}^2\tilde{\boldsymbol{\sigma}}\right) \tag{30}$$

$$= -\tilde{\boldsymbol{\sigma}}^T\boldsymbol{M}\tilde{\boldsymbol{\sigma}} \ , \tag{31}$$

where $\boldsymbol{M} = \left(\boldsymbol{K}^2(\boldsymbol{K} + \boldsymbol{\Lambda})^T\right)$ is a positive definite matrix. It then follows that $\dot{V}$ is negative definite and that there is an asymptotic convergence towards the sliding surface $\boldsymbol{s} = \boldsymbol{0}$ given that $\lim_{t\to\infty}\tilde{\boldsymbol{\sigma}} = \boldsymbol{0}$. To prove that the latter condition holds, consider the following CLF

$$V^* = \frac{1}{2}\tilde{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}} \ . \tag{32}$$

Differentiation of (32) w.r.t. time and utilizing the fact that $\dot{\boldsymbol{\sigma}}_d$ is constant yields

$$\dot{V}^* = \tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}} \tag{33}$$

$$= -\tilde{\boldsymbol{\sigma}}^T\dot{\boldsymbol{\sigma}} \ . \tag{34}$$

By inserting (9), (13), (15) into (34), and assuming that $\lambda$ is sufficiently small, it follows that (34) becomes

$$\dot{V}^* = -\tilde{\boldsymbol{\sigma}}^T\boldsymbol{K}\tilde{\boldsymbol{\sigma}} \tag{35}$$

Hence, with $\boldsymbol{K}$ chosen as positive definite, $\dot{V}^*$ is negative definite, and $\tilde{\boldsymbol{\sigma}} = \boldsymbol{0}$ is exponentially stable. It follows that the sliding surface converges asymptotically to zero. Note that the results only hold locally, i.e. if $\boldsymbol{\sigma}$ belongs to a compact set, as $\tilde{\boldsymbol{\sigma}}$ cannot take on all values in $\mathbb{R}$. The task error has a limit on the distance to the object and the object must be within the camera's horizontal and vertical FOV for $\boldsymbol{\sigma}$ (and $\boldsymbol{\sigma}_d$) to exist (i.e. the object must be seen).

## 6. Experimental testing

This section presents the experimental testing results, represented by two case studies. Case study 1 has been dedicated to vehicle DP, while case study 2 also examined end-effector control and grasping. In both of these cases, it was assumed that the object's position was constant. The desired position relative to the object was chosen as $\begin{bmatrix} x_d & z_d & \psi_d \end{bmatrix} = \begin{bmatrix} 0.18\text{ m} & -0.05\text{ m} & -5^\circ \end{bmatrix}$ for all the experimental tests. The desired relative position implies that the vehicle's camera should keep a relative distance of 18 cm to the object, with the object being 5 cm below the center of the camera in the vertical direction and -5 degrees from the center of the camera in the camera's horizontal direction (width). The reason for choosing these desired values was based on the assumption that the manipulator would have an easy time reaching the object, as it was mounted below the vehicle on its left side. The end-effector position has been plotted in the manipulator base frame to more easily see the motions provided by the arm. A flow chart presenting the procedure is found in Fig. 6, and the applied tuning parameters are presented in Table 4.

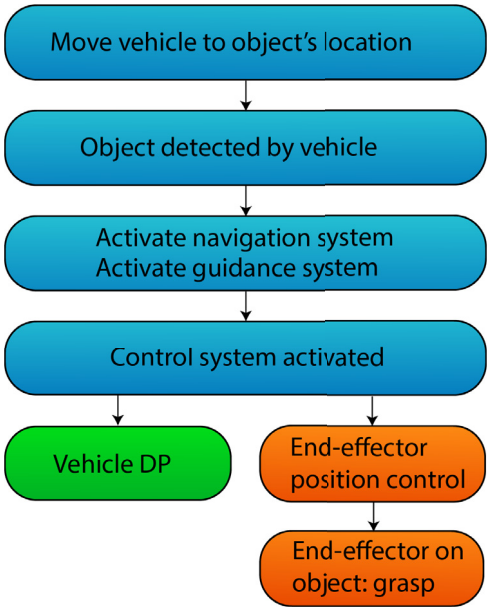


**Fig. 6.** A flow chart showing the experimental testing procedure.

**Table 4**
Tuning parameters used for the reference velocity generation and sliding mode controller.

| Parameter | Value |
|---|---|
| $\boldsymbol{K}_1$ | $[2.5 \quad 2.5 \quad 0.3]$ |
| $\boldsymbol{\Lambda}$ | $[0.02 \quad 0.12 \quad 0.2]$ |
| $\boldsymbol{K}_D$ | $[7.2 \cdot 10^{-4} \quad 4.5 \cdot 10^{-3} \quad 3.6 \cdot 10^{-6}]$ |
| $\boldsymbol{K}_S$ | $[3 \cdot 10^{-3} \quad 1.9 \cdot 10^{-2} \quad 1.5 \cdot 10^{-5}]$ |

**Table 5**
RMSE for the vehicle's relative position and velocity during the vehicle DP operation.

| Position | Value | Velocity | Value |
|---|---|---|---|
| $\text{RMSE}_x$ | 0.025 [m] | $\text{RMSE}_u$ | 0.027 [m/s] |
| $\text{RMSE}_z$ | 0.025 [m] | $\text{RMSE}_w$ | 0.024 [m/s] |
| $\text{RMSE}_\psi$ | 6.7 [deg] | $\text{RMSE}_r$ | 2.0 [deg/s] |

In total, seven grasping experiments were conducted and two of these led to a successful grasp, yielding a grasp success rate of 29%. If only actual grasping attempts are used as a basis for estimating grasp success rate, the rate is increased to 67%.

### 6.1. Case study 1: Vehicle DP

The first case study examines the performance of vehicle DP during object detection in terms of relative position and velocity tracking errors. The results show that there are small errors in both position and orientation in Fig. 7 and linear and angular velocity in Fig. 8, with small oscillations in $z$-direction. The root mean square error (RMSE) is 2.5 cm in x- and $z$-direction, and around 6.7 degrees for the heading angle, as can be seen in Table 5. No difficulties were encountered during this test, and the object was detected at every recorded frame. Low tracking errors report a stable DP control system and increases the probability that a successful grasp can take place.

### 6.2. Case study 2: Vehicle DP and object grasping

For the object grasping experiments, it was decided that the gripper should be closed manually, making this a semi-autonomous operation. Manual gripping of the object was performed in order to decrease the time before the object would be grasped and to increase the probability of a successful grasp. With the low number of DOF, circumventing camera occlusion became difficult, and since no sensor existed near the end-effector to accurately determine when the gripper should close (autonomously) in order to grasp the object was ambiguous. For this case study, seven experimental tests were conducted, where two tests succeeded in grasping the object. The five grasping experiments that failed were terminated because the arm encountered errors, and no attempts to perform the grasp was made here. Therefore, it is argued that these tests should not be conclusive to whether the object could actually be grasped if errors were avoided. Furthermore, it is argued that if these errors had been avoided, the grasping success rate would have increased significantly, as the object would be grasped given enough time. The errors encountered by the manipulator were either related to motions that made it hit the vehicle that turned off the servo motors or entering too close to a singularity configuration. In the first successful test, the system used a lot of time to grasp

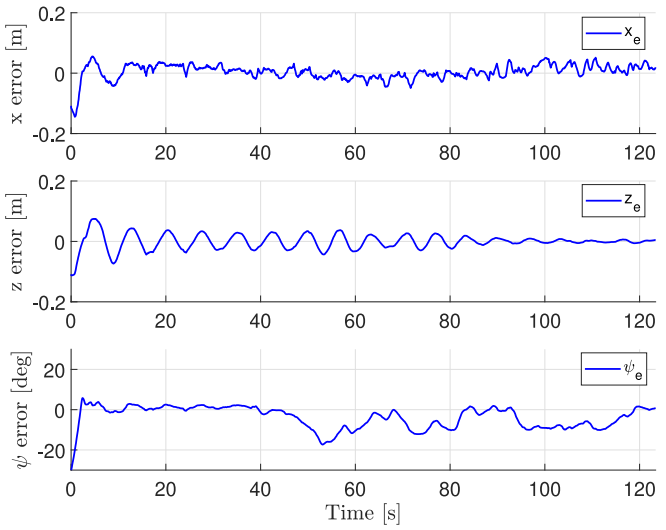


**Fig. 7.** Vehicle error position in x- and $z$-direction [m] and heading angle $\psi$ [deg] during the vehicle DP operation.

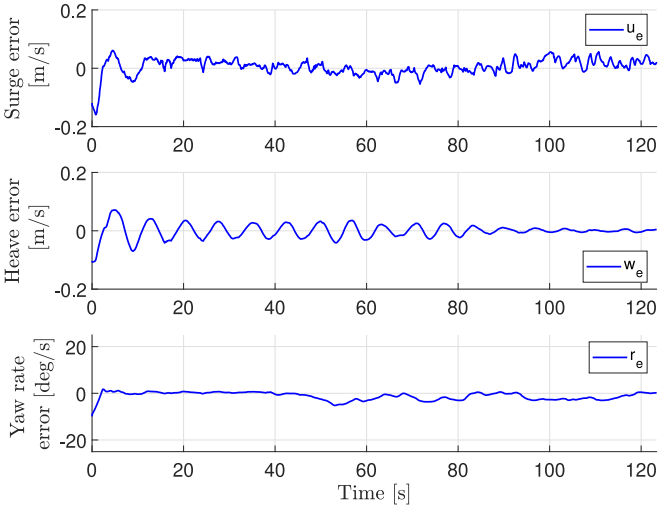


**Fig. 8.** Vehicle velocity error in surge and heave [m/s] and yaw rate $\psi$ [deg/s] during the vehicle DP operation.

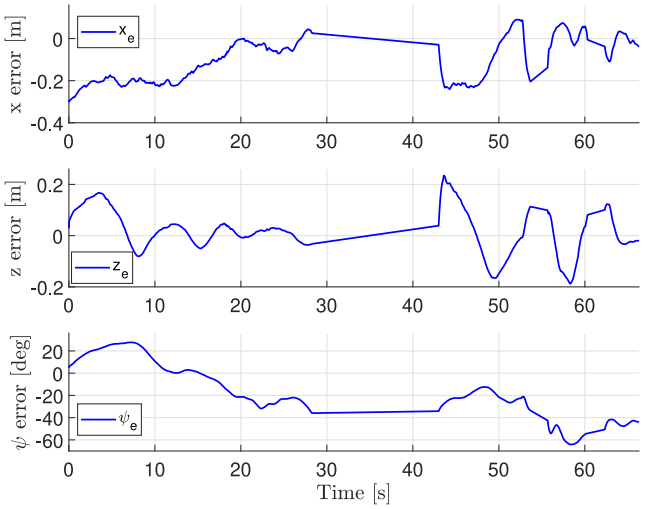


**Fig. 9.** Vehicle error position in x- and $z$-direction [m] and heading angle $\psi$ [deg] during the first grasping operation.

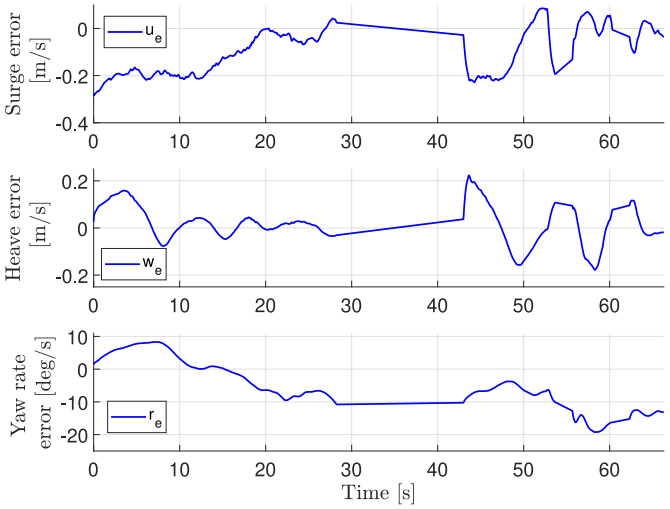


**Fig. 10.** Vehicle velocity error in surge and heave [m/s] and yaw rate $\psi$ [deg/s] during the first grasping operation.

**Table 6**
RMSE for relative vehicle position and velocity and end-effector position during the first grasping test.

| Position | Value | Velocity | Value | Position | Value |
|---|---|---|---|---|---|
| $RMSE_x$ | 0.14 [m] | $RMSE_u$ | 0.13 [m/s] | $RMSE_{ee,x}$ | 0.09 [m] |
| $RMSE_z$ | 0.08 [m] | $RMSE_w$ | 0.07 [m/s] | $RMSE_{ee,y}$ | 0.07 [m] |
| $RMSE_\psi$ | 28 [deg] | $RMSE_r$ | 8 [deg/s] | $RMSE_{ee,z}$ | 0.17 [m] |

the object, while this was achieved much faster in the second successful test. By considering all seven experiments, the case study reports a grasping rate of 29%, but by considering only actual grasping attempts, the grasping rate is 67%.

*Object grasping - Test 1*

For the first object grasping experiment, vehicle position and orientation is presented in Fig. 9, while linear and angular velocity is shown in Fig. 10. The end-effector position, its desired position, and the gripper angle are shown in Fig. 11. The RMSE values are presented in Table 6. The experiment yields an RMSE for the end-effector position of 9.4 cm, 7.5 cm and 16.9 cm in x-, y- and $z$-direction, respectively. In this test, the object was successfully grasped after approximately 65 s.

In the early stages of the experiments, velocity references were tracked slowly and the vehicle struggled reaching the desired distance and heading angle. The integral effect built up slowly, and cable drag prevented the vehicle from reducing the error as the produced surge force was too low. Once the errors were reduced, the end-effector was attempted moved to the object's position. It can be seen in Figs. 9–10, which was also observed during testing, that the relative position and velocity errors suddenly started changing at a constant rate, e.g. between $t = 27$ s–43 s, at around $t = 54$ s and $t = 60$ s. This occurred due to full or partial occlusion of the object by the manipulator. Full occlusion led to no position data, while a partially detected object resulted in either no position data or a bad position estimate. Shortly after $t = 60$ s the object was visible and close to the end-effector, as can be seen in Fig. 11. The vehicle position and manipulator joint angles now allowed a grasp to take place. It can be seen that if the challenge with occlusion at the corresponding time instances was solved, the time before a grasp could be attempted would have been reduced significantly.

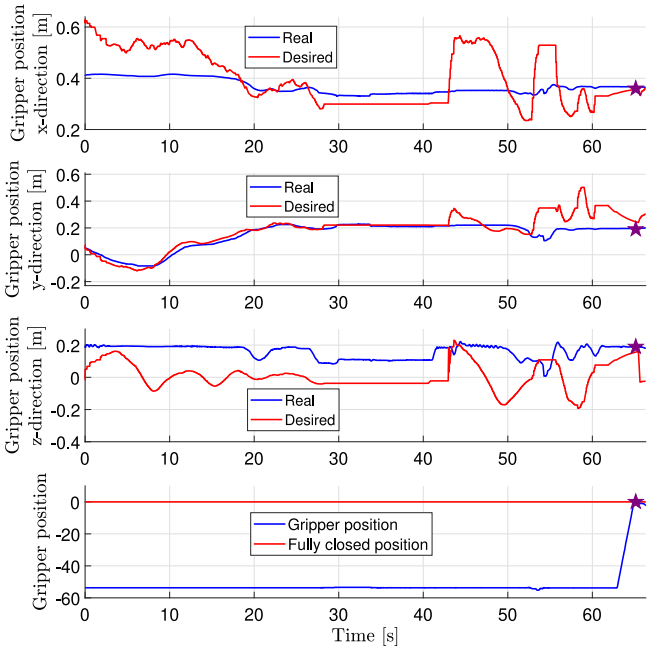


**Fig. 11.** The real end-effector (blue) and desired (red) position in x-, y-, $z$-direction during the first grasping operation. The star-symbol marks the point in time where the successful grasp of the object took place. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

*Object grasping - Test 2*

In the second test, the desired relative position was chosen to be the same as for the previous experiment in order to better compare the two tests. The results from this test are very similar to findings in the previous test, however, this time the object was grasped much sooner. Little occlusion was observed during this operation, and it is reasonable to believe that this caused early object grasping attempts. In total, two grasping attempts were performed (at $t = 14.5$ s and $t = 22$ s). The velocity plot of the vehicle is not included here, as it follows the behavior of the position plot.

The position error of the vehicle can be seen in Fig. 12 and the end-effector position is presented in Fig. 13. The RMSEs for this test are shown in Table 7, and the values are found to be

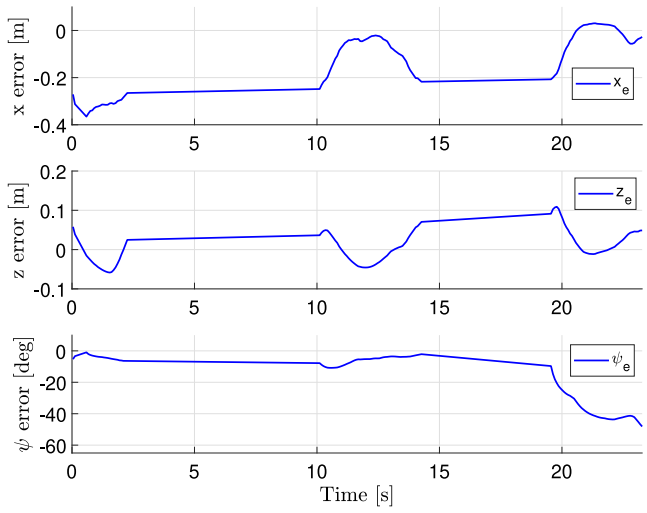


**Fig. 12.** Vehicle error position in x- and $z$-direction [m] and heading angle $\psi$ [deg] during the second grasping operation.

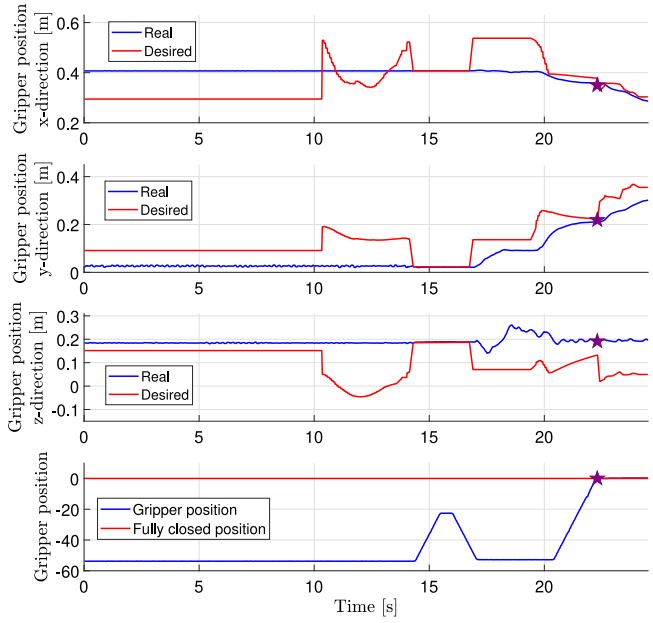


**Fig. 13.** The real end-effector (blue) and desired (red) position in x-, y- and $z$-direction during the second grasping operation. The star-symbol marks the point in time where the successful grasp of the object took place. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

slightly lower in this test compared to the previous experiment. The system experienced some difficulties reaching the desired position, and a rapid increase in the position errors can be found after about 10 s, and were due to partial occlusion of the object. Partial occlusion did not stop the manipulator from moving, but instead allowed motions to be executed. This led to less object occlusion, and an attempt to grasp the object was performed after around $t = 15$ s. However, the grasp failed and the object was pushed outside grasp range. The system then attempted to reposition the end-effector in order to perform a new grasp, and managed to grasp the object after approximately 22 s. Images from this test show a successful grasps, both from the outside and inside the vehicle, in Figs. 14–15.

**Table 7**
RMSE for relative vehicle position and velocity and end-effector position during the second grasping test.

| Position | Value | Velocity | Value | Position | Value |
|---|---|---|---|---|---|
| $\text{RMSE}_x$ | 0.15 [m] | $\text{RMSE}_u$ | 0.17 [m/s] | $\text{RMSE}_{ee,x}$ | 0.09 [m] |
| $\text{RMSE}_z$ | 0.04 [m] | $\text{RMSE}_w$ | 0.04 [m/s] | $\text{RMSE}_{ee,y}$ | 0.07 [m] |
| $\text{RMSE}_\psi$ | 27 [deg] | $\text{RMSE}_r$ | 8 [deg/s] | $\text{RMSE}_{ee,z}$ | 0.11 [m] |

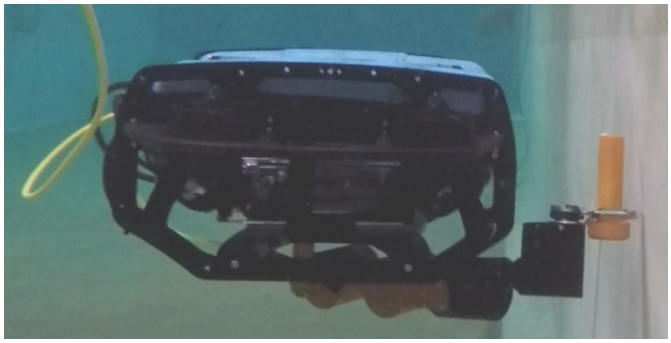

**Fig. 14.** The UVMS successfully grasping the object seen from the outside during the second grasping operation.

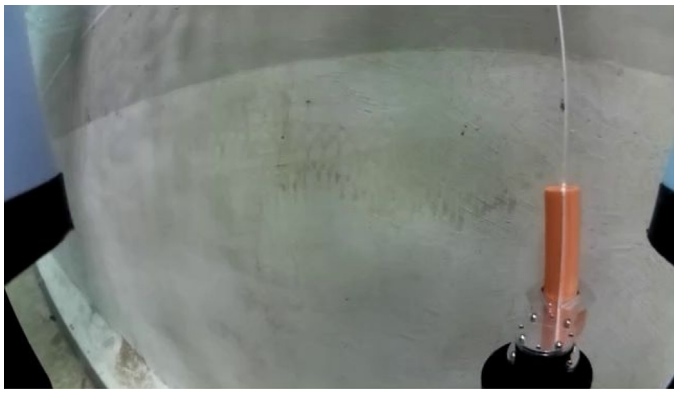

**Fig. 15.** The UVMS successfully grasping the object seen from the vehicle's camera during the second grasping operation.

## 7. Discussion

### 7.1. Grasping success rate

In total, seven experiments were conducted in the MC-lab pool in calm waters and lighting conditions similar to the MARIS project [25] during the day. A test was considered successful if the object was grasped. Two of these experiments led to a successful grasp, yielding a grasp success rate of 29%. This is similar to the results achieved by the MARIS project [25] under the same environmental conditions where the Doppler velocity logger was not used. The five unsuccessful experiments failed when the manipulator stopped, either by entering and being unable to escape a singularity or hitting the vehicle (servo max torque limit was reached, which led to a safety shutdown of the corresponding servo). The fact that the manipulator stopped may be inconclusive to whether the grasp experiment would be successful, given that the system had more time and stopping was avoided. This means that the success rate should be estimated by at least one alternative method.

It is also possible to estimate the grasping success rate by only considering the actual grasping attempts. In the first (successful) grasping test, only one grasp attempt was made, which led to

a successful grasp. In the second test, two attempts were made with one being successful. With this method, the grasping success rate increases to 67%, which is 3% below the results reported by [25] when assisted by a Doppler velocity logger. If only actual attempts of grasping should be considered, the grasp success rate is 67%, but a larger sample size is needed to conclude the effectiveness of the proposed setup. If given a larger sample size and the actual success rate would lie close to 90%, there are still several improvements that must be seen before such a system is utilized in a real case setting.

Nevertheless, these failed attempts show one of the weaknesses with the proposed setup and the manipulator, where occlusion avoidance presents itself as the greatest challenge.

### 7.2. Overall performance and challenges

When it comes to performance and challenges, it is important to note that it was decided that a simulation study would not be carried out before conducting experiments. The methods applied were deemed as simple yet powerful, and the main expected challenge, i.e. occlusion, would be challenging to replicate in a simulation environment in a realistic fashion. The results from a simulation study could have provided knowledge when it comes to tuning of control parameters and enhanced control behavior, but would still run into challenges of overcoming occlusion. Some potential solutions will be addressed in this section.

From the results it can be seen that the UVMS is able to perform DP relative to the object and reach it with the end-effector. As can be seen from Fig. 7, the vehicle position error is low when only the vehicle is actively tracking the object. There are some oscillations in the z-position, which implies that the gains could have been tuned a bit more, and a small spike in the heading angle error after approximately 55 s due to cable drag. The remaining part of the testing shows quick convergence to the desired states, good vehicle velocity tracking, and that the desired position is maintained with small errors. However, when the manipulator arm is actuated and end-effector positions are attempted reached, there are some problems that arise. The time it takes to perform a successful grasp varies for each test, but the system still manages to maintain a desired position relative to the object and grasp it given enough time, as can be seen by comparing the end-effector and gripping results in Figs. 11 and 13.

From the first test in case study 2, it can be seen from Fig. 9 that the object was not found for the period between approximately 27 and 43 s, which led to a constant position change over time for the given duration. This problem occurred a few more times, e.g. after around 53 and 60 s. These incidents occurred due to occlusion of the object by the manipulator arm. Around the 43 s mark the desired end-effector position was reached, but because of partial occlusion wrong position estimates were made. This resulted in a sudden change in the vehicle's and manipulator's estimated x- and z-positions, while the heading angle was approximately the same. It is clear that this is what happened, as the estimated x- and z-positions both depend on the area of the object in the image frame, while the heading angle is estimated through the center of the observed object in the image. In the second test, the object was partially occluded after approximately 10 s, which is given by the sudden jump in the position errors in Figs. 12–13. Attempts could have been made to remove the data where object position was not found in order to reduce the presented RMSE values, but it was found important to show the effect occlusion had on the given system. A low-pass filter was added for relaxing sudden spikes that occurred due to a partially covered object, but it caused a delay on the system that was seen as unwanted. A better approach here may have been to include an outlier detection filter. Optimally, the detector should not have detected the object in this case, or it should have smoothed out the position estimate to reduce the spike.

Several different desired positions were tested, and it was found that the object needed to be at some distance away from the vehicle while also being within the manipulator's workspace. After a few initial tests, desired relative position and heading angle were set to $\begin{bmatrix} x_d & z_d & \psi_d \end{bmatrix} = \begin{bmatrix} 0.18\text{ m} & -0.05\text{ m} & -5^\circ \end{bmatrix}$. The manipulator arm was mounted under the vehicle on its left side, which made it reasonable to try keeping the object close to the arm, within the view of the vehicle camera, and the arm's maximum length. It should also be noted that the proposed relative position and orientation estimation procedure is not 100% accurate. This means that the position and orientation angles had some uncertainties, and the same argument holds for the presented RMSE values. The interpolation that was applied to generate a function to map pixel area to object distance was based on measurements on land with the vehicle's camera. While the object distances were short both in air and water, there would still be a misalignment when estimating the object's position below the surface. Furthermore, it was assumed that light bending and attenuation effects were so small that they could be neglected because of the dome in front of the camera, while it can be seen from light bending effects Fig. 15 that the system may have benefited from a camera calibration procedure.

The proposed method still resulted in good object tracking, especially when object occlusion was avoided, and grasping of the object. It is worth mentioning that the position or vehicle errors do not need to be zero for the vehicle when the grasp takes place, as can be seen from the second test in case study 2 in Fig. 12. Here, the heading error is around 50 degrees at the time of the grasp, but ultimately, it is the position of the end-effector that determines whether a grasp can be successful. This implies that as long as the object is seen, a grasp may still be possible. However, keeping a small vehicle position error is believed to increase the chance of a successful grasp taking place, as the object has a lower probability of escaping the camera image. An interesting point worth mentioning here is that in both of successful grasp tests, the heading angle error is negative and quite large. This may imply that a different (and larger) desired heading angle for the vehicle should have been chosen in order for the manipulator to grasp the object sooner.

When it comes to the stability properties of the system, it is important to mention the assumption of perfect velocity tracking. It clear that the velocity is not tracking the reference velocity perfectly. However, as can be seen in the beginning of the vehicle DP procedure in case study 1 in Fig. 8, the errors admit to taking small values, with RMSEs of $0.027\frac{m}{s}$, $0.024\frac{m}{s}$ and $2\frac{deg}{s}$ in surge, heave, and yaw rate, respectively (see Table 5). Therefore, the assumption of perfect velocity tracking should hold when occlusion is avoided and when the object is within the camera's field of view, which is also the reason stability properties only hold locally. A similar stability proof could be made for the manipulator and the combined system, e.g. with the approach presented here or with the inclusion of end-effector stabilization by consulting [33]. However, the stability properties of the manipulator were not studied here because the manipulator has an in-built velocity controller.

### 7.3. Possible solutions and improvements

The main difficulty with reaching the desired end-effector position and grasping the object was related to the combination of the arm occluding the object and the arm only having 3 DOF. In general, object occlusion implies wrong or non-existent position estimates, while a low number of DOF leads to a small

workspace and a higher probability that the arm approaches singularities. The damping least squares method was applied to avoid singularities, which made the manipulator joints stop when the manipulator was close to a singularity configuration. Attempts could have been made to tune the damping factor $\lambda$ in order to reduce the risk and consequence of the arm reaching a singularity. Nevertheless, controlling a 3 DOF manipulator arm in the 3D task space is difficult, and while exploiting the kinematic redundancy of the UVMS is possible, it is not straight-forward how this should be conducted with the proposed setup.

Overall, no particular action was taken when governing disturbing effect, occlusion, took place. There are several ways this problem can be dealt with, but the effectiveness of each method will vary. Returning the manipulator to its initial position or simply moving the arm away from the camera constitute simple methods for reestablishing the control behavior, but do not deal with the situation directly. Actively circumventing occlusion can be done by masking a model of the object over the occluded part or including occlusion as a task in the task priority framework, but this was not investigated here. Occlusion avoidance is difficult to achieve in a 3D end-effector positioning task with a 3 DOF manipulator, where the system may become underactuated because of the constraints on the vehicle's DOF. With the requirement that the object must be seen by the vehicle's camera, vehicle motions are limited, and exploiting the extra DOF becomes difficult. By utilizing a manipulator with a higher number of DOF, occlusion avoidance becomes easier to achieve, which also allows for end-effector path-following or tracking during vehicle DP. Another interesting idea is to have a camera near the end-effector. Not only is this a possible solution to the occlusion problems encountered here, it may also be regarded as an interesting control problem where vehicle and manipulator control can be conducted through the end-effector camera. Naturally, occlusion avoidance is crucial if such a system is to be utilized in a real world application, where damage to structures, ecosystems and equipment are realistic consequences of failure.

Another difficulty with using the manipulator was that its motions led to hydrodynamic drag on the whole system, which slightly pushed both vehicle and end-effector away from the desired positions. While the detector was able to detect the object, partly or full occlusion of the object resulted in wrong or no position data. This incorrectly led to errors and contributed to high RMSE values. When the object was seen after it had been occluded by the arm, estimated and reference positions jumped to undesired values, which gave sudden motion by the vehicle and manipulator. Furthermore, the drag force created by arm motions could have been reduced by limiting the joint angle velocities, but was not seen as a hindrance in these tests. Cable drag was also an issue that had to be dealt with, which was the main motivation for incorporating integral effect in the controller. In an environment where slowly varying disturbances exist (e.g. an UVMS with tether or weak currents) integral effect is a necessity.

As can be seen from Table 1, the arm has 5 servo motors, which would make the arm 4 DOF. However, the fourth servo motor is responsible for the roll angle of the gripper, and it was chosen to set this angle to 0 in order to align it with the object's roll angle (making the manipulator 3 DOF). Naturally, controlling a 3 DOF manipulator in 3D space is not recommended, but it is of interest to study the potential of such a system in an autonomous or semi-autonomous operation as presented here. In certain applications it might be desired to have short manipulators, e.g. within a confined space such as a fish cage where manipulation operations take place on or close to the net structure. In these experiments, there were no environmental forces that disturbed the system. However, there is still room for improvement, especially when it comes to robustness, inclusion of environmental disturbances, occlusion avoidance, incorporation of the procedure in a large scale mission through field trials and increased level of autonomy.

## 8. Conclusions and further work

In this paper, a method for tracking and grasping an object of interest using a UVMS and monocular camera object detection is presented and verified through experimental testing. Images containing an object of interest were labeled through a previously developed automatic labeling procedure. A model of the object was trained and the object was detected using the YOLOv3 object detection framework. The testing results show that the object can be grasped using the proposed method with a 3 DOF manipulator arm. However, the manipulator occasionally occluded the camera, which led to wrong or no relative position estimates. Moreover, the manipulator reached singularities and stopped in some of the tests which led to failed attempts. By using a single monocular camera, which is common in most small-sized underwater vehicles, grasping objects with a light-weight manipulator in an autonomous fashion is possible, even without the need to implement stereo camera solutions. A stability analysis conducted for the sliding mode controller proves local exponential and asymptotic stability properties of the task error and sliding surface, respectively.

With the proposed setup it is a challenge to perform end-effector position control and grasping without occluding the camera. Further work should study the proposed methodology with a higher DOF manipulator and may include end-effector path-following or tracking. It may also incorporate a camera occlusion task in a task priority framework in order to reduce occlusion effects. It is possible to place a camera near the end-effector and use this camera for grasping. Furthermore, vehicle and end-effector position control can also be done through a camera on the manipulator's end-effector, preferably in conjunction with an algorithm for estimating optimal grip position on the object of interest. Experimental testing may be conducted with such or other improvements, in addition to adding current forces and end-effector stabilization, before moving to field trials. Furthermore, the proposed approach can be applied for other missions as well, e.g., subsea cleaning operations with a brush tool, hot-stab operations within the oil and gas industry, and repairing holes in a fish cage net.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is supported by the Norwegian Research Council project SFI Exposed [grant number 237790] and the Norwegian University of Science and Technology, Department of Marine Technology. The authors would like to thank Mikkel Cornelius Nielsen and Albert Sans Muntadas for support during experimental testing.

## References

[1] I. Schjølberg, T.B. Gjersvik, A.A. Transeth, I.B. Utne, Next generation subsea inspection, maintenance and repair operations, IFAC-PapersOnLine 49 (23) (2016) 434–439, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316320316.

[2] Z. Chen, Z. Zhang, F. Dai, Y. Bu, H. Wang, Monocular vision-based underwater object detection, in: Sensors, 2017.
[3] Z. Chen, Z. Zhang, Y. Bu, F. Dai, T. Fan, H. Wang, Underwater object segmentation based on optical features, Sensors (Basel, Switzerland) 18 (2018).
[4] H. Cho, J. Gu, H. Joe, A. Asada, S.-C. Yu, Acoustic beam profile-based rapid underwater object detection for an imaging sonar, J. Mar. Sci. Technol. 20 (1) (2015) 180–197.
[5] F. Bonin-Font, G. Oliver, S. Wirth, M. Massot, P.L. Negre, J.-P. Beltran, Visual sensing for autonomous underwater exploration and intervention tasks, Ocean Eng. 93 (2015) 25–44.
[6] Blue Robotics Homepage. [Online]. Available: https://bluerobotics.com/.
[7] G. Antonelli, Underwater Robots, Springer, 2014.
[8] Q. Xi, T. Rauschenbach, L. Daoliang, Review of underwater machine vision technology and its applications, Mar. Technol. Soc. J. 51 (1) (2017) 75–97, [Online]. Available: https://doi.org/10.4031/MTSJ.51.1.8.
[9] Y. He, B. Zheng, Y. Ding, H. Yang, Underwater image edge detection based on k-means algorithm, in: 2014 Oceans - St. John's, in: Conference Proceedings, 2014, pp. 1–4.
[10] M. Narimani, S. Nazem, M. Loueipour, Robotics vision-based system for an underwater pipeline and cable tracker, in: OCEANS 2009-EUROPE, in: Conference Proceedings, 2009, pp. 1–6.
[11] Z. Chen, Z. Zhang, F. Dai, Y. Bu, H. Wang, Monocular vision-based underwater object detection, Sensors (Basel, Switzerland) 17 (8) (2017) 1784, [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/28771194 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5580077/.
[12] H. Madjidi, S. Negahdaripour, On robustness and localization accuracy of optical flow computation for underwater color images, Comput. Vis. Image Underst. 104 (1) (2006) 61–76, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S107731420600083X.
[13] Z. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: A review, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–21, [Online]. Available: https://doi.org/10.1109/TNNLS.2018.2876865.
[14] Z. Chen, H. Gao, Z. Zhang, H. Zhou, X. Wang, Y. Tian, Underwater salient object detection by combining 2D and 3D visual features, Neurocomputing (2019) [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219304230.
[15] H. Qin, X. Li, Y. Zhixiong, M. Shang, When underwater imagery analysis meets deep learning: A solution at the age of big visual data, in: OCEANS 2015 - MTS/IEEE Washington, in: Conference Proceedings, 2015, pp. 1–5.
[16] M. Moniruzzaman, S.M.S. Islam, M. Bennamoun, P. Lavery, Deep learning on underwater marine object detection: A survey, in: J. Blanc-Talon, R. Penne, W. Philips, D. Popescu, P. Scheunders (Eds.), Advanced Concepts for Intelligent Vision Systems, in: Conference Proceedings, Springer International Publishing, 2017, pp. 150–160.
[17] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, 2013, CoRR, vol. abs/1311.2524 [Online]. Available: http://arxiv.org/abs/1311.2524.
[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C. Fu, A.C. Berg, SSD: Single shot multibox detector, 2015, CoRR, vol. abs/1512.02325.
[19] J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, 2015, CoRR, vol. abs/1506.02640.
[20] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, CoRR, vol. abs/1804.02767.
[21] H. Choi, M. Kang, Y. Kwon, S. eui Yoon, An objectness score for accurate and fast detection during navigation, 2019.
[22] M.W. Spong, S. Hutchinson, M. Vidyasagar, Robot Modeling and Control, John Wiley and Sons, Hoboken, NJ, 2006.
[23] G. Maroni, S. Choi, J. Yuh, Experimental study on autonomous manipulation for underwater intervention vehicles, 2007, pp. 1088–1094.
[24] E. Simetti, G. Casalino, S. Torelli, A. Sperind, A. Turetta, Floating underwater manipulation: Developed control methodology and experimental validation within the TRIDENT project, J. Field Robot. 31 (3) (2014) 364–385.
[25] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D.L. Rizzini, J. Aleotti, G. Palli, L. Moriello, U. Scarcia, Autonomous underwater intervention: Experimental results of the maris project, IEEE J. Ocean. Eng. 43 (3) (2018) 620–639.
[26] P. Ridao, M. Carreras, D. Ribas, P.J. Sanz, G. Oliver, Intervention AUVs: The next challenge, IFAC Proc. Vol. 47 (3) (2014) 12146–12159, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016435494.
[27] A. Sahoo, S.K. Dwivedy, P.S. Robi, Advancements in the field of autonomous underwater vehicle, Ocean Eng. 181 (2019) 145–160, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0029801819301623.
[28] J. Gancet, D. Urbina, P. Letier, M. Ilzokvitz, P. Weiss, F. Gauch, G. Antonelli, G. Indiveri, G. Casalino, A. Birk, M.F. Pfingsthorn, S. Calinon, A. Tanwani, A. Turetta, C. Walen, L. Guilpain, Dexrov: Dexterous undersea inspection and maintenance in presence of communication latencies, IFAC-PapersOnLine 48 (2) (2015) 218–223, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S240589631500275X.
[29] A. Birk, T. Doernbach, C. Mueller, T. Łuczynski, A. Gomez Chavez, D. Koehntopp, A. Kupcsik, S. Calinon, A.K. Tanwani, G. Antonelli, P. Di Lillo, E. Simetti, G. Casalino, G. Indiveri, L. Ostuni, A. Turetta, A. Caffaz, P. Weiss, T. Gobert, B. Chemisky, J. Gancet, T. Siedel, S. Govindaraj, X. Martinez, P. Letier, Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles, IEEE Robot. Autom. Mag. 25 (4) (2018) 24–33.
[30] E. Simetti, F. Wanderlingh, G. Casalino, G. Indiveri, G. Antonelli, Dexrov project: Control framework for underwater interaction tasks, in: OCEANS 2017 - Aberdeen, in: Conference Proceedings, 2017, pp. 1–6.
[31] M.B. Skaldebø, B.O.A. Haugaløkken, I. Schjølberg, Dynamic positioning of an underwater vehicle using monocular vision-based object detection with machine learning, in: 2019 Oceans - Seattle, 2019.
[32] O.A.N. Eidsvik, B.O. Arnesen, I. Schjølberg, Seaarm-a subsea multi-degree of freedom manipulator for small observation class remotely operated vehicles, in: 2018 European Control Conference (ECC), in: Conference Proceedings, 2018, pp. 983–990.
[33] B.O.A. Haugaløkken, E.K. Jørgensen, I. Schjølberg, Experimental validation of end-effector stabilization for underwater vehicle-manipulator systems in subsea operations, Robot. Auton. Syst. 109 (2018) 1–12, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889018300952.
[34] NTNU, Marine cybernetics laboratory webpage, 2019, Accessed September 10. 2019. [Online]. Available: https://www.ntnu.edu/imt/lab/cybernetics.
[35] I. Schjølberg, T.I. Fossen, Modelling and control of underwater vehicle-manipulator systems, in: In Proc. Rd Conf. on Marine Craft Maneuvering and Control, Citeseer, 1994.
[36] E.K. Jørgensen, I. Schjølberg, ROV end-effector stabilization for unknown, time-varying currents, in: 2016 European Control Conference (ECC), in: Conference Proceedings, 2016, pp. 1303–1308.
[37] T.I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control, John Wiley and Sons, 2011.
[38] J.-J.E. Slotine, W. Li, et al., Applied Nonlinear Control, Vol. 199, Prentice-Hall, Englewood Cliffs, NJ, 1991, no. 1.
[39] M. Alhelou, A. Dib, C. Albitar, Lyapunov theory vs. sliding mode in trajectory tracking for non-holonomic mobile robots, 2015, pp. 1–5.
[40] I.-L.G. Borlaug, J. Sverdrup-Thygeson, K. Pettersen, J. Gravdahl, Combined kinematic and dynamic control of an underwater swimming manipulator, IFAC-PapersOnLine 52 (21) (2019) 8–13, 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896319321603.
[41] G. Antonelli, Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems, IEEE Trans. Robot. 25 (5) (2009) 985–994.

**Bent O. A. Haugaløkken** received his M.Sc. in 2016 and Ph.D. in 2020 at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. His research areas are within development of autonomous technology for inspection, maintenance and repair (IMR) operations in the Norwegian aquaculture, with emphasis on utilization of small-sized underwater vehicles, manipulator arms and sensor technologies.

**Martin B. Skaldebø** received his M.Sc degree in marine technology in 2019 at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. He is currently pursuing his Ph.D, also at NTNU, in marine cybernetics with main focus on underwater robotics. His field of research involves investigating methods for increasing autonomy in underwater operations, mainly within navigation and intervention task, while also quantifying and mitigating the involved risk for the vehicle in such operations.

**Ingrid Schjølberg** is professor in marine technology at the Norwegian University of Science and Technology (NTNU), and is Dean for Research and Innovation at Faculty of Engineering. The focus of Prof. Schjølberg's research is underwater technology mainly related to underwater inspection, maintenance and repair of underwater installations. She has worked with robotics and automation for more than 20 years and in close collaboration with the industry, such as oil and gas, manufacturing, aquaculture and process industry.