Autonomous Grasping Using Novel Distance Estimator

Martin Skaldebø[1,*], Bent A. Haugaløkken[2], and Ingrid Schjølberg[3]

[1] Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway
[2] Department of Seafood Technology, SINTEF Ocean, Trondheim, Norway; Email: bent.haugalokken@sintef.no
[3] Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Trondheim, Norway; Email: ingrid.schjolberg@ntnu.no
*Correspondence: martin.b.skaldebo@ntnu.no

***Abstract*—This paper introduces a novel distance estimator using monocular vision for autonomous underwater grasping. The presented method is also applicable to topside grasping operations. The estimator is developed for robot manipulators with a monocular camera placed near the gripper. The fact that the camera is attached near the gripper makes it possible to design a method for capturing images from different positions, as the relative position change can be measured. The presented system can estimate relative distance to an object of unknown size with good precision. The manipulator applied in the presented work is the SeaArm-2, a fully electric underwater small modular manipulator. The manipulator is unique in its integrated monocular camera in the end-effector module, and its design facilitates the use of different end-effector tools. The camera is used for supervision, object detection, and tracking. The distance estimator was validated in a laboratory setting through autonomous grasping experiments. The manipulator was able to search for and find, estimate the relative distance of, grasp, and retrieve the relevant object in 12 out of 12 trials.**

***Keywords*—object tracking, underwater manipulator, monocular vision, autonomous intervention**

I. Introduction

The field of underwater operations has followed the same trajectory as every other industry today: moving toward increased autonomy. Increased autonomy has the potential to improve various manual operations and even provide solutions to as-yet unsolved challenges. In subsea inspection, maintenance, and repair (IMR), development of autonomous solutions arises out of the desire to reduce operational costs and improve safety [1]. To increase autonomy in robotic systems, robots must be able to capture and make use of environmental information available through the use of sensory equipment. Most underwater operations today are conducted manually, with an operator remotely using the camera of the underwater robotic system as the main tool for creating awareness and perception of the environment to perform a set of tasks. In addition to cameras, the robotic system may be equipped with a number of sensors supplying operation data to the system and operator. Typical sensors include, among others, inertial measurement units (IMUs); several types of cameras (e.g., event cameras) and camera setups (e.g., monocular and stereo camera setups); various types of sonars (e.g., echo sounders, mechanical scanning sonars); force feedback sensors; and pressure sensors. The choice of the specific sensors to be integrated in a robotic system depends on the purpose of the system and the tasks it should solve. It is also necessary to develop systems that allow the robotic system to understand and correctly apply the sensor information to perform the desired task. In addition to the computational capabilities necessary for sensor data analysis, the cost and physical size of sensors can be restricting factors in attaining information needed to conduct a task in an autonomous fashion. Furthermore, intelligent solutions based on limited sensory equipment that can exploit every piece of available information can perform as well as more sensor-heavy systems, thereby reducing both cost and system complexity. Unmanned underwater vehicles (UUVs) are essential in IMR operations and have replaced human divers in the majority of underwater operations performed today. UUVs are also in an exceedingly manner equipped with manipulators for intervention purposes; such vehicles are referred to as underwater vehicle-manipulator systems (UVMSs) [2]. An UVMS provides a moving base for the manipulator and strengthens manipulators’ capabilities and importance in automating various manipulation operations.

Underwater manipulators vary, ranging from simple small electric manipulators with limited lifting capacity and depth ratings to large hydraulic manipulators capable of lifting up to 500 kg at depths of up to several thousand meters with a variety of integrated capabilities (e.g., force feedback, joint position readings) [3]. A manipulator is a versatile tool that has the potential for accessibility and maneuverability and the flexibility to use a range of end-effector tools and different manipulator assemblies for modular arms. They are used in the oil and gas industry [1] as well as aquaculture [4], ocean mapping, environmental monitoring, and surveillance, among others [5].

Manuscript received September 18, 2022; revised November 18, 2022; accepted December 30, 2022.

doi: 10.18178/ijmerr.12.2.64-77

This research applied a fully electric small modular underwater manipulator called SeaArm-2 that is capable of lifting up to 5 kg at full reach. The manipulator was developed at the Norwegian University of Science and Technology (NTNU) and is an excellent testing platform for underwater grasping. The uniqueness of the manipulator is its modularity, with continuous joint revolutions and an integrated monocular camera in the end-effector module [6]. The camera enables environmental awareness and perception for either an operator or autonomous operations. Developing tools for exploiting this visual perception is vital when incorporating autonomous functionality in a manipulator. Such tools may include object detection and tracking. However, these are limited to 2D image information; for intervention operations, a 3D understanding is necessary in order to autonomously maneuver in the environment. This paper extends the work of [6] to incorporate a distance estimator capable of estimating object size and distance. The distance estimator is based on measured joint positions of the manipulator and 2D monocular images from SeaArm-2's integrated camera. Such objects may include fish of unknown size, clams, plastic waste, and so on. Estimating object size and relative distance and using this information in autonomous grasping has, to the authors' knowledge, never been done, not even in lab experiments. The main contributions of this paper are listed below.

(1) Training of an object detector and object tracker using state-of-the-art neural networks on a self-generated image dataset.

(2) Development of a distance estimator capable of estimating object size and distance to objects of unknown size.

(3) Verification of the object detector, tracker, and developed distance (and size) estimator in a laboratory pool with the SeaArm-2 manipulator.

(4) Verification of the developed system's ability to perform autonomous grasping of underwater objects in a laboratory pool.

The paper is structured as follows. Section 2 presents related work. Section 3 discusses the specifications of the computer vision framework with object detection and tracking. Section 4 introduces the novel distance estimator along with theoretical background and a description of the execution of the method. Section 5 presents the experimental setup with the manipulator's specifications, including kinematics, the control system, and the laboratory setup. Section 6 presents the experimental testing and results, including the introduction of two case studies. A discussion of the methods, experiments, and results is provided in Section 7 before the paper is summarized with concluding remarks in Section 8.

II. Related Work

Underwater perception and feature extraction methods are primarily concentrated around acoustic and visual aids. Low visibility, absorption, and scattering of light and turbidity in water give acoustic sensors an advantage over visual-aided sensors that is exceptional for underwater scenes. However, technological advances in camera systems and the use of visual aid prove that camera systems have the potential to be a preferable source for perception, especially for short-range navigation, where the significant time delay and low bandwidth inherent in acoustic communication produce a system incapable of reacting sufficiently in accordance with sensor input [7]. Moreover, visual-aided systems may provide systems with higher spatial and temporal resolutions than their acoustic counterparts [8]. Nonetheless, it is not straightforward to use visual-aided systems in underwater environments, especially when paired with robotic systems during semi- or fully autonomous operations. The underwater scene is considered one of the most challenging environments in which to perform optical detection and recognition of features and patterns, partly because of the problems with visibility, scattering of light, and the like mentioned above [9-10]. Moreover, signal data derived from acoustic sensors are not without errors, given that such signals are prone to data loss due to transmission losses, acoustic noise in thrusters and machinery, signal reflections on different surfaces, absorption loss, and more [11].

The improvement and continuous development of neural networks have promoted the use of visual-aided tools. A conditional generative adversarial network (GAN) for real-time underwater image enhancement was developed in [12]. Their model, FUnIE-GAN, can train on both paired and unpaired images and is capable of boosting performance on several underwater perception tasks, such as object detection and pose estimation. A model for simultaneous image enhancement and super-resolution (SESR) capable of real-time application was proposed by [13]. Their model, Deep SESR, is a residual-in-residual network-based generative model capable of restoring images with up to four times higher resolution.

Underwater grasping involves a vast amount of different scenarios, from pipelines and operational panels in offshore industry to collecting organisms such as plants, shells, and fish. The latter case requires a gentle and agile grasp in order to not damage or injure the object of interest [14]. Such scenarios require a system with high accuracy and delicate movements, which again sets certain requirements for both hardware and software.

This has led to a considerable variety of innovative solutions in the research community. To avoid the common problems of crushing or otherwise damaging objects in the grasping procedure, [15] developed an underwater suction gripper (USG) capable of performing pick-and-place tasks in a quicker motion compared with typical two-finger grippers. The gripper consists of a thruster covered in a 3D-printed resin case with a weight of almost 300 g. Long before that, [7] presented one of the first approaches for autonomous manipulation for underwater intervention with the SAUVIM (Semi-Autonomous Underwater Vehicle for Intervention Mission; University of Hawaii) and performed one of the first sea trials of autonomous intervention in the oceanic environment. Autonomous manipulation with an underwater biomimetic vehicle-manipulator system (UBVMS) was performed by [16]. Their UBVMS used binocular vision to navigate and collect underwater image information and monitoring of the target of interest. Autonomous grasping of objects in a cluttered scene using RGB-D cameras to combine object detection and semantic segmentation was performed by [17]. They used the DenseCap network for object detection to generate bounding boxes and object classes as well as a segmentation network based on the work of [18]. Focusing solely on software solutions, Bagnell *et al.* 2012 [19] developed software for autonomously grasping objects and performing dexterous manipulation tasks with only high-level supervision. The authors were able to effectively localize and grasp individual objects, both previously unseen versions of objects and common manipulation tools. Their system was based on one high-resolution monocular camera, a Bumblebee2 stereo pair, and an SR4000 time-of-flight (ToF) camera providing 3D-sensing capabilities. They used 3D point clouds for detection and localization and 2D vision information, such as color, edges, and textures, to improve match score.

Neural network solutions have also proven popular, where C. Wang *et al.* 2020 [20] conducted reinforcement learning for mobile autonomous grasping with a robot on land, and E. G. Ribeiro *et al.* 2021 [21] trained a network to obtain the location of an object as well as its pose and gripping points. The gripper points predicted by the network form a grasp rectangle representing the position, orientation, and opening of the gripper. Their system achieved millimeter-level accuracy in localizing different objects and could also cope with moving objects by using a second convolutional network to predict necessary linear and angular velocities for the camera to ensure the object remains in the robot’s field of view. On-land autonomous manipulation with a Barrett WAM robot arm using a Bumblebee2 stereo camera as the sensor head was performed in [22]. They successfully executed experiments with unlocking and opening doors, stapling papers, turning a flashlight on and off, and picking up household objects. However, they did not compute grasp points for the gripping procedure, but instead demonstrated pre-grasp poses by manually moving the arm to the desired location relative to objects and storing these relative pre-grasp poses. Sensor calibration and 3D data segmentation for ToF cameras to sample information to use in automatically planning grasping and manipulation actions for a service robot was performed in [23]. They planned grasps for picking up an unknown object and scooping icecream. Neural networks are often described as black-box models, since studying the complicated structure provides little to no insight into how the function works. Ways to determine performance and examine behavior are thus also a focal point in research, and [24] verified tracking performance for underwater manipulation of an ECA ARM 7E Mini. In their experiments, they located errors in tracking performance caused by lack of control performance of joints under low velocities and load.

Stereo-vision and ToF cameras have become popular solutions where 3D space awareness is vital. However, intelligent solutions enable the use of lower-level sensors. Moreover, underwater autonomous intervention was performed in [25] where a monocular camera was used as a primary sensor in combination with Doppler velocity log (DVL) and inertial measurement unit (IMU), without the use of external acoustic sensors. The authors were able to determine six degrees of freedom (DoF) relative pose information between a subsea vehicle and a subsea structure using a combination of model-referenced pose estimation (MRPE) and various navigation sensors. With their proposed navigation algorithm, they were able to successfully perform precise relative navigation and underwater autonomous interventions in experiments. This demonstrated the possibility of using a monocular vision-based navigation approach to conduct real-world underwater intervention tasks on subsea structures. Automating underwater intervention tasks for robotic systems with monocular vision was also investigated in [26]. The presented system was able to estimate the relative pose between an underwater vehicle and surrounding structures of known shape, by combining monocular vision with inertial navigation. Combining monocular vision with an extended Kalman filter, fully autonomous trajectory tracking was successfully achieved in [27]. Here, a vehicle was successfully localized with the respect to a visual map, where the 3D information was determined by fusing inertial measurement data with monocular vision data in the extended Kalman filter. An extensive survey of underwater positioning and navigation was conducted in [28]. The work summarized the use of different positioning systems such as acoustics, global position system (GPS), and monocular and binocular vision. One of the conclusions of the work was that vision-based positioning and navigation can be an effective way to resolve error accumulation that occurs in i.e. acoustic navigation. Furthermore, monocular vision enables real-time performance of extracting movement information and to reduce the error in target location. However, the authors discussed that this could require a large amount of computational power of the corresponding software and hardware systems.

Yet another distinctive solution for autonomous intervention is skill transfer learning (STL): the ability to transfer human skills to robots. This ability is relevant in intervention operations because it can be a method for teaching the manipulator how to grasp objects. An overview on the current state of the art of STL was presented in [29], where there are several versions of STL that can be used for intervention operations: physical interaction (physically guiding and moving the manipulator in desired motions and positions), teleoperation (moving the manipulator using its own sensors and actuators, i.e., using a joystick to move the arm in the desired motions and positions); and human physiological signals (the most futuristic; using human biological signals to perceive human motion to mimic motions or to guide the motions of the manipulator).

III. Object Detecton and Tracking Framework

This section presents the specifications of the computer vision framework. The manipulator is equipped with a low-light HD USB camera in the end-effector module. Footage from the camera is continuously streamed to a topside computer and can be used for supervision aid in manual control or as a tool for autonomous intervention. The novelty of the presented method compared to state of the art solutions, is the that monocular vision is used without the aid of other sensors. The method is capable of detecting, tracking, and estimating distance to objects of unknown size without external sensor information.

A. Object Detector

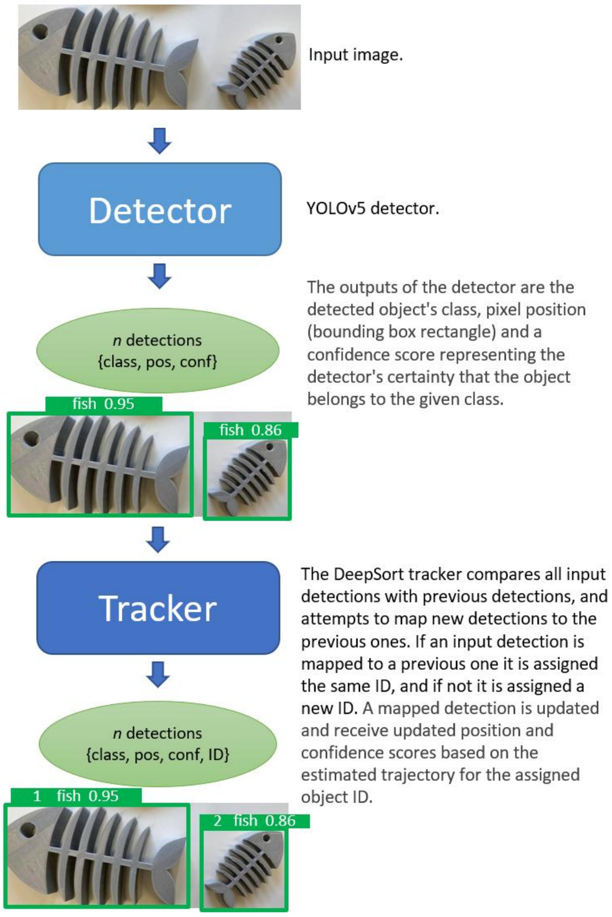
An object detection and tracking model motivated by state-of-the-art solutions capable of identifying and tracking objects of interest for the autonomous grasping procedure has been develop and tested. The object detector is based on the You Only Look Once (YOLO) framework developed by [30], YOLOv5, which is an extension of the popular YOLO algorithm originally developed by Joseph Redmond and later with the help of Ali Farhadi [31, 32]. The model is a neural network trained on a custom image dataset developed for a particular set of objects. The dataset includes images of objects within the laboratory pool environment as well as images of objects in more cluttered scenes, such as the laboratory control room, office space, and tool shelf. The more cluttered scenes expose the detector to other objects of similar colors and shapes and help minimize false positives in the results. All images were labeled with the help of the Computer Vision Annotation Tool (CVAT), an open-source web-based annotation tool [33]. The datasets include 6,533 images and corresponding labels of two object classes: a 3D-printed fish and a 3D-printed fish skeleton. The objects were 3D printed in different sizes to indicate that the distance-estimation process is independent of the object size. The image dataset was split into training, testing, and validation sets (70%/20%/10%) and trained for 300 epochs with batch size 24 on a Nvidia GeForce RTX-2080 Ti GPU. The object detector achieved a mean average precision (mAP) of 0.994 with intersection of union (IoU) at 0.5 and mAP of 0.945 for IoU values between 0.5 and 0.95. The precision score and recall rate of the trained detector were 0.976 and 0.997, respectively. High precision indicates a low number of false positives, whereas high recall represents a low number of false negatives. The obtained precision and recall values indicate low numbers of both false positives and false negatives.

B. Object Tracker

The object detector can effectively locate relevant objects within a frame. However, using an object detector to detect objects in each individual frame means that temporal information is neglected. Temporal information represents information perceived over several time steps. For instance, it may be challenging to determine from a single image of a car whether the car is moving or standing still. However, by looking at a time interval of consecutive images, it becomes clear whether or not the car is in motion. Tracking objects over consecutive image frames can be achieved by utilizing an object tracker, which also facilitates robustness related to temporary occlusions and improves stability in detection. In this work, we applied the DeepSort tracker developed by [34], which is an extension of the popular Simple Online and Realtime Tracking (SORT) algorithm [35]. Similar to SORT, the DeepSort tracker uses a Kalman filter and the Hungarian algorithm for the tracking components. Moreover, in DeepSort, the appearance information of objects is integrated through a pre-trained association metric. This enables DeepSort to track over longer periods of occlusion while continuing to run in real time. Furthermore, steady bounding box detection with minimized noise is critical in ensuring accuracy in determining the object's pixel height and width. An attempt is made to minimize this noise through the integration of the tracker.

C. Computer Vision Framewrok

Inspired by [36], the DeepSort tracker is incorporated with the suggested object detector model described above. Fig. 1 outlines the resulting computer vision framework.



The outputs of the detector are the detected object's class, pixel position (bounding box rectangle) and a confidence score representing the detector's certainty that the object belongs to the given class.

The DeepSort tracker compares all input detections with previous detections, and attempts to map new detections to the previous ones. If an input detection is mapped to a previous one it is assigned the same ID, and if not it is assigned a new ID. A mapped detection is updated and receive updated position and confidence scores based on the estimated trajectory for the assigned object ID.

Figure 1. Outline of the computer vision framework with YOLOv5 detector and DeepSort tracker.

The fully developed framework with the object detector and tracker is capable of both detecting and tracking multiple objects of interest. Moreover, the tracking provides smooth detection between frames and tracking through temporary occlusions, contributing more accurate input to the distance estimator. Previously, M. B. Skaldebø *et al.* [6] proved that an object detector whose bounding box size varied slightly between frames produced noise in the controller input. With the capabilities of the tracker, this noise is reduced significantly. Furthermore, the tracker provides a unique ID for each detection, enabling the system to track and grasp multiple objects while keeping track of which objects have been successfully, unsuccessfully, or not yet grasped or are out of reach.

IV. Size and Distance Estimation

This section presents the novel distance estimator, which is designed to estimate the relative distance between a manipulator and objects of interest. The system requires a manipulator with awareness of joint positions and a monocular camera whose position can be adjusted by joint manipulation. Moreover, the system assumes that the object of interest is fixed, given that a moving object will distort the relative distances measured by the manipulator's movement.

Due to the nature of monocular cameras, the available data from the video stream are 2D images. The state-of-the-art object detector produces bounding boxes to establish the detected object's 2D pixel position within the image frame. In order to grasp the object in a real-world scenario, the object's 2D pixel position and bounding box must be translated to 3D position data. Acquiring the 3D position of objects from 2D images has previously been achieved using objects of known shape and size, in which context a translation between the size of the bounding box and the actual size of the object was demonstrated [6, 37]. However, this method falls short when the size of the object is unknown. Moreover, inspired by these works, a method has been developed in this paper for estimating the size of an object given that the object can be found in the image through object detection based on a trained model.

A. Estimating Object Size

Our method for estimating object size was motivated by [6] and [37], who derived a method for calculating the distance to an object of known size. The distance between the camera and an object of interest x can be calculated as follows:

$$x = \frac{W}{w_{bb}} f_x = \frac{H}{h_{bb}} f_y \,, \tag{1}$$

where W and H are the width and height of the object, w_{bb} and h_{bb} are the pixel width and height of the bounding box enclosing the object in the image frame, and f_x and f_y are the focal lengths of the camera in the x- and y-axis respectively (in the dimension pixels). Eq. (1) is derived from Fig. 2, where the focal lengths $f = fx = fy$ represents the distance from the origin Oc to the principal point ($pxy0$, $pxz0$) in the image frame. If the object width and height are known, estimating the distance to the object is straightforward. However, if they are unknown, a method is required to estimate the object's width and height. These parameters can be estimated using Eq. (1) and by capturing an image of the object of interest for two different locations of the camera. Assuming that two images of the object provide the distances x_i and x_j, the relative distance of the camera positions is given by $\Delta x_{ij}=x_j-x_i$, which can be calculated as follows using the width:

$$x_j - x_i = \Delta x_{i,j} = \left(\frac{W}{w_{bb,j}} - \frac{W}{w_{bb,i}}\right) f \tag{2}$$

and the height:

$$x_j - x_i = \Delta x_{i,j} = \left(\frac{H}{h_{bb,j}} - \frac{H}{h_{bb,i}}\right) f \,, \tag{3}$$

where ($w_{bb,i}$, $h_{bb,i}$) and ($w_{bb,j}$, $h_{bb,j}$) are the bounding box width and height measured at two different positions of the manipulator, more specifically at distances x_i and x_j from the object. Although the distances xi and xj are unknown, the relative distance Δx_{ij} —known as the translation of the manipulator—can be determined through the kinematics of the manipulator. Thus, based on Eq. (2) and Eq. (3), it is possible to estimate the width and height of the object as follows:

$$\widehat{W} = \frac{w_{bb,j} w_{bb,i} \Delta x_{i,j}}{f(w_{bb,i} - w_{bb,j})} \,, \tag{4}$$

$$\widehat{W} = \frac{w_{bb,j} w_{bb,i} \Delta x_{i,j}}{f(w_{bb,i} - w_{bb,j})} \,. \tag{5}$$

Naturally, this only works if the object's position and orientation do not change.

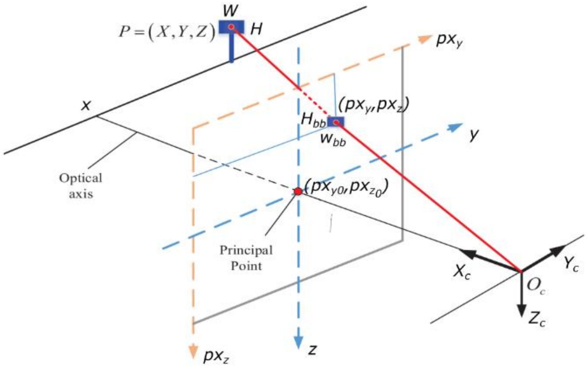


Figure 2. Illustration of domain transformation for an image and corresponding 3D scene, inspired by [37].

B. Multiple Measurements

The distance estimator exploits multiple measurements to increase the accuracy of the object size estimation. To calculate $\widehat{W}$ and $\widehat{H}$ from Eq. (4) and Eq. (5), two measurements are needed from different distances (with the same relative orientation) to the object. Using multiple measurements enables the calculation of new estimates of $\widehat{W}$ and $\widehat{H}$ with every combination of two measurements, meaning an update of the $\widehat{W}$ and $\widehat{W}$ from Eq. (4) and Eq. (5) to $\widehat{W}_{i,j}$ and $\widehat{H}_{i,j}$. The final estimates then become:

$$\widehat{W} = \frac{1}{n_c} \sum_i \sum_{j \neq i} \widehat{W}_{i,j} \,, \tag{6}$$

$$\boldsymbol{\widehat{H}} = \frac{\mathbf{1}}{\boldsymbol{n_c}} \sum_{\boldsymbol{i}} \sum_{\boldsymbol{j \neq i}} \boldsymbol{\widehat{H}_{i,j}} \,, \tag{7}$$

where $\widehat{W}_{i,j}$ and $\widehat{H}_{i,j}$ are calculated from Eq. (4) and Eq. (5) and n_c is the number of possible combinations between samples. n_c can be found as follows:

$$n_c = \frac{n_s^2 - n_s}{2} \tag{8}$$

where n_s is the number of samples.

C. Calculating 3D Position

When the width and height of the object are estimated, the distance between the camera and the object can be calculated as follows:

$$x = \left(\frac{\widehat{W}}{w_{bb}} + \frac{\widehat{H}}{h_{bb}}\right)\frac{f}{2}. \tag{9}$$

With an estimated relative distance between the camera and the object, the object's corresponding relative 3D position in space $\sigma = [x, y, z]$ can now be calculated. From [6], y and z can be calculated as follows:

$$y = x\left(px_y - px_{y,0}\right)dy \tag{10}$$

$$z = x\left(px_z - px_{z,0}\right)dz\,, \tag{11}$$

where px_y and px_z are the pixel positions in the y- and z-directions (i.e., horizontal and vertical directions) of the object's center in the image frame. The parameters dy and dz relate to the physical dimensions of each pixel in the y- and z-directions. According to [6], these can be calculated as follows:

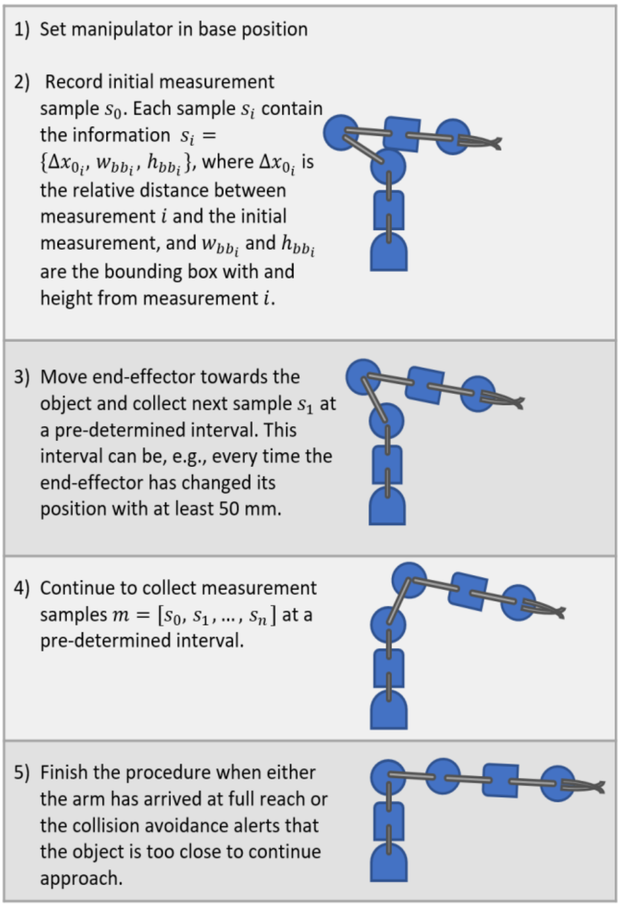
$$dy = \frac{2x\ \tan(\frac{FOV_y}{2})}{PX_y}\,, \tag{12}$$

$$dz = \frac{2x\ \tan(\frac{FOV_z}{2})}{PX_z}\,, \tag{13}$$

where FOV_y and FOV_z are the camera's field of view in the horizontal and vertical directions, respectively, and PX_y and PX_z are the total number of pixels in the image frame in the y- and z-directions, respectively.

D. Distance Estimator Procedure

To improve accuracy in estimated width and height, the distance estimator combines multiple measurements in its calculations. The distance estimator utilizes the full reach of the manipulator to ensure the highest possible variation in measurements. The procedure follows an automatic control procedure with pre-determined steps as a lower-level and higher-level controller that ensures object centering and collision avoidance. Collision avoidance ensures that the manipulator stops approaching the object if either the bounding box exceeds the size of the image frame or the current distance estimate displays a distance below a given threshold. The lower-level procedure follows the steps outlined in Fig. 3.



1) Set manipulator in base position

2) Record initial measurement sample s_0. Each sample s_i contain the information $s_i = \{\Delta x_{0_i}, w_{bb_i}, h_{bb_i}\}$, where Δx_{0_i} is the relative distance between measurement i and the initial measurement, and w_{bb_i} and h_{bb_i} are the bounding box with and height from measurement i.

3) Move end-effector towards the object and collect next sample s_1 at a pre-determined interval. This interval can be, e.g., every time the end-effector has changed its position with at least 50 mm.

4) Continue to collect measurement samples $m = [s_0, s_1, \ldots, s_n]$ at a pre-determined interval.

5) Finish the procedure when either the arm has arrived at full reach or the collision avoidance alerts that the object is too close to continue approach.

Figure 3. Outline of distance estimator procedure.

When approaching the object, the higher-level control procedure ensures that the manipulator keeps the object in the center of the image frame and that the entire object is visible in the frame at all times. This is ensured by continuously centering the object in the image frame and stopping the approach if the detected bounding box occupies the entire image frame width or height. The distance estimator estimates the width and height from Eqs. (4)-(5) as soon as two measurements are sampled. Once more samples are added, the system procedure provides estimates of a new average width and height from all combinations of two samples with Eqs. (6)-(7). The higher-level control procedure also ensures that the manipulator stops if it moves too close to the object.

V. Experimental Setup

The manipulator used in the experiments (SeaArm-2) is presented with corresponding specifications and kinematics. The control system for the manipulator is explained, and the laboratory where the experiments were conducted is presented.

A. SeaArm-2 Underwater Manipulator

The most important features and attributes of the manipulator previously presented in [6] can be summarized with the main specifications listed in Table I.

TABLE I. SEAARM-2 MAIN SPECIFICATIONS.

Parameter	Value
Degrees of freedom	4
Weight in air	3,58 kg
Weight in water	0,35 kg
Max reach (base to end effector)	693,75 mm
Number of servos	5
Stall torque at 12.0 V	25,2 Nm
Full reach lift	5 kg
Depth rating	500 m
Gear ratio	3:1
Onboard computer	Raspberry Pi 3B
Communication	RS485 and Ethernet
Camera	Low-light HS USB camera

1) *SeaArm-2 Manipulator Kinematics*

The transformation matrix is established in order to determine the manipulator’s kinematics and is composed according to the Denavit–Hartenberg (DH) convention [38]. Fig. 4 illustrates the manipulator’s coordinate axis system; the corresponding DH parameters are listed in Table II. Note that these parameters represent the manipulator configuration as assembled in Fig. 4. The coordinate axes are chosen to sufficiently represent the manipulator considering both the camera position and gripper position. Coordinate frame 8 is located at the camera position with the same coordinate configuration as produced from imagery of the camera. The camera is tilted 6*o* toward the gripper to obtain a better view of the gripper; however, this is neglected in the DH parameters since it does not impact the results. Coordinate frame 10 is located exactly in the middle of the gripper fingers and represents the optimal point for positioning an object during grasping operations.

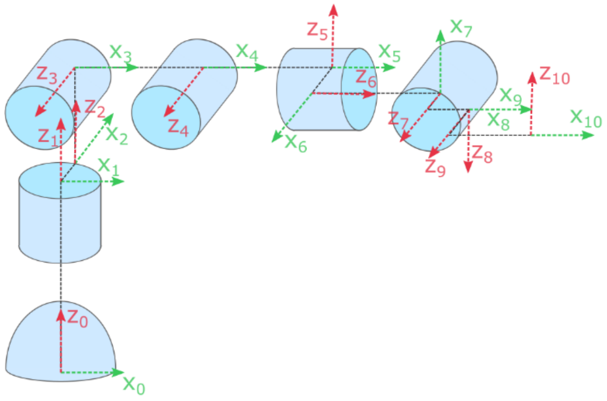


Figure 4. Coordinate axis system of manipulator.

TABLE II. DENAVIT-HARTENBERG PARAMETERS.

i	d_i [mm]	θ_i [rad]	a_i [mm]	α_i [rad]
1	145,4	q_1	0	0
2	0	$\pi/2$	50,1	0
3	80,0	$-\pi/2$	0	$\pi/2$
4	0	q_2	112,0	0
5	0	q_3	80,0	$-\pi/2$
6	0	$-\pi/2$	50,1	$-\pi/2$
7	143,0	$-\pi/2 + q_4$	0	$-\pi/2$
8	25,0	$-\pi/2$	60,0	$\pi/2$
9	0	0	0	$-\pi/2$
10	60	$q_2/2$	40,0	$-\pi/2$

The relationship between the transformation matrix and the DH parameters is represented as:

$$\boldsymbol{T}_{i-1}^{i} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a\,c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a\,s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

where $\boldsymbol{T}_{i-1}^{i}$ is the transformation matrix between coordinate frames $i-1$ and i and c and s correspond to cos(·) and sin(·), respectively. The complete transformation matrix from the base to the end effector can then be written as:

$$\boldsymbol{T}_{base}^{ee} = \boldsymbol{T}_0^1\boldsymbol{T}_1^2\boldsymbol{T}_2^3\boldsymbol{T}_3^4\boldsymbol{T}_4^5\boldsymbol{T}_5^6\boldsymbol{T}_6^7\boldsymbol{T}_7^8\boldsymbol{T}_8^9\boldsymbol{T}_9^{10}. \tag{15}$$

This results in a very comprehensive matrix, which is not shown here due to space constraints. The **T**80 and **T**10 0 are the two most often used matrices in the control approach adopted in this work, as they represent the transformation from the base to camera and the base to end effector, respectively. The transformation matrix is used to determine the forward kinematics as follows:

$$\begin{bmatrix} \sigma_{\{b\}} \\ 1 \end{bmatrix} = T_{\{b\}}^{\{ee\}} \begin{bmatrix} \sigma_{\{ee\}} \\ 1 \end{bmatrix}, \tag{16}$$

where $\sigma_{\{b\}}$ is the position relative to the base frame and $\sigma_{\{ee\}}$ is the position relative to frame ee.

2) *Control system*

This section presents the control system for the SeaArm-2 manipulator. All of the manipulator’s joints are controlled through a kinematic control framework (i.e., geometrical relations), as opposed to kinetics (dynamic) control, which relates the motions to forces and torques. The gripper is controlled by directly controlling the pulse-width modulation (PWM) output, which creates a force-sensitive controller. This ensures a maximum gripping force that can be altered based on what object should be grasped: a more gentle grasp for brittle and more fragile objects and a more forceful grasp for heavier and sturdier objects. A heavy solid object, for example, is naturally grasped with a sufficiently high force, whereas fish or scallops might require a more gentle grasp.

The remaining servos are controlled with angular velocity controllers and have internal proportional–integral–derivative (PID) controllers to distribute the input velocities and convert these velocities to a PWM signal. The PWM signal determines the servo outputs. The Jacobian of the manipulator represents the effect of joint velocities on end-effector velocities and is used with the inverse kinematics to represent the transformation between Cartesian velocities and joint velocities:

$$\dot{\sigma}_{\chi} = J_{\chi}(q)\dot{q}, \tag{17}$$

where the value χ represents the task. The tasks considered in this paper are: (1) manipulator control with camera and $\boldsymbol{T}_0^8$ as a reference system and (2) manipulator gripper control with the gripper and $\boldsymbol{T}_0^{10}$ as a reference system. The task variables $\boldsymbol{\sigma}_{\chi}$ and $\boldsymbol{J}_{\chi}$ are determined by the task-specific transformation matrix along with (16) in the section above and the DH parameters (Table II), respectively. Moreover, to determine the reference joint velocities based on the reference Cartesian velocities, the pseudo-inverse of the Jacobian, $\boldsymbol{J}^{\dagger}$, is used:

$$\dot{q}_r = J_{\chi}^{\dagger}(q)\dot{\sigma}_{\chi,\mathrm{r}}\,, \tag{18}$$

where the pseudo-inverse is augmented with the damped least-squares method and is calculated as:

$$J_{\chi}^{\dagger} = J_{\chi}^{T}(J_{\chi}J_{\chi}^{T} + \lambda\mathrm{I})^{-1} \tag{19}$$

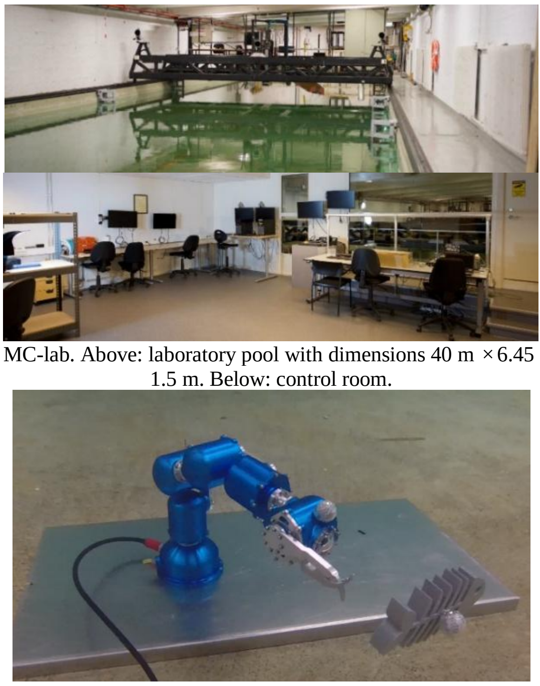
where λ corresponds to the damping term that slows joint movement when the manipulator closes in on a singularity. This introduces small position errors for the end-effector position. However, these are negligible for small values of λ [39]. Moreover, the reference Cartesian velocities $\dot{\boldsymbol{\sigma}}_{\chi,\mathrm{r}}$ are determined by:

$$\dot{\sigma}_{\chi,\mathrm{r}} = \gamma_{\chi}\left(\dot{\sigma}_{\chi,\mathrm{d}} - \dot{\sigma}_{\chi}\right). \tag{20}$$

Here, γ_{χ} is the task-specific gain, and $\dot{\boldsymbol{\sigma}}_{\chi,\mathrm{d}}$ and $\dot{\boldsymbol{\sigma}}_{\chi}$ are the task-specific desired and measured values for $\dot{\boldsymbol{\sigma}}$, respectively.

B. *Laboratory Setup*

The experiments were conducted in the Marine Cybernetics laboratory (MC-lab) at NTNU [40]. The laboratory facility consists of a pool and control room, depicted in Fig. 5a. In the experiments, the manipulator was attached to a steel plate that was lowered to the bottom of the pool. The laboratory setup is depicted in Fig. 5b. The manipulator was placed in the pool, and Qualisys motion markers were placed on the manipulator, just above both the camera and the object. The markers were used with a set of Oqus cameras and the Qualisys Motion Tracking system in order to obtain ground-truth data for evaluating the system's performance during size and distance estimation and grasping accuracy.



(a) MC-lab. Above: laboratory pool with dimensions 40 m ×6.45 m × 1.5 m. Below: control room.

(b) Searm-2 Manipulator and skeleton object placed in the MC-lab pool, with silver Qualisys motion markers attached.

Figure 5. Laboratory setup with facilities and manipulator placement.

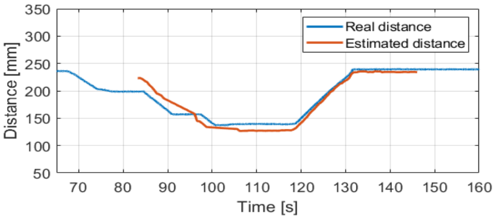
VI. Experimental Testing and Results

This section presents the experimental testing and results. Two case studies were performed to determine the system's capabilities. The first involved testing the distance estimator over multiple trials, in which the estimated distances were compared to the real distances recorded by the 3D motion-capture system Qualisys. The second involved autonomous grasping, where the system used the distance estimator to estimate an object's location before grasping it. Ground-truth position and velocity data of the manipulator and object of interest were logged continuously and were used to compare and validate algorithms, procedures, and overall system performance.

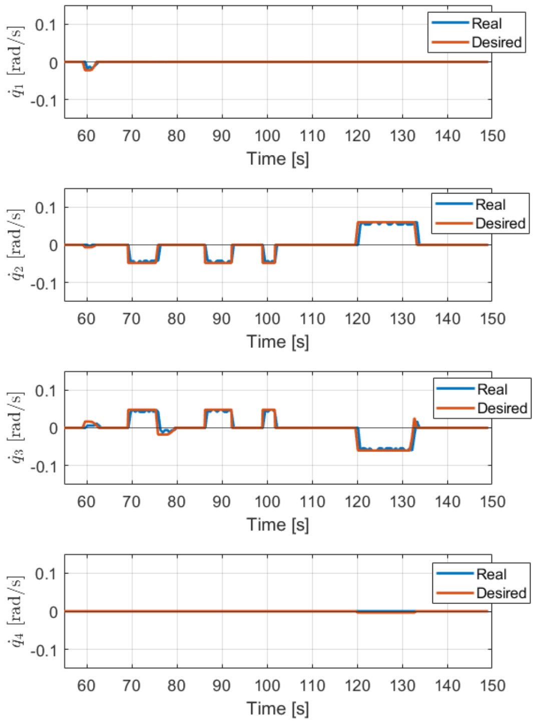
A. *Case Study 1: Distance Estimation*

These experiments are meant to offer an understanding of the performance, reliability, and accuracy of the distance estimator and to highlight any unfavorable behavior of the system. Here, the distance to detectable objects of unknown shape and size was measured. Objects were recognized using the computer vision framework presented in Section III, and their 3D positions in space were estimated using the distance estimator described in Section IV. A set of 20 trials was conducted with two different objects at different positions in the laboratory pool. In each trial, the manipulator placed the detected object in the center of the image frame before conducting the estimation process. This is ensured by the higher-level controller that controls the manipulator to continuously keep the object in the center of the frame. The relative distance between the manipulator and the object was plotted for one of the experiments in Fig. 6a. In this experiment, the system was able to quite accurately estimate the relative distance to the object. The estimated distance was very close to the real relative distance throughout the experiment. After approximately 120 seconds, the relative distance increases again, representing the manipulator's return to the base position (when it is folded backwards to maximize both the camera viewing angle and the potential reach outwards of this position).

The joint velocities for the same experiment can be seen in Fig. 6b. The manipulator's return to base position can also be seen in these velocity plots at approximately 120 seconds, where the joint velocities for joints 2 and 3 are high values with opposite signs. The velocity plots also illustrate the joint movement at time intervals t = [69s, 76s], t = [86s, 92s] and t = [99s, 101s], where the manipulator approaches the object. Between these intervals, the joint velocities are 0 and the manipulator is stationary. This demonstrates the time wherein the manipulator collects new measures, as explained in step 4 of the distance estimator and as outlined in Fig. 3.



(a) Relative distance between manipulator and object. Red line: Estimated distance. Blue line: Actual distance from Qualisys motion capture system.



(b) Joint velocities for joints 1-4. Red line: Desired velocity. Blue line: Actual measured joint velocity.

Figure 6. Relative distance and joint velocities for one distance estimation experiment.

The two objects used in the experiments are depicted in Fig. 7. The silver spheres attached behind the fish skeleton and in the tail of the fish are the markers for the Qualisys motion-capture system. The labels atop the bounding boxes represent tracker ID, object class, and detection confidence.

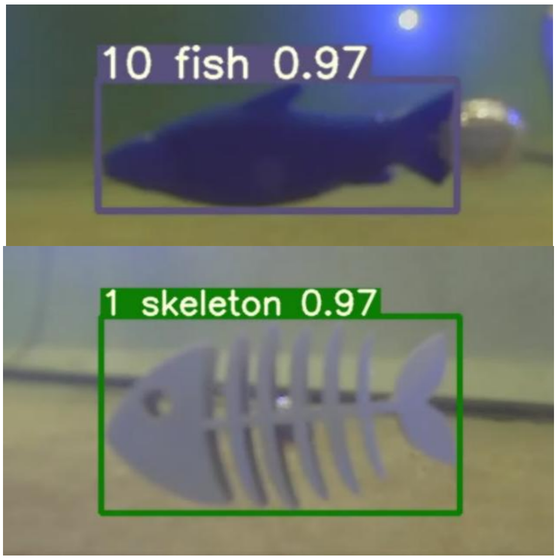


Figure 7. Detection of fish and skeleton objects used in experiments with attached Qualisys markers. The labels atop the bounding boxes represent tracker ID, object class, and detection confidence.

The distance estimator's root mean square error (RMSE) was calculated and is listed in Table III for the 10 distance-estimation trials with the (blue) fish and the 10 trials with the skeleton fish, along with a combined RMSE for all trials. The RMSE for the first 10 trials with the blue fish was approximately 124 mm, whereas the RMSE for the skeleton fish was approximately 27 mm. This large difference resulted from an inaccuracy in the computer vision framework. As depicted in Fig. 7, the blue fish object has a Qualisys marker attached to the tail. The object detector was trained without these markers present in the image training set. This meant that the object detector had not explicitly learned to exclude them from the detections, even though they were attached to the blue fish. When detecting the blue fish object, the detector occasionally included the marker in the bounding box, which again convinced the tracker to retain it in the final detection in order to maintain a consistent bounding box. Moreover, when the manipulator closed in on the object and the image was clearer, the detector correctly detected the object without the attached marker. This resulted in bounding boxes defining differently sized objects, meaning that the initial bounding box measure of the estimator was too large. This resulted in a faulty estimate, affected the results of the distance estimator, and is the main reason for the large RMSE values for the fish object.

TABLE III. DISTANCE ESTIMATOR RMSE VALUES FOR CASE STUDY 1.

RMSE fish	**RMSE skeleton**	**RMSE total**
124,24 mm	26,81 mm	75,53 mm

B. Case Study 2: Autonomous Grasping

In this case study, the manipulator attempted to autonomously grasp an object following the experimental procedure explained in Table IV. The case study involved grasping objects of unknown shape and size that could be recognized through a computer vision framework. The manipulator used the same object detection and tracking system as in case study 1 to detect objects of interests and classify them within the correct object category. The categories used for case study 2 were fish and fish skeleton. The distance estimator explained in Section 4 was used to estimate the size of and distance to the object before attempting to grasp it. Ground-truth values were also measured in order to compare the system's estimates. In total, 12 trials were conducted in which different objects were placed at different locations around the manipulator.

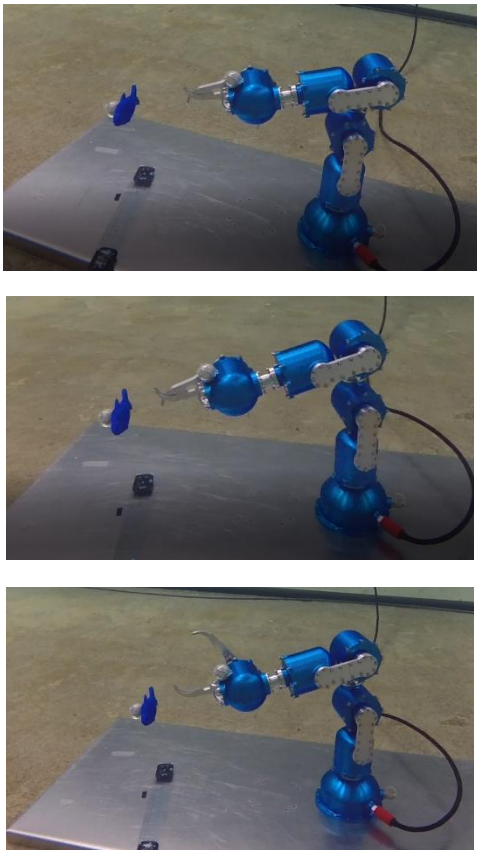
TABLE IV. EXPERIMENTAL PROCEDURE: GRASPING.

	Mode	**Description**
1)	Base position	Arm goes into base position. This position functions as the starting position for subsequent steps of the procedure
2)	Search	Arm rotates around its own base. The arm can either search for a single object and lock in on it or search in a pre-set positional interval and log positions for all relevant objects
3)	Distance	The arm performs the distance estimation as described in Section 4. This step enables the system to estimate the relative distance to the object
4)	Grasp	Grasping the object. The arm approaches the object, estimates optimal grasping angle, and reaches out to close the gripper around the object.
5)	Retrieve	With an object in the gripper, the arm withdraws, relocates to a pre-set retrieval position, and deposits the object.

TABLE V. DISTANCE ESTIMATOR RMSES FOR CASE STUDY 2.

RMSE fish	RMSE skeleton	RMSE total
8,96 mm	20,47 mm	9,81 mm

For all 12 trials, the manipulator was able to successfully grasp and retrieve the object. The distance estimator RMSEs for the experiments are listed in Table V. Each trial started with the manipulator at a unique starting position, with the object in different positions, and with different relative distances between the object and manipulator. These experiments differ from the previous experiments involving pure distance estimation in that the objects were placed at a closer relative distance within grasping reach. This ensured clearer imagery and better detection, thus avoiding the inaccuracies of the size and distance estimation in case study 1. The grasping sequence of one trial is presented in Fig. 8, demonstrating the grasping and initial retrieval of a fish object. Fig. 9 plots the desired and real joint velocities for the same grasping experiment. The modes defined in the experimental procedure in Table IV are highlighted in the plots. The long time period of zero velocities at the start of mode 4 is due to the manipulator planning the grasp sequence, which includes calculating gripping rotation angle and gripping approach. In mode 5, the manipulator retrieves the object, which in this experiment simply involved returning to base position. In the illustrated experiment, joint 4 receives $\dot{q}_4 = 0$ throughout the experiment. This can be explained by Fig. 8, where the object is placed horizontally, resulting in an optimal rotation angle of $q_4 = 0$.



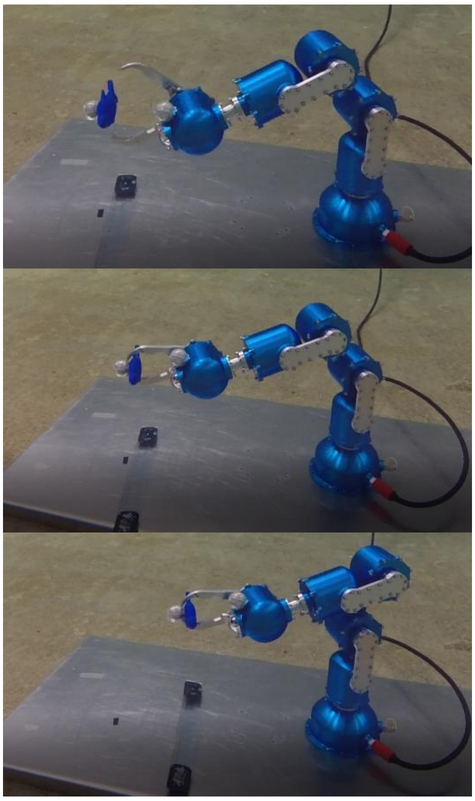


Figure 8. Grasping sequence.

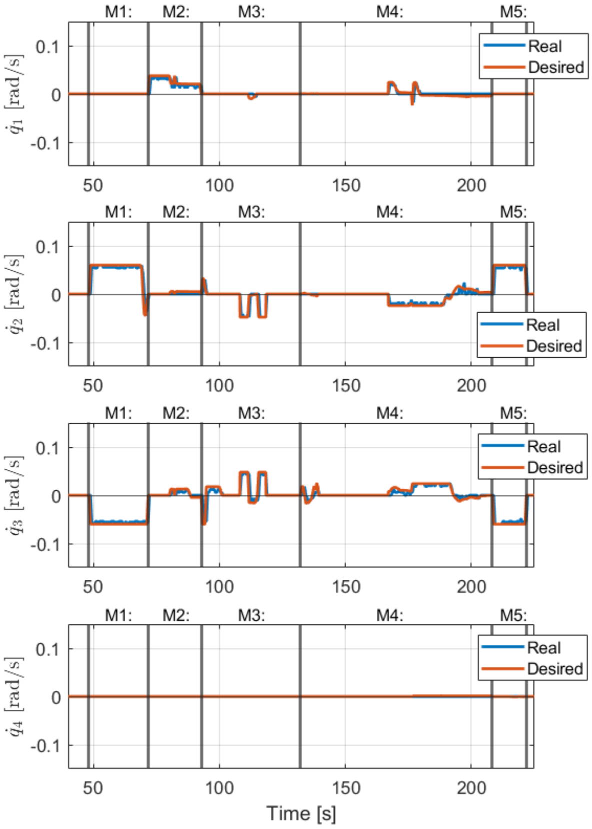


Figure 9. Joint velocities for the grasping experiment depicted in Fig. 8 for joints 1-4 with time on the x-axis and angular velocity on y-axis. The modes from the experimental procedure are highlighted. Red line: Desired velocity. Blue line: Actual measured joint velocity.

End-effector velocities in Cartesian coordinates are presented in Fig. 10. The modes from the experimental procedure are highlighted, and the gripper angular velocity and gripper angle are presented in the last plot. This plot demonstrates how the gripper is activated and opens as soon as mode 4 Grasp is initiated and how it closes at the end of mode 4 in order to grasp the object. The gripper retains an angle of approximately 30° when the object is grasped, indicating that the object is between the gripper's fingers.

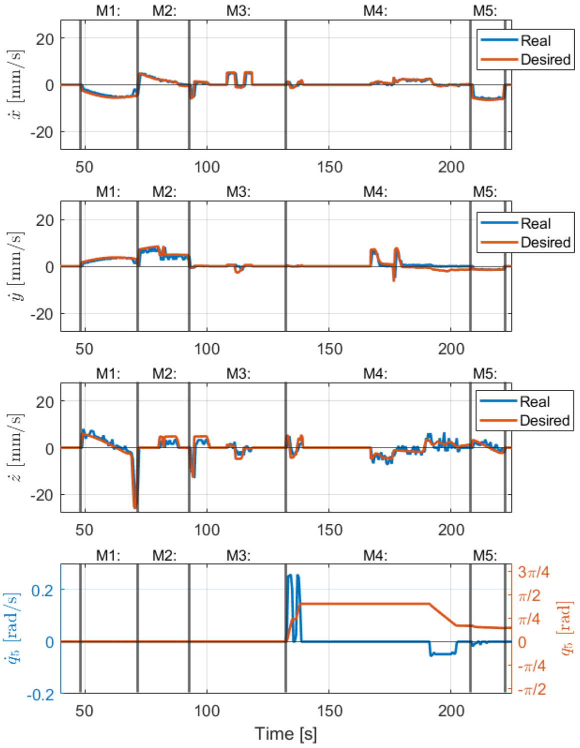


Figure 10. Plots 1-3 show Cartesian velocities for end-effector for the grasping experiment in Fig. 8 with time on x-axis and velocity on y-axis. Red line: Desired velocity. Blue line: Actual measured joint velocity. Plot 4 shows time on x-axis and gripper angle in red with y-axis on the right side and gripper angular velocity in blue with y-axis on the left. The modes from the experimental procedure are highlighted.

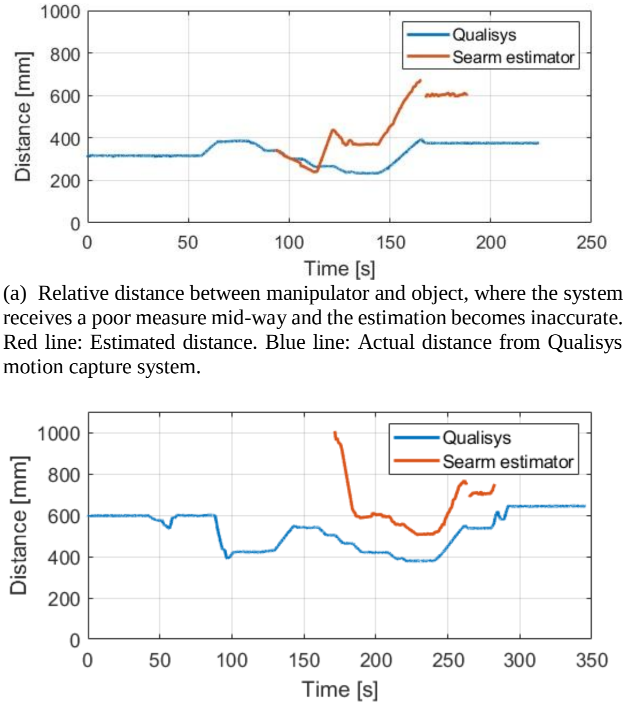
VII. Discussion

The first case study, which was concerned with validating the distance estimator, yielded results with a total RMSE of 75.53 mm. There was a large difference between the two objects: The experiments with the fish object showed an RMSE 4.6 times higher than that for the skeleton fish experiments. The high RMSE value and, in particular, the large difference in RMSEs between the two objects indicate the main challenge of the system setup—namely, the object detector. The detector did not encounter images of the Qualisys markers in the training image dataset and therefore falsely included the markers in the detection of the fish object. This behavior is somewhat understandable in that the marker was unknown to the detector and actually attached to the fish. However, it is nevertheless undesired for the detector to identify the marker as part of the fish object. This led to unstable detections wherein the marker was sometimes included in the bounding box and sometimes left out. Consequently, the pixel positions of the object, which served as input in the distance estimator, were highly varied and therefore inaccurate, which resulted in an inaccurate size and distance estimation.

Two of the experiments that gave rise to high RMSE values were affected by this exact issue. The distance estimates and ground-truth distances for these experiments are plotted in Fig. 11a and Fig. 11b. The plot in Fig. 11a demonstrates how the distance estimate is close to the ground-truth distance given by Qualisys at the start, but at approximately 120 s, the estimator deviates from the ground truth. At this point, an inaccurate measurement was fed as input to the distance estimator, which resulted in an inaccurate estimate. From this point forward, the estimator overshot in its estimates by up to 200 mm, which is insufficient accuracy for grasping. For the experiment in Fig. 11b, the same occurred in the opposite direction: The initial estimate was far off and was then corrected for in the next measurements. However, the system was unable to fully correct for the poor measurements. Therefore, the final estimates overshot by approximately 150 mm.

Another minor challenge discovered in the computer vision framework was the latency introduced by the tracker. In search mode, the system naturally only discovers parts of the object at first, before the entire object is within the field of view. The tracker constantly attempts to retain a stable detection, but the size of the object increases too rapidly for the tracker to keep up when larger portions of the object are revealed in search mode. The tracker then counteracts the intent of increasing the bounding box by slowly increasing the size of the bounding box. Eventually, the bounding box covers the object entirely, but some latency is introduced. This was experienced during testing, but it was not considered a major issue in the proposed setup because the system operates using slow and steady movements. However, it could develop into a larger problem in the context of other, more time-critical scenarios.

The main issue with the designed system, which may be due to a poor detector or latency, was the influence of inaccurate measures. It is difficult to identify an inaccurate measure of something unknown when there are limited data with which to compare. However, the system designed in this paper gathers new measures whenever practical. Hence, there is a potential to filter out inaccurate measurements. One way to do this could be to only accept measurements that do not differ significantly over short periods of time. Moreover, the challenge in this particular case is that the distance estimator requires two different measures to generate one estimate. With three measurements (e.g., measurements A, B, and C), the final distance estimate will be the average of the estimates based on a combination of A–B, B–C, and A–C. This means that, if C represents the inaccurate measurement, it will affect both the B–C and A–C combinations, which in turn implies that the remaining A–B estimate deviates from the others—even if this is the most correct estimate. Thus, how to implement a filtering procedure to omit inac- curate measurements for this system is not straightforward. Furthermore, consider a case in which four measurements are available. Each measurement will be involved in 50% of the combinations. For five or more measurements, the percentage of involvement per measurement decreases. This encourages the use of multiple measurements to attempt to minimize the variance. Other comparison methods (e.g., leave-one-out) can also be implemented, but this requires additional computations and could deteriorate real-time capabilities. This demonstrates the importance of the quantity of measures in determining the quality of measures.



(a) Relative distance between manipulator and object, where the system receives a poor measure mid-way and the estimation becomes inaccurate. Red line: Estimated distance. Blue line: Actual distance from Qualisys motion capture system.

(b) Relative distance between manipulator and object, where the initial measurement is poor and the system is unable to fully recover from this poor measurement. Red line: Estimated distance. Blue line: Actual distance from Qualisys motion capture system.

Figure 11. Estimated and measured relative distance between manipulator and object for two experimental trials that yielded high RMSE values.

Currently, the system has been tested in laboratory experiments, which presents near ideal conditions when it comes to light and turbidity in the water. The object detection and tracking framework showed excellent capabilities of locating the relevant objects, and it is believed that given short distances between object and camera and assuming sufficient lighting sources, the system should be able to perform under more imperfect conditions. However, this need to be investigated in further work to demonstrate the full capabilities of the presented system.

The distance estimator estimates the relative distance by exploiting the system's knowledge of the manipulator's movements. The relative distance is the same: The base is either fixed or moving. This means that the procedure is transferable to a moving base system (e.g., a UVMS, where the manipulator is attached to an underwater vehicle). In a UVMS, the reacting forces between the manipulator and the vehicle are important, as are the overall system's forces and torques due to interactions with the environment. Moreover, by implementing the distance estimator while also considering the reacting forces, a coupled system capable of estimating distances using a monocular camera should be achievable. A UVMS with these capabilities could intervene in a more complex search-and-retrieve scenario and clean out larger areas of fish, plastic, scallops, and so on.

VIII. Conclusions

This paper presented a novel distance estimator using monocular vision for underwater grasping in a kinematic control framework. The estimator can be applied to any robot manipulator where the camera is placed close to the gripper. The proposed estimator was implemented and tested in laboratory experiments, and its performance in autonomous gripping of objects was validated. The experiments were organized into two case studies: one for the estimator and one for the combined distance estimation and grasping operation. Testing the distance estimator highlighted some important challenges with the object detector, as inaccurate detections led to inaccurate distance estimates. An enhanced detector and tracker are expected to notably strengthen the distance estimates. In case study 2, a total of 12 experimental trials with autonomous grasping were conducted. The manipulator was able to successfully search for, locate, estimate the relative distance of, grasp, and retrieve an object in all 12 trials. The inaccuracies of the size and distance estimates were mitigated in the grasping experiments. Future work will include vehicle–manipulator operations. The manipulator will be mounted on a small remotely operated vehicle (ROV) to perform autonomous grasping with a moving base. Multiple object grasping will also be investigated, wherein the system searches for and locates multiple objects before grasping them one by one.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Martin Skaldebø: Conceptualization, methodology, software, hardware integration, review, and editing. Bent A. Haugaløkken: Conceptualization, validation, and review. Ingrid Schjølberg: Validation, review, and editing.

Funding

This work is supported by the Norwegian University of Science and Technology, Department of Marine Technology, and the Research Council of Norway project SEAVENTION (grant no. 280934).

References

[1] I. Schjølberg and I. B. Utne, "Towards autonomy in rov operations," in *Proc. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV*, vol. 48, no. 2, pp. 183–188, 2015.

[2] G. Antonelli, *Underwater Robotics*, Springer, Cham, 2014.

[3] S. Sivčev, J. Coleman, E. Omerdić, G. Dooly, and D. Toal, "Underwater manipulators: A review," *Ocean Engineering*, vol. 163, pp. 431–450, 2018.

[4] H. V. Bjelland, M. Føre, P. Lader, D. Kristiansen, I. M. Holmen, A. Fredheim, E. I. Grøtli, D. E. Fathi, F. Oppedal, I. B. Utne, et al.,

"Exposed aquaculture in Norway," in *Proc. IEEE OCEANS 2015-MTS/IEEE Washington*, 2015, pp. 1–10.

[5] E. Simetti, "Autonomous underwater intervention," *Current Robotics Reports*, vol. 1, pp. 117–122, 2020.

[6] M. B. Skaldebø, B. O. A. Haugaløkken, and I. Schjølberg, "Seaarm-2-fully electric underwater manipulator with integrated end-effector camera," in *Proc. 2021 European Control Conference (ECC)*, pp. 2021, 236–242.

[7] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manip-lation for intervention missions auvs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009.

[8] F. Bonin-Font, G. Oliver, S. Wirth, M. Massot, P. Lluis Negre, and J. P. Beltran, "Visual sensing for autonomous underwater exploration and intervention tasks," *Ocean Engineering*, vol. 93, pp. 25–44, 2015.

[9] Q. Xi, T. Rauschenbach, and L. Daoliang, "Review of underwater machine vision technology and its applications," *Marine Technology Society Journal*, vol. 51, no. 1, pp. 75–97, 2017.

[10] Z. Chen, H. Gao, Z. Zhang, H. Zhou, X. Wang, and Y. Tian, "Underwater salient object detection by combining 2d and 3d visual features," *Neurocomputing*, vol. 391, pp. 249–259, 2020.

[11] M. Skaldebø, A. S. Muntadas, and I. Schjølberg, "Transfer learning in underwater operations," in *Proc. OCEANS 2019-Marseille*, 2019, pp. 1–8.

[12] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3227–3234, 2020.

[13] M. J. Islam, P. Luo, and J. Sattar, "Simultaneous enhancement and super-resolution of underwater imagery for improved visual perception," *CoRR*, vol. abs/2002.01155, 2020.

[14] H. Huang, Q. Tang, J. Li, W. Zhang, X. Bao, H. Zhu, and G. Wang, "A review on underwater autonomous environmental perception and target grasp, the challenge of robotic organism capture," *Ocean Engineering*, vol. 195, p. 106644, 2020.

[15] H. Kumamoto, N. Shirakura, J. Takamatsu, and T. Ogasawara, "Underwater suction gripper for object manipulation with an underwater robot," in *Proc. 2021 IEEE International Conference on Mechatronics (ICM)*, 2021, pp. 1–7.

[16] C. Tang, Y. Wang, S. Wang, R. Wang, and M. Tan, "Floating autonomous manipulation of the underwater biomimetic vehicle-manipulator system: Methodology and verification," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4861–4870, 2018.

[17] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 437–451, 2018.

[18] F. Husain, H. Schulz, B. Dellen, C. Torras, and S. Behnke, "Combining semantic and geometric features for object class segmentation of indoor scenes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 49–55, 2017.

[19] J. A. Bagnell, F. Cavalcanti, L. Cui, T. Galluzzo, M. Hebert, M. Kazemi, M. Klingensmith, J. Libby, T. Y. Liu, N. Pollard, M. Pivtoraiko, J. S. Valois, and R. Zhu, "An integrated system for autonomous robotics manipulation," in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2955–2962.

[20] C. Wang, Q. Zhang, Q. Tian, S. Li, X. Wang, D. Lane, Y. Petillot, and S. Wang, "Learning mobile manipulation through deep reinforcement learning," *Sensors*, vol. 20, no. 3, 2020.

[21] E. G. Ribeiro, R. de Queiroz Mendes, and V. Grassi, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robotics and Autonomous Systems*, vol. 139, p. 103757, 2021.

[22] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Autonomous Robots*, vol. 36, pp. 11–30, Jan. 2014.

[23] Z. Xue, S. W. Ruehl, A. Hermann, T. Kerscher, and R. Dillmann, "Autonomous grasp and manipulation planning using a tof camera," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 387–395, 2012.

[24] J. Chae, T. Yeu, Y. Lee, Y. Lee, and S. M. Yoon, "Trajectory tracking performance analysis of underwater manipulator for autonomous manipulation.," *Journal of Ocean Engineering and Technology*, vol. 34, pp. 180–93, 2020.

[25] J. Park, T. Kim, and J. Kim, "Model-referenced pose estimation using monocular vision for autonomous intervention tasks," *Autonomous Robots*, vol. 44, pp. 205–216, Jan. 2020.

[26] J. Park, T. Kim, and J. Kim, "Model-referenced pose estimation using monocular vision for autonomous intervention tasks," *Autonomous Robots*, vol. 44, pp. 1–12, 01 2020.

[27] A. Manzanilla, S. Reyes, M. Garcia, D. Mercado, and R. Lozano, "Autonomous navigation for unmanned underwater vehicles: Real-time experiments using computer vision," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1351–1356, 2019.

[28] Y. Wu, X. Ta, R. Xiao, Y. Wei, D. An, and D. Li, "Survey of underwater robot positioning navigation," *Applied Ocean Research*, vol. 90, p. 101845, 2019.

[29] Y. Liu, Z. Li, H. Liu, and Z. Kan, "Skill transfer learning for autonomous robots and human–robot cooperation: A survey," *Robotics and Autonomous Systems*, vol. 128, p. 103515, 2020.

[30] G. Jocher, (2020). YOLOv5 by Ultralytics (Version 7.0) [Computer software]. https://doi.org/10.5281/zenodo.3908559J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[31] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," ComputerScience (2018), https://doi.org/10.48550/arXiv.1804.02767 arXiv: 1804.02767.

[32] CVAT.ai Corporation. (2022). Computer Vision Annotation Tool (CVAT) (Version 2.2.0) [Computer software]. https://github.com/opencv/cvat, Aug. 2020.

[33] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. 2017. IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 17–20 September 2017; pp. 3645–3649. https://doi.org/10.1109/ICIP.2017.8296962.

[34] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. 2016 IEEE International Conference on Image Processing (ICIP)*, Sept. 2016.

[35] M. Brostr öm. Real-time multi-object tracker using yolov5 and deep sort. [Online]. Available: https://github.com/mikel-brostrom/Yolov5_DeepSort_Pytorch, 2020

[36] A. Makarov, V. Lukić, and O. Rahnama, "Distance and speed measurements from monocular images," *Real-Time Image and Video Processing 2016* (N. Kehtarnavaz and M. F. Carlsohn, eds.), vol. 9897, pp. 130–140, International Society for Optics and Photonics, SPIE, 2016.

[37] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Trans. ASME E, Journal of Applied Mechanics*, vol. 22, pp. 215–221, June 1955.

[38] A. S. Deo and I. D. Walker, "Overview of damped least-squares methods for inverse kinematics of robot manipulators," *Journal of Intelligent and Robotic Systems*, vol. 14, 1995.

[39] NTNU, "Marine cybernetics teaching laboratory." [Online]. Available: https://www.ntnu.edu/imt/lab/cybernetics. 2020.

Martin Skaldeb ø holds a M.Sc degree in marine cybernetics from the Norwegian University of Science and Technology (NTNU) and is currently a PhD candidate at the Department of Marine Technology at NTNU. His research consist of investigating low cost intelligent solutions for increased autonomy in underwater robotics with emphasis on vision based machine learning applications.

Bent A. Haugaløkken holds an M.Sc. and a Ph.D. degree from the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. He is currently working as a researcher at SINTEF Ocean, Aquaculture Robotics and Automation, in Trondheim. His main research areas are underwater vehicles, manipulators and sensor systems, systems for motion planning, navigation, guidance and control.

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.