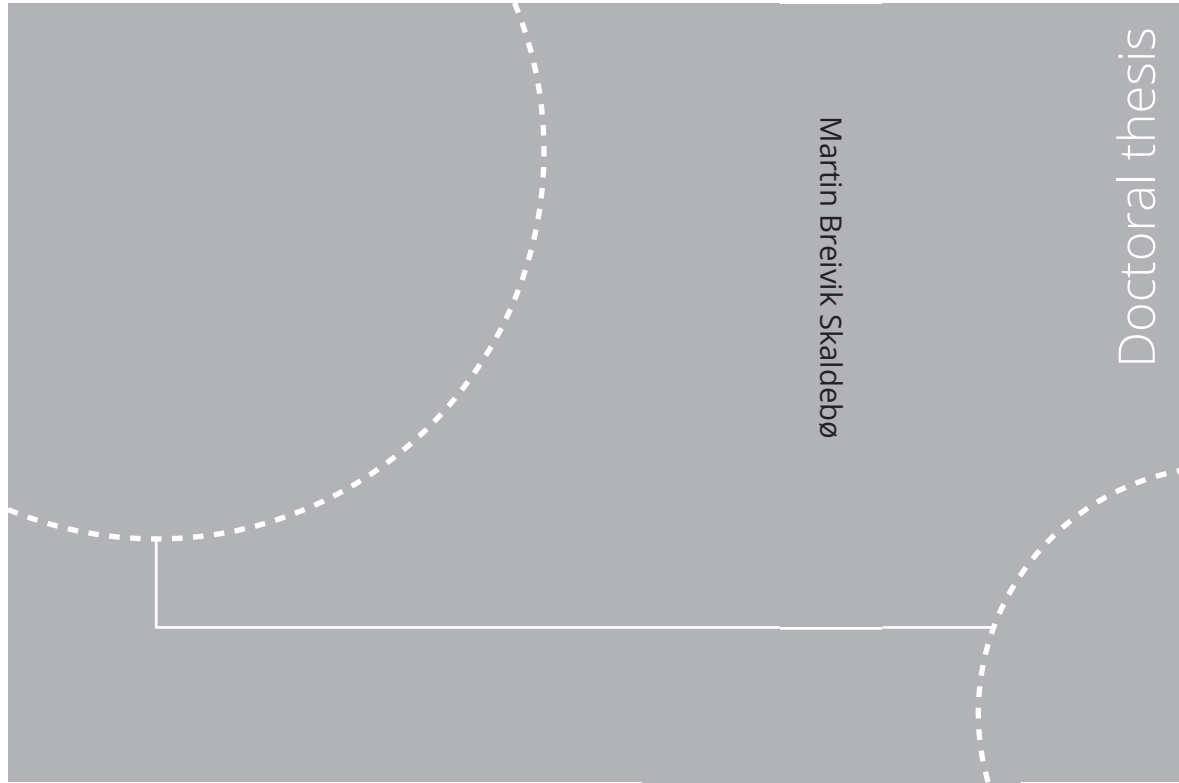


ISBN 978-82-326-6099-5 (printed ver.)
ISBN 978-82-326-5480-2 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)



Doctoral theses at NTNU, 2023:27

Martin Breivik Skaldebø

Intelligent low-cost solutions for underwater intervention using computer vision and machine learning

Doctoral theses at NTNU, 2023:27

NTNU
Norwegian University of
Science and Technology
Thesis for the degree of
Philosophiae Doctor
Faculty of Engineering
Department of Marine Technology

 **NTNU**
Norwegian University of
Science and Technology

 NTNU

 **NTNU**
Norwegian University of
Science and Technology

Martin Breivik Skaldebø

Intelligent low-cost solutions for underwater intervention using computer vision and machine learning

Thesis for the degree of Philosophiae Doctor

Trondheim, January 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology



Norwegian University of
Science and Technology

NTNU

Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering
Department of Marine Technology

© Martin Breivik Skaldebø

ISBN 978-82-326-6099-5 (printed ver.)
ISBN 978-82-326-5480-2 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)

Doctoral theses at NTNU, 2023:27



Printed by Skipnes Kommunikasjon AS

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (Ph.D.) at the Norwegian University of Science and Technology (NTNU). The research presented in this thesis was conducted during the period of January 2019 to July 2022. The work was funded by the Norwegian University of Science and Technology (NTNU). Professor Ingrid Schjølberg has been the main supervisor and Professor Ingrid B. Utne, Albert Sans Muntadas and Bent O. A. Haugaløkken have been the co-supervisors.

The target audience for this thesis is personnel working with development of autonomous applications for underwater intervention systems and underwater vehicle systems, both in industry and academia. The findings from the presented research may help to improve the future development of implementing autonomous functionalities in underwater systems, using mainly low cost equipment.

Current solutions in underwater operations includes underwater vehicles and manipulators that are operated by remote control. Increasing our knowledge about how to use the available technology to make these operations safer and more efficient with less human interaction is essential to achieve more efficient operations. This thesis addresses low cost solutions for increased autonomous functionalities and is, therefore, of great value as it aids in the technological advance towards autonomous systems.

Abstract

This thesis considers intelligent solutions that facilitates for autonomous technology in underwater intervention and navigation. A special focus have been on implementing methods and solutions in inspection, maintenance, and repair (IMR) operations using low cost equipment. The presented work involves development and implementation of solutions to increase the efficiency and safety of operations, and includes both theoretical contributions and experimental testing. The work includes learning algorithms to improve visual authenticity of simulators and digital twin scenarios, computer vision in guidance and navigation of underwater vehicles and intervention systems, development and testing of novel equipment, and experimental verification of presented methods and equipment.

The introduction of advanced learning algorithms enables systems to perform tasks that was previous too complex and complicated for any modelled solutions. This thesis explores the use of generative adversarial networks to improve the realism of simulated environments, which again will improve the transferred learning between simulated environments and real world operations. Such a mapping between domains is complex to model, especially in the underwater scene given the intricate scenery with scattering of light and marine particles. Machine learning algorithms provides new solutions for this mapping, and can aid in improving result from simulation tools to have greater impact on real world operations.

In the same way humans uses their senses to experience life, autonomous systems requires sensory feedback to act and react upon. A sensor is only as effective as the information that can be extracted from the sensory output, and increasing and strengthening this information will improve the support from the sensor. Camera footage contain information with higher spatial and temporal resolution than acoustic information, and utilizing this information to its full will improve today's sensory

systems. This thesis explores the use of visual aid and the potential it brings to increase autonomous capabilities of underwater vehicles and intervention systems. The explored methods includes object detection and tracking networks trained on custom dataset that are able to locate objects within the camera frame. Robust localization of objects is vital in intervention operations to perform autonomous grasping efficiently and safe.

The area of focus for this thesis has been the research and development of low cost solutions for intervention using machine learning. The platform for testing has been the SeaArm-2 manipulator, which is a small electric modular manipulator with an integrated end-effector camera. Part of the thesis work has included development and assembly of the manipulator which is designed to enable both remote control and is optimized for autonomous operations, where the camera enables visual aid to be used as the main sensor input in autonomous grasping operations. Object detectors and trackers in combination with mathematical 3D feature extractors provides a robust system capable of locating the relative 3D position of objects, or features, of interest. A large part of this thesis is dedicated to experimental verification of methods using deep learning and visual aid. This is implemented on the SeaArm-2 manipulator and an underwater vehicle for experimental testing of autonomous functionalities. Experimental testing is important in order to verify the proposed methods and solutions, and in combination with improving results from simulations, and development of digital twins, this thesis provides a good platform for advancing towards a more autonomous tomorrow.

Dedicated to my dear parents

Elisabeth Breivik Skaldebø

Jonn Inge Skaldebø

Acknowledgements

This thesis would not have been possible without the support of my main supervisor, Prof. Ingrid Schjølberg, who gave me this opportunity and has provided me with guidance throughout my PhD. You encouraged me to pursue my ideas, was always available for questions and help whenever needed, and challenged me and showed confidence in me which helped me grow both academically and as a person. I would also like to thank my co-supervisor Prof. Ingrid B. Utne for her encouragement and inspiring conversations.

A special thanks to Bent O. A. Haugaløkken. You helped me get started with my research, and you were there for numerous collaborations where you brought great expertise and guidance in both laboratory experiments and academic writing. Thanks to Mikkel C. Nielsen, Albert Sans Muntadas and Jeevith Hedge for welcoming me with inspiring words and helping me find my place in our research group.

A big thanks to my office mate of 3 years, Lorenzo Balestra, who has always contributed with humor, laughter, good conversations, and of course, cake. My years as a PhD candidate would definitely not have been the same without you sitting on the desk next to me, keeping my spirits high and motivation up. Thanks to all my mates in the NTNU-BIL football group. Our football sessions and matches through these years have been important after busy office hours, and our social gatherings have always been enjoyable and elevated my time in Trondheim.

I would like to thank my family for their support through these years. Thanks to my parents that have always motivated me to keep going in hard times when working on my degree, and my brothers for keeping me updated on the life in my hometown. Last, but not least, I would like to thank my girlfriend and best friend, Synnøve Aune. You help me take my mind of work and relax, which enables me to recharge

and stay focused and motivated even through tough times. Your encouragement and motivation have been a light in my life through a degree affected by a global pandemic with lockdowns and uncertain times. Thank you.

Martin Breivik Skaldebø
Trondheim, Norway

Contents

List of Figures	xv
List of Tables	xvii
Nomenclature	xvii
I Main Report	1
1 Introduction	3
1.1 Background and Motivation	3
1.1.1 Overview	3
1.1.2 Underwater Manipulation	5
1.1.3 Underwater Vehicles	8
1.1.4 Autonomy in Underwater Operations	12
1.2 Research Objectives	15
1.3 Overview of Publications and Contributions	18
1.3.1 Contributions from Articles	18
1.3.2 Other Contributions	22
	ix

1.4	Structure of the Thesis	23
2	State of the Art	25
2.1	Visual Aid in Underwater Robotics	25
2.1.1	Preliminary Discussion	25
2.1.2	Machine Vision	26
2.1.3	Computer Vision	27
	Object Detection and Tracking	28
	2.5D Vision	30
	Transfer Learning	32
2.2	Manipulator Kinematics and Control	34
2.2.1	Preliminary Discussion	34
2.2.2	Manipulator Kinematics	34
	Transformation Matrices	35
	Jacobian Matrix	36
2.3	Underwater Vehicle Manipulator Systems	38
2.3.1	Preliminary Discussion	38
2.3.2	Equations of Motion	38
2.3.3	Dynamic Positioning of Underwater Vehicle Manipulator Systems	39
2.3.4	System Integration of Underwater Vehicle Manipulator Systems	39
3	Facilities and Equipment	41
3.1	Facilities: the Marine Cybernetics Laboratory	41
3.2	Equipment	42
3.2.1	BlueROV2 Underwater Vehicle	42
3.2.2	SeaArm-2 Manipulator	44

3.2.3	UVMS combining the BlueROV2 and the SeaARM-2 . . .	45
4	Summary of Results	47
4.1	Article 1: Transfer Learning in Underwater Operations	47
4.2	Article 2: Dynamic Positioning of an Underwater Vehicle using Monocular Vision-Based Object Detection with Machine Learning	51
4.3	Article 3: Monocular Vision-Based Gripping of Objects	54
4.4	Article 4: SeaArm-2 - Fully Electric Underwater Manipulator with Integrated End-Effector Camera	57
4.5	Article 5: Unsupervised Domain Transfer for Task Automation in Unmanned Underwater Vehicle Intervention Operations	60
4.6	Article 6: Autonomous Underwater Grasping using a Novel Vision- Based Distance Estimator	64
4.7	Article 7: Underwater Vehicle Manipulator System (UVMS) with BlueROV2 and SeaArm-2 Manipulator	68
4.8	Article 8: Dynamic Bayesian Networks for Reduced Uncertainty in Underwater Operations	71
4.9	Article 9: Autonomous Subsea Intervention (SEAVENTION) . . .	74
4.10	Article 10: System Integration of Underwater Vehicle Manipulator System (UVMS) for Autonomous Grasping	75
5	Conclusions and Further Work	81
5.1	Conclusions	81
5.2	Addressing Research Objectives	83
5.3	Other applications and limitations of the work	85
5.4	Further Work	86
	References	89

II	Collection of Articles	99
	Article 1	101
	Article 2	111
	Article 3	122
	Article 4	137
	Article 5	146
	Article 6	159
	Article 7	175
	Article 8	185
	Article 9	193
	Article 10	203
III	Previous PhD Theses Published at the Department of Marine Technology	221

List of Figures

1.1	Illustration demonstrating underwater operations using UUVs. . . .	4
1.2	Underwater manipulators.	7
1.3	Different classes of ROVs.	10
1.4	Subsea docking station and resident ROVs.	11
1.5	An overview of how an autonomous framework can be developed for an underwater manipulator system.	14
2.1	Intersection over union (IoU).	29
2.2	A YOLOv5 model applied to detect 3D-printed fish objects. . . .	30
2.3	Geometric relations between image frame and 3D space.	31
3.1	The MC-lab—towing tank and control room.	42
3.2	The BlueROV2 underwater vehicle.	43
3.3	The SeaArm-2 manipulator.	44
3.4	Tethered communication of the UVMS.	46
3.5	The SeaArm-2 connected underneath the BlueROV2.	46
4.1	Input and output with reconstructed image.	48

4.2	Results from generating images in the real domain with rendered images as inputs.	49
4.3	Results from generating images in the rendered domain with real images as inputs.	50
4.4	DP operation with object detection in the MC-lab pool.	52
4.5	Relative position error and velocity error between the vehicle and object over time.	52
4.6	The UVMS successfully grasping the object.	55
4.7	Vehicle and manipulator end-effector positioning performance during a successful object grasping operation.	55
4.8	The manipulator system during experimental testing.	58
4.9	Desired and measured values for one of the 20 experimental trials.	59
4.10	Results of the image-to-image translation.	61
4.11	Object detector AP score with IoU at 0.5, and mAP score for IoU from 0.5 to 0.95, over training epochs for the CycleGAN network.	62
4.12	Multiframe image of underwater vehicle DP.	63
4.13	The manipulator system in the MC-lab pool detecting an object, seen from outside and from the manipulator end-effector camera.	65
4.14	Grasping operation for one experiment.	66
4.15	Joint velocities and Cartesian velocities for a grasping experiment.	67
4.16	The MC-lab pool in the Gazebo simulator.	69
4.17	Digital twins of the SeaArm-2 manipulator and the UVMS applied in the MC-lab pool environment.	70
4.18	BBN structure with causal dependencies between nodes.	72
4.19	Visualization of CPTs in the BBN structure, with conceptual data.	73
4.20	Three-machine system integration of UVMS.	76
4.21	The SeaArm-2 manipulator and the fish object with camera display in the MC-lab environment in the Gazebo simulator.	77

4.22 Manipulator joint positions and relative gripper-object position for a simulation-based autonomous fixed-base grasping experiment. . .	78
4.23 UVMS in DP operations in the MC-lab with relevant object in front of the vehicle.	79
4.24 SMC output and vehicle velocities.	79

List of Tables

1.1	Levels of autonomy.	13
3.1	BlueROV2 specifications.	43
3.2	SeaArm-2 main specifications.	45
4.1	RMSE for position and velocity of the vehicle relative to the object.	53
4.2	RMSE for joint velocities (in deg/s) over all 20 trials.	58
4.3	Distance estimator RMSE for case study 1.	65
4.4	Distance estimator RMSE values for case study 2.	66
4.5	Time delay between different sections of the system.	70
4.6	Description of the attributes in the input file to ABBA.	72

Nomenclature

List of Abbreviations

AP	Average Precision
ASV	Active stereo vision
AUV	Autonomous Underwater Vehicle
BBN	Bayesian Belief Network
CAD	Computer Aided Design
CNN	Convolutional Neural Network
CPT	Conditional Probability Tables
DH	Denavit-Hartenberg
DOF	Degree of Freedom
DP	Dynamic Positioning
DVL	Doppler Velocity Log
GAN	Generative Adversarial Network
GUI	General User Interface

HMI	Human-Machine-Interface
IMR	Inspection, Maintenance and Repair
IMU	Inertial Measurement Unit
LoA	Levels of Autonomy
NED	North, East, Down
PSV	Passive stereo vision
QTM	Qualisys Track Manager
RMSE	Root Mean Square Error
ROS	Robotic Operating System
ROV	Remotely Controlled Vehicle
RPN	Region Proposals Network
SMC	Sliding Mode Controller
TMS	Tether Management System
UUV	Unmanned Underwater Vehicle
UVMS	Underwater Vehicle Manipulator System

List of Symbols

$\eta \in \mathbb{R}^6$ Position and orientation vector [m, rad]

$\nu \in \mathbb{R}^6$ Body fixed velocity vector [$\frac{m}{s}, \frac{rad}{s}$]

$\omega \in \mathbb{R}^3$ Body fixed angular velocity [$\frac{rad}{s}$]

$\sigma \in \mathbb{R}^3$ Position vector [m]

$\tau \in \mathbb{R}^6$ Thrust vector [kg, kgm^2]

$\Theta \in \mathbb{R}^3$ Orientation vector [rad]

$\mathbf{J} \in \mathbb{R}^n$ Jacobian matrix

$\mathbf{J}^\dagger \in \mathbb{R}^n$ Pseudoinverse of the Jacobian matrix

$\mathbf{q} \in \mathbb{R}^n$ Joint angles [*rad*]

$\mathbf{v} \in \mathbb{R}^3$ Body fixed linear velocity [$\frac{m}{s}$]

Part I

Main Report

Chapter 1

Introduction

1.1 Background and Motivation

This thesis considers intelligent solutions for low-cost underwater robotic systems, focusing on visual aid in guidance, navigation, and control of underwater vehicles and intervention systems. The main research areas considered are the introduction of autonomous functionalities in manipulator systems and underwater vehicle manipulator systems (UVMSs), and the use of digital twins and machine learning to reduce the discrepancy between simulation and real-world operations. The thesis has an emphasis on experimental testing to verify proposed theoretical solutions. The proposed solutions are applicable for most remotely operated vehicles (ROVs) and small electric manipulators. However, some of the guidance and navigation software for autonomous intervention is based on visual aid from a manipulator end-effector camera. This theory assumes visual data from end-effector position is available and requires either manipulators with integrated cameras or other solutions to retrieve such data. The theory and experimental results are highly relevant for inspection, maintenance, and repair (IMR) operations in various industries, particularly in oil, gas, and aquaculture.

1.1.1 Overview

Underwater installations are essential in most subsea industries, for example, the oil and gas industry relies on subsea sites with cables, wells, and machinery, while aquaculture uses fish cages that require infrastructure for supervision, repairs, and cleaning. These facilities are often installed in highly exposed areas offshore. In addition, offshore renewable energy installations are emerging—offshore wind alone is projected to reach \$56.8 billion by 2026 [46] and the global underwater

1. Introduction

robotics market in general is expected to reach \$4.914 billion at a compound annual growth rate of 12.5% from 2018 to 2025 [41]. Due to the remote locations, deep waters, and increasingly challenging conditions, IMR operations are often performed by unmanned underwater vehicles (UUVs). UUVs operating at an underwater site are illustrated in Figure 1.1. In the last decades, UUVs have been endowed with autonomous functionalities; however, these are mainly limited to non-intervention type tasks such as inspection, hovering, and path following. Intervention operations such as turning valves, cleaning, and picking up objects are mostly controlled remotely by one or more human operators. Thus, the efficiency and success rate of such operations is dependent on skills of the operators, and factors such as limited visibility and ocean currents can make operations very challenging to perform in a safe and efficient manner. Hence, there is a need for methods for robust perception and intervention to meet these industry challenges. Increasing the autonomous capabilities of UUVs could improve the safety and reduce the cost and risk of IMR operations [67, 71].

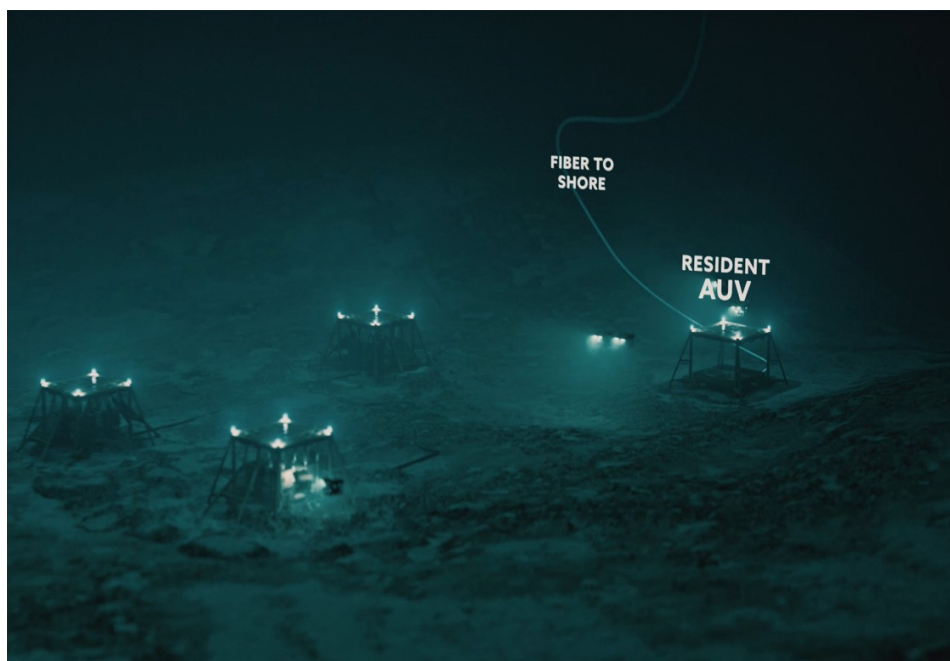


Figure 1.1: Illustration demonstrating underwater operations using UUVs.

For UUVs to operate in underwater intervention, the vehicles should be equipped with intervention tools, most typically manipulators. UUVs equipped with manipulators are commonly referred to as UVMSs [4]. Such a system provides a moving base for the manipulator, which immensely improves the possibilities for

underwater intervention, especially when it comes to increasing the autonomy in intervention operations. Increasing autonomous capabilities in underwater intervention is relevant in everything from operating pipelines and subsea panels in the oil and gas industry to collecting organisms such as plants and fish. The latter case requires a gentle and agile grasp to avoid damaging or injuring the object of interest [29]. Such scenarios require a system with high accuracy and delicate movements, facilitated by specialized hardware and software.

Autonomous underwater intervention has long been a prominent research topic, and numerous innovative approaches have been studied. One of the first experimental operations of autonomous underwater intervention in the oceanic environment was conducted through the SAUVIM project [40]. Since then, other projects have targeted autonomous intervention operations, including the TRIDENT, MARIS, and DexROV projects. TRIDENT demonstrated an object recovery operation using a stereo camera solution, MARIS conducted a pipe grasping mission through three campaigns for day and night light conditions in calm waters [73, 74], and DexROV focused on long-distance teleoperation of ROVs and made provision for advanced dexterous manipulation capabilities [24]. One of the leading ongoing projects in autonomous underwater intervention SUONO, which aims to develop autonomous systems capable of performing operations such as free-floating manipulation tasks on subsea panels [83].

1.1.2 Underwater Manipulation

Most of the existing underwater manipulators, also referred to as underwater robotic arms, are anthropomorphic, meaning they resemble the human arm. A manipulator is composed of a sequence of rigid bodies (links) connected by revolute joints, and its size is often described by their reach. The manipulator reach describes the whole length of the manipulator kinematic chain, including the links and joints, representing the workspace of points reachable by the manipulator end-effector [13]. Most of the currently available manipulators, both commercial and experimental, are actuated by either oil hydraulics or electrical power. Hydraulic manipulators have been in commercial use for longer, but in recent years, electrical manipulators have increasingly entered the market. Hydraulic drive and electrical power both have advantages and disadvantages. In general, hydraulic drive produce a superior output force/torque, thus having a higher power to weight ratio than electrical power. On the other hand, electrical power is better capable of precise motion and force/torque control [76]. A hybrid approach was proposed by [22] to combine the benefits of both power structures. This approach is not yet been commercialized; however, it presents an interesting approach for future manipulator design.

1. Introduction

Depending on their design criteria and operational requirements, manipulators may possess a variety of integrated capabilities, for example, force feedback in the gripper, joint position readings, and integrated cameras or sonars. Manipulators are versatile tools with high accessibility and maneuverability, and the flexibility to use a range of end-effector tools and different manipulator assemblies for modular arms. Blueprint Lab produces the Alpha and Bravo series, which are both recognized as small electric manipulators. The Alpha series ranges from two to four degrees of freedom (DOF) with a lifting capacity of 2.5 kg at full reach. The Bravo series is more powerful and can at full reach lift 15 kg for the five-function, four-DOF arm, ARM 5, and 10 kg for the seven-function, six-DOF arm, ARM 7. The maximum reach of the ARM 7 is 900 mm, and this is the largest manipulator in Blueprint Lab's catalogue (see Figure 1.2(a)). A larger electric manipulator is the ARM 7E produced by the ECA group, shown in Figure 1.2(b). This is also a six-DOF manipulator, however it has a full reach of 1790 mm and can lift 40 kg at full reach. Another interesting electric manipulator is the UMA 1500 produced by Graal Tech. The UMA 1500 is designed with a modular approach and can be assembled with different combinations of sets of one-DOF and two-DOF joints, and different lengths of links. This results in a highly customizable manipulator, and a standard assembly of the UMA manipulator is shown in Figure 1.2(c). Moreover, with this assembly, the manipulator is a six-DOF arm and has 2 m reach and 10 kg lifting capacity. The hydraulic manipulator Kraft Raptor is comparable with the larger electric manipulators in size, with its 1655 mm reach and six DOF (see Figure 1.2(d)). Its power, however, far outperforms the electric manipulators, with a full reach lifting capacity of 91 kg.

Hydraulic actuation typically uses oil; however, biodegradable fluids have also been introduced lately to minimize the environmental impact of leaks. The power structure pumps the fluid through the hydraulic system and converts hydraulic energy to mechanical energy to actuate the joints. By having different diameters on the input and output pipes, with Pascal's law implying that the pressure is equal at both ends, hydraulic actuators are capable of producing an output force much larger than the input force. Moreover, hydraulic systems have much higher power-to-weight ratios than their electric counterparts, enabling more compact systems. This is why hydraulic actuation has long been the preferable option in commercial systems. However, there are also drawbacks to hydraulic actuators. They have poor positioning accuracy and so are unsuited for tasks that require high precision or autonomy. Furthermore, minor leakages and high costs are ubiquitous in hydraulic manipulators.

Electric actuation has always had the advantage of precise motion and force/torque control. Moreover, manipulator designs changed rapidly with the introduction of

1.1. Background and Motivation



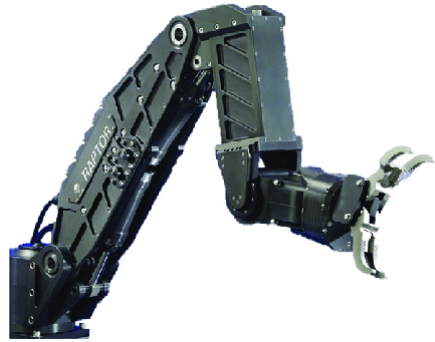
(a) The electric Bravo 7 manipulator, produced by Blueprint Lab [32].



(b) The electric Eca robotics arm 7E, produced by ECA Group [64].



(c) The electric UMA 1500 manipulator, produced by Graal Tech [81].



(d) The hydraulic manipulator Raptor, produced by Kraft TeleRobotics [82].

Figure 1.2: Underwater manipulators.

electrical integrated motors. Modular manipulators, with integrated actuators in each module, can be designed and assembled as per requirement. This circumvents the need to rewire manipulators with every minor design change. Furthermore, in recent years, electric motors are better integrated with gears, feedback sensors, and serial communication ports, making the actuator design simpler and more compact. Since automated and autonomous functionalities are increasingly introduced in manipulator control, and given the improvement of electrical systems, the dominance of hydraulic systems is decreasing. Electric manipulators are the most relevant for this thesis, given that autonomous functionality is an important research objective.

1. Introduction

1.1.3 Underwater Vehicles

A large variety of UVMSs are deployed in a wide range of situations, for example, in the oil and gas industry, aquaculture, ocean mapping, subsurface mineral mining, environmental monitoring, and surveillance [68, 6, 72, 71]. A UVMS is commonly an assembly consisting of a manipulator and either an ROV or an autonomous underwater vehicle (AUV). The diverse array of ROVs is matched with an equally diverse selection of accompanying manipulators, as discussed in Section 1.1.2. Given the large variation in accompanying manipulator size, shape, DOF, and end-effector tools, UVMSs are very versatile tools. ROVs are typically classified within categories based on their size and abilities. With the accelerated technological advancement and development of new areas to investigate, the number and definitions of ROV classes have increased in the last few decades [19]. Moreover, various industries and nationalities follow different standards for ROV classification, and a review of existing ROV standards was performed by Hedge et al. [28] to investigate the gap between the existing ROV standards and future autonomous IMR operation requirements. They listed the American Petroleum Institute, Norsk Sokkels Konkuranseposisjon (NORSOK), the International Organization for Standardization, and the International Marine Contractors Association as the main contributors to ROV standardization guidelines. NORSOK is a project aiming to reduce the development time and cost maintenance cost of petroleum installation on the Norwegian continental shelf. Recently, they presented NORSOK U-102:2020, a set of standards including class categorization of ROVs [84]. They categorized ROVs into the eight classes, further divided into sub-classes, as follows:

Class I—Pure observation. Small vehicles intended for observation only. Primarily fitted with camera, lights, and sonar.

Class II—Observation with payload option. Capable of handling several sensors such as additional cameras, sonar systems, and basic manipulators.

- **Class II A—**Observation class vehicles with payload option .
- **Class II B—**Observation class vehicles with light intervention, survey, and construction capabilities.

Class III—Work class vehicles. Larger and more powerful vehicles than class I and II. Capable of carrying additional sensors and manipulators. Large variation in depth and power capabilities.

- **Class III A—Standard work-class vehicles.**
- **Class III B—Advanced work-class vehicles.**

Class IV—Towed and bottom crawling vehicles. Vehicles maneuvering on the seabed. Large and heavy vehicles that are often designed for specific tasks such as pipeline trenching, excavation, dredging, and other seabed construction work.

- **Class IV A—Towed vehicles.**
- **Class IV B—Bottom crawling vehicles.**

Class V—Prototype or development vehicles. Includes vehicles still in development, vehicles regarded as prototypes, and special-purpose vehicles that do not fit into the other categories.

Class VI—AUVs.

- **Class VI A—AUVs weighing ≤ 100 kg.**
- **Class VI B—AUVs weighing ≥ 100 kg.**

Class VII—High speed survey vehicles.

Class VIII—Fall pipe ROV.

Classes I–III are the most relevant ROV classes for this thesis. Class III vehicles are the typically used ROVs for IMR operations due to their high payload capacity and high depth ratings. Furthermore, Class III ROVs typically allow for additional sensors and tools to operate without being “hardwired” through the umbilical system. This ensures adaptable vehicles that can be easily equipped with necessary tools for different operations. These ROVs are often identified by their power and payload capacities and their ruggedness and reliability. In addition to mission-specific tools, the vehicles are often rigged with thrusters for actuation, one or more manipulators, lights, cameras, inertial measurement units, magnetometers or gyrocompasses, acoustic transceivers, and a Doppler velocity log. A class III ROV equipped with all these devices, the NEXXUS ROV from Oceaneering [50], is illustrated in Figure 1.3(a) along with some other ROVs from other classes.

In the last decade, Class I and II ROVs have significantly increased in popularity, both in industry and in the consumer market. Advances in technology have enabled better systems that includes more sensors and functionalities at lower costs. Low-cost observation-class ROVs now hold the potential to embark on operations previously only accessible with expensive work-class ROVs. The BlueROV2 from Bluerobotics and the Blueeye vehicles from Blueeye Robotics are small Class I and II ROVs that have established strong market positions in the last few years (see Figure 1.3). Both are low-cost ROVs recognized especially in the consumer market

1. Introduction



(a) Class III ROV, the NEXXUS ROV, produced by Oceaneering [50].



(b) Class II ROV, the BlueROV2, produced by BlueRobotics [8].



(c) Class I ROV, the Blueye Pro, produced by Blueye Robotics [62].



(d) Class II ROV, the Blueye X3, produced by Blueye robotics [63].

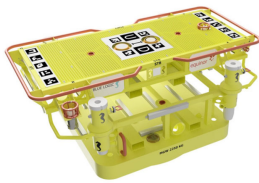
Figure 1.3: Different classes of ROVs.

for leisure activities and exploration as well as for research. The observation-class ROV, BlueROV2, is used extensively in this thesis, both as a model in simulations and for experimental purposes.

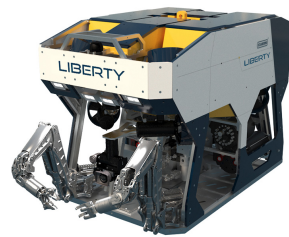
In recent years, there has been extensive research on a new type of ROV, referred to as resident ROVs or R-ROVs. These vehicles are designed to be deployed permanently or semi-permanently at subsea installations, supported by a subsea docking station (see Figure 1.4(a)). R-ROVs can be viewed as an extension of existing tether management systems. A tether management system usually consists of a large weight or module connected to the surface vessel through an armed cable. This reduces the load on the ROV and facilitates deep-sea operations. However,

1.1. Background and Motivation

such a system is still reliant on a surface vessel, associated crew, and weather conditions. An R-ROV system supported by an onshore operations center for remote operations can circumvent these dependencies, reducing costs and increasing safety, operability, and productivity. R-ROVs can adopt similar designs and specifications to work-class ROVs, such as Oceaneering’s Liberty E-ROV, or a hybrid of AUV and ROV designs, such as Oceaneering’s Freedom vehicle and the Eelume snake robot (see Figure 1.4). The combination of AUV and ROV designs allows for both autonomous and remote control by a detachable tether that can be connected and disconnected based on operation specifications.



(a) Subsea docking station, produced by Blue Logic [36].



(b) The Liberty E-ROV, produced by Oceaneering [49].



(c) The Freedom resident ROV, produced by Oceaneering [48].



(d) The Eelume robot [79].

Figure 1.4: Subsea docking station and resident ROVs.

R-ROVs coupled with autonomous solutions are expected to play a significant role in forthcoming subsea operations. The development of autonomous functionalities such as autonomous docking at resident subsea docking stations, autonomous recovery in the case of loss of functions, station keeping, and path following, should provide considerable benefits. Although the main experimental contributions of this thesis focus on manipulators and observation-class ROVs, the functionalities and autonomous solutions that are tested will be highly relevant for R-ROVs as well.

1. Introduction

1.1.4 Autonomy in Underwater Operations

In current IMR operations, the human operator receives visual and sensory feedback to help them assess the situation, make decisions, and remotely execute tasks. Making such operations more autonomous increases the demands regarding the sensory systems and software. For sensors in underwater operations, acoustic sonar has for a long time been the preferred option. However, recent technological advances in camera systems and the use of visual aid have made visual systems more attractive, especially for short range navigation. Moreover, visually aided systems may provide higher spatial and temporal resolutions than their acoustic counterparts [10]. Nonetheless, it is not straightforward to use camera systems in underwater environments, especially when paired with robotic systems during partially or fully autonomous operations. The underwater scene is considered one of the most difficult contexts in which to perform optical detection and recognition of features and patterns, partly because underwater image quality is heavily affected by absorption and scattering of light [91, 16].

Technological advances in autonomous systems can improve safety, efficiency, and performance by aiding in decision-making [87]. However, it is difficult to provide a universally applicable definition of an autonomous system. It can be defined as a system that possesses self-governing characteristics, which allow it to perform pre-specified tasks without human intervention. From this perspective, it is clear that AUVs are autonomous. Yet, it can be argued that AUVs that perform pre-specified tasks with some onboard models for low-level decision-making are in fact not autonomous, due to the lack of basic cognitive processes such as evaluation and reasoning. Thus, it can be preferable to define autonomy in terms of different levels of autonomy (LoA). Classifying a system within several LoAs, rather than as a binary value, provides a more flexible description of the system. LoA is often used as a classifier for characterizing the technical requirements for the overall system as well as the tolerable operational risk.

A set of LoAs should be adapted to the relevant applications. In the literature, several sets of LoAs have been suggested; Reese defines 6 levels of autonomous cars [60] while Proud et al. [53] and Perez [52] defines 10 LoAs for aerial vehicles and the National Research Council [45] defines 4 LoAs for aerial vehicles, which were later adopted by Utne et al. [87] for marine applications. It should be noted when only 4 levels are defined it can be problematic to classify various systems within these 4 categories. Small deviations in systems can result in the same system being classified in different levels, which can be a substantial leap in technical requirements and specifications. Systems are therefore able to transition between levels and may have sub-systems classified within different levels [86]. The LoAs adopted from Utne et al. [87] are shown in Table 1.1.

Table 1.1: Levels of autonomy proposed by Utne et al. [87].

LoA	Operation type	System description
1	Remote control	Human-in-the-loop/human operated. The system operates automatically. The operator directs and controls all high-level mission planning functions, which are often preprogrammed. System states, environmental conditions, and sensor data are presented to the operator through a human-machine interface.
2	Management by consent	Human delegated. The system automatically makes recommendations for mission or process actions related to specific functions, and the system prompts the human operator at important points in time for information or decisions. At this level the system may have limited communication bandwidth, including time delays due to distance. The system can perform many functions independently of human control when instructed to do so.
3	Management by exception	Human supervisory control. The system automatically executes mission-related functions when response times are too short for human intervention. The human may override or change parameters and cancel or redirect actions within defined time periods. The operator's attention is only brought to exceptions for certain decisions
4	Highly autonomous operation	Human-out-of-the-loop. The system automatically executes mission- or process-related functions in an unstructured environment, with the ability to plan and re-plan the mission or process. The human may be informed about the progress. The system is independent and "intelligent."

Increasing the LoA in various types of underwater operations in order to reduce cost and risk and increase safety is highly relevant for this thesis. Underwater intervention, for example, where the operator is remotely controlling one or more manipulators with sensors and cameras as guidance, may be a cumbersome task. Increasing the LoA of the intervention system to include autonomous functionalities may go a long way in aiding the operator and optimizing tasks. Such autonomous functionalities can include object centering, object detection and classification, autonomous grasping sequences, and object retrieval. As previously discussed,

1. Introduction

the accuracy of electric manipulators compared makes them more suitable than hydraulic manipulators for meticulous tasks and autonomous functionalities. An overview of the governing components in an autonomous framework for an underwater manipulator system is presented in Figure 1.5. A manipulator system should include the required hardware components to enable an increased LoA, such as relevant sensors, force feedback, and position reading in actuators. The mechanical design should also be sufficiently robust at the required depth. If relevant sensors or actuator abilities are absent, this can to some extent be compensated for in the software. Computer vision techniques or sensor fusion can for example compensate for missing 3D vision or other spatial sensors. However, for some operator requirements, specialized hardware is necessary.

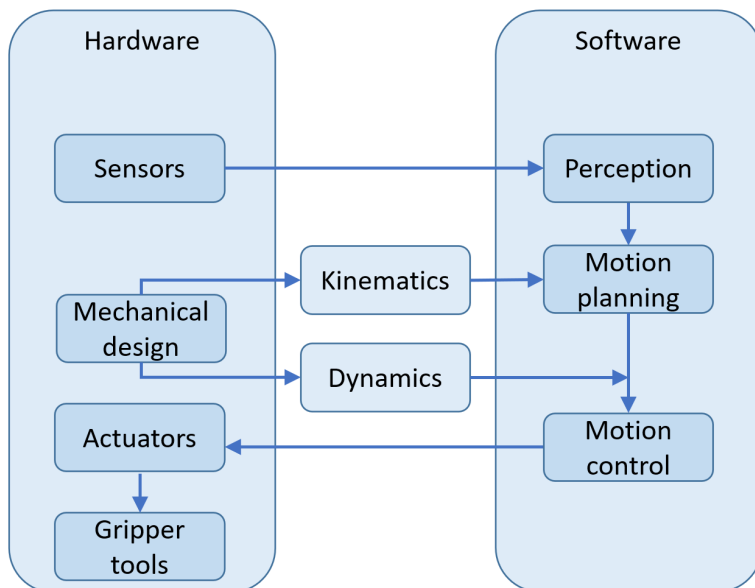


Figure 1.5: An overview of how an autonomous framework can be developed for an underwater manipulator system.

1.2 Research Objectives

The aim of this PhD project is to investigate low-cost solutions for underwater operations, focusing on machine learning and intelligent solutions in intervention tasks and vehicle–manipulator systems. More intelligent solutions allow for increased autonomy, potentially reducing operational costs and risks. In particular, autonomy reduces risks associated with human error. However, it can also introduce new forms of errors. Therefore, it is important to ensure that the design of autonomous systems reduces the overall risks.

The work performed in the course of this PhD project benefits relevant industries. The main beneficiary will initially be the oil and gas industry, where underwater intervention and vehicle–manipulator systems are extensively used. This industry is undergoing changes toward increased autonomy in order to cut costs and improve safety. Remote access to surveillance and control of underwater sites are increasingly necessary and the work done in this project will help achieve this. However, the work should also be relevant to aquaculture, which is a growing industry and should benefit greatly from research on underwater operations and especially applications incorporating machine vision. Monitoring fish welfare, inspection of net, and overall surveillance in fish cages are operations that can potentially be improved by incorporating machine vision. The research conducted in this thesis is organized under 4 main objectives, as follows:

Research Objective 1

Investigate and research the use of machine learning and intelligent software solutions to increase autonomy in underwater intervention tasks.

Increased autonomy in underwater intervention tasks will help decrease costs and the occurrence of human errors. Modeled solutions have long been the standard for automatic and autonomous functionalities; however, the underwater environment is in constant flux, with phenomena that are challenging to model such as marine snow, dark environments, and currents. To model such uncertain environments would require vast amounts of information about the system. Machine learning provides an alternative solution to this—the use of neural networks can help ensure that autonomous systems can cope with the high uncertainty present in underwater environments. Moreover, autonomous solutions are dependent on sensory feedback to react to; however, sensory equipment can be expensive and can increase the weight and complexity of the system. Machine learning applications may be able to extract more relevant information from existing sensor data. This is the main focus of Research Objective 1.

1. Introduction

Research Objective 2

Research and develop vision-based machine learning solutions to improve the visual authenticity of simulated underwater environments to improve the transfer of knowledge from simulated environments to real-world applications.

Training machines in an underwater environment can be extremely time-consuming and error-prone due to the harsh environment. However, models trained in a simulated environment often experience problems when they are applied in the real world, due to the dissimilarities between the two domains. Developing robust techniques for transferring the knowledge between these domains is therefore of great interest to operators in this field. Machine learning offers various benefits, such as increasing the graphical authenticity of the manipulator, mapping the similarities between the domains, and generating such a complex and diverse simulator environment that the real world would be perceived as just another version of the same environment. Investigating such methods to improve the transformation from simulations to real-world operations is the main focus of Research Objective 2.

Research Objective 3

Design, develop, and assemble efficient hardware and software for the SeaArm manipulator, and verify the manipulator capabilities with experimental testing.

Currently, the norm is to use heavy and expensive manipulators driven by hydraulics. There is a gap in the market for small, low-cost electric manipulators. These manipulators should have intuitive controls because they are aimed at low-cost applications for which extensive operator training is not feasible. The SeaArm manipulator is a small electric manipulator developed by Searo AS. This manipulator offers great mobility to perform simple intervention tasks, and the modular design enables it to be reconfigured for different applications. However, to increase the potential for autonomy in the manipulator, a new version with increased capabilities is planned. The new version of the SeaArm manipulator should include a camera housing in the gripper module. This solution is unique and can bring substantial benefits by combining machine vision and underwater manipulation. The kinematic parameters of the manipulator should be mapped in addition to performing pressure testing, determining depth capabilities, lifting capacity, and hydrodynamic behavior, and mapping of current and potential use cases. This is the main focus of Research Objective 3.

Research Objective 4

Conduct experimental testing to verify proposed solutions in underwater manipulation as well as vehicle–manipulator systems.

The knowledge and solutions obtained from Research Objectives 1, 2, and 3 should be used to enhance existing solutions. The obtained theoretical solutions and the potential of the new manipulator should be verified by experimental testing. Some properties of the suggested solutions may need changing when applied in real operations, and experimental testing of the methods and equipment is therefore vital to determine the potential of the solutions. This will be the focus of Research Objective 4.

1.3 Overview of Publications and Contributions

This section presents the publications included in this thesis as well as the key academic and industrial contributions. The list of publications includes both conference papers and journal articles and they are listed in chronological order by publication date.

1.3.1 Contributions from Articles

Article 1—Conference Paper

M. Skaldebø, A. S. Muntadas, and I. Schjøberg, “Transfer Learning in Underwater Operations,” OCEANS 2019 - Marseille, 2019, pp. 1–8, doi: [10.1109/OCEANSE.2019.8867288](https://doi.org/10.1109/OCEANSE.2019.8867288).

- ◇ Contribution 1: Collection of two image datasets of underwater structures including real photos and images of rendered environments from simulations and computer-aided design software.
- ◇ Contribution 2: Training and verification of CycleGAN models for the two image datasets. The models demonstrated domain transfer from a simulated environment to photo-realistic images, and an application of a transfer learning framework to vision-based underwater operations.
- ◇ Contribution 3: Suggested solutions for reducing the reality gap in the machine learning process by the use of the proposed models.

Article 2—Conference Paper

M. Skaldebø, B. O. A. Haugaløkken, and I. Schjøberg, “Dynamic Positioning of an Underwater Vehicle using Monocular Vision-Based Object Detection with Machine Learning,” OCEANS 2019 MTS/IEEE SEATTLE, 2019, pp. 1–9, doi: [10.23919/OCEANS40490.2019.8962583](https://doi.org/10.23919/OCEANS40490.2019.8962583).

- ◇ Contribution 4: Collection of a large image dataset of an object of interest including images and corresponding labels, and training of an object detection model using state-of-the-art machine learning algorithms.
- ◇ Contribution 5: Development of a method for efficiently labeling large image datasets based on color detection.
- ◇ Contribution 6: Development of a method for extracting spatial information from 2D images using a scaling function in combination with object detection.

1.3. Overview of Publications and Contributions

- ◇ Contribution 7: Design of a dynamic positioning system that keeps the underwater vehicle at a desired position and orientation relative to a known object.
- ◇ Contribution 8: Demonstration and experimental validation of the proposed methods in a laboratory pool.

Article 3—Journal Article

B. O. A. Haugaløkken, M. B. Skaldebø, and I. Schjølberg, “Monocular vision-based gripping of objects,” *Robotics and Autonomous Systems*, Volume 131, 2020, 103589, ISSN 0921-8890, doi: [/10.1016/j.robot.2020.103589](https://doi.org/10.1016/j.robot.2020.103589).

- ◇ Contribution 9: Design of a dynamic positioning system for a vehicle to maintain a desired position and orientation relative to an object detected through a monocular vision-based system based on object detection with spatial extraction of features.
- ◇ Contribution 10: Development of a task definition and kinematic control system for the end-effector that allows it to move to an object’s location.
- ◇ Contribution 11: Experimental testing that proves the effectiveness of the proposed solution for grasping the object with a low cost UVMS.

Article 4—Conference Paper

M. B. Skaldebø, B. O. A. Haugaløkken, and I. Schjølberg, “SeaArm-2 - Fully electric underwater manipulator with integrated end-effector camera,” *2021 European Control Conference (ECC)*, 2021, pp. 236–242, doi: [10.23919/ECC54610.2021.9655121](https://doi.org/10.23919/ECC54610.2021.9655121).

- ◇ Contribution 12: Presentation of a new small, modular, flexible, and low-cost electric underwater manipulator.
- ◇ Contribution 13: Autonomous detection, movement, and grasping of known objects in unknown positions.
- ◇ Contribution 14: Experimental validation of the functionality of the manipulator and the autonomous fixed-base grasping operation.

1. Introduction

Article 5—Journal Article

A. Sans-Muntadas, M. B. Skaldebø, M. C. Nielsen, and I. Schjølberg, “Unsupervised Domain Transfer for Task Automation in Unmanned Underwater Vehicle Intervention Operations,” *IEEE Journal of Oceanic Engineering*, vol. 47, no. 2, pp. 312–321, April 2022, doi: [10.1109/JOE.2021.3126016](https://doi.org/10.1109/JOE.2021.3126016).

- ◇ Contribution 15: Presentation of a method for real-time detection of an underwater object using real footage and a neural network trained on images produced by a generative adversarial network (GAN).
- ◇ Contribution 16: Demonstration of how overtraining the adversarial network results in lower detection scores.
- ◇ Contribution 17: Experimental validation of the approach to automating some of the control tasks using an observation-class ROV.

Article 6—Journal Article

M. B. Skaldebø, B. O. A. Haugaløkken, and I. Schjølberg, “Autonomous underwater grasping using a novel vision-based distance estimator,” *International Journal of Mechanical Engineering and Robotics Research*. *Status: Submitted (August 2022)*.

- ◇ Contribution 18: Training of an object detector and object tracker using state-of-the-art neural networks on a self-generated image dataset.
- ◇ Contribution 19: Development of a distance estimator capable of estimating object size and distance to objects of unknown size.
- ◇ Contribution 20: Verification of the object detector, tracker, and developed distance (and size) estimator in a laboratory pool with the SeaArm-2 manipulator.
- ◇ Contribution 21: Verification of the developed system’s ability to perform autonomous grasping of underwater objects in a laboratory pool.

Article 7—Conference Paper

M. B. Skaldebø, I. Schjølberg, and B. O. A. Haugaløkken, “Underwater vehicle manipulator system (UVMS) with BlueROV2 and SeaArm-2 manipulator,” 41st International Conference on Ocean, Offshore & Arctic Engineering (OMAE 2022) - Hamburg 2022, doi: [10.1115/OMAE2022-79913](https://doi.org/10.1115/OMAE2022-79913).

1.3. Overview of Publications and Contributions

- ◇ Contribution 22: Development of a digital twin for a UVMS consisting of the BlueROV2 and the SeaArm-2.
- ◇ Contribution 23: Presentation of a simulated laboratory environment where the digital twin can be deployed.
- ◇ Contribution 24: Validation of inherent time delays in the system through simple task tracking.

Article 8—Conference Paper

M. B. Skaldebø and I. Schjøberg, “Dynamic Bayesian Networks for Reduced Uncertainty in Underwater Operations,” 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS 2022), *Status: Presented*.

- ◇ Contribution 25: Discussion of the importance of risk assessment in underwater operations and how increased autonomy brings higher requirements for risk assessment.
- ◇ Contribution 26: An overview of existing Bayesian belief network (BBN) frameworks.
- ◇ Contribution 27: The development of ABBA, a dynamic approach to designing BBNs with graphical visualization and adaptable source code for implementation with other systems, along with discussion about the novelties of ABBA compared to existing solutions
- ◇ Contribution 28: A case study demonstrating how ABBA can be used to reduce uncertainties in a UVMS.

Article 9—Conference Paper

A. A. Transeth, I. Schjøberg, A. M. Lekkas, P. Risholm, A. Mohammed, M. B. Skaldebø, B. O. A. Haugaløkken, M. Bjerkgeng, M. Tsiourva, and F. Py, “Autonomous subsea intervention (SEAVENTION),” 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS 2022) - Lyngby 2022, *Status: Presented*.

- ◇ Contribution 29: Presentation and discussion of the main results and recent developments on UUV autonomy from the SEAVENTION project.
- ◇ Contribution 30: An overview of the state of the art in UUV intervention with the status and trends of UUVs in IMR in the oil and gas industry.

1. Introduction

Article 10—Journal Article

M. B. Skaldebø, I. Schjølberg, and B. O. A. Haugaløkken, “System integration of underwater vehicle manipulator system (UVMS) for autonomous grasping,” *Journal of Marine Science and Technology*. *Status: Submitted (September (2022))*.

- ◇ Contribution 31: System integration of a three-machine low cost UVMS.
- ◇ Contribution 32: Autonomous grasping experiments in a simulated environment, verifying both the software integration and the simulator capabilities.
- ◇ Contribution 33: Implementation of an extended camera system, where both the system integration and the camera system are validated in laboratory experiments.

1.3.2 Other Contributions

- ◇ Contribution 34: The design and development of the SeaArm-2 manipulator in partnership with the company Searo AS. The candidate has held a pivotal role in the company for the development of the new version of the manipulator arm. Searo AS was temporarily dissolved due to the retreat of financial partners.
- ◇ Contribution 35: Assembly of the SeaArm-2 manipulator with electrical wiring and soldering, in addition to building software packages for the manipulator with proficient control and camera system capabilities.

1.4 Structure of the Thesis

This thesis is divided into three parts: the main report, a collection of articles, and a list of previously published theses at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU). Part I, the main part of the thesis, discusses the use of visual aid in underwater robotics, the control and kinematics of low-cost manipulators, and the design and control of UVMSs in research and preliminary IMR operations using low-cost equipment. Part I is structured as follows:

- Chapter 1 describes the background, research objectives, and contributions of this thesis.
- Chapter 2 describes the state of the art and summarizes the main methods used in the thesis.
- Chapter 3 presents the facilities and equipment used for experimental testing.
- Chapter 4 summarizes the research results of the included articles.
- Chapter 5 presents the final conclusions of the thesis and suggests further research avenues.

A collection of ten articles is enclosed in Part II, and a list of previously published theses at the Department of Marine Technology is enclosed in Part III.

Chapter 2

State of the Art

This chapter presents the state of the art including some of the most important methodologies used in the thesis. The emphasis is on the main part of the thesis, which is the use of visual aid in underwater robotics and control and identification of underwater manipulator systems and UVMSs.

2.1 Visual Aid in Underwater Robotics

This section is dedicated to the use of visual aid in underwater robotics and focuses on the implementation of machine vision and computer vision solutions. This section provides clarification for Research Objectives 1 and 2, and is relevant for Articles 1, 2, 3, 4, 5, 6, and 10.

2.1.1 Preliminary Discussion

Visual aid in the underwater environment has to contend with challenges that are non-existent at surface level, including limited visibility, light scattering, low contrast, marine snow, and object occlusion. Thus, other sensors are often used instead of, or in cooperation with, visual sensors. However, with the promise shown by visual aid systems, they could play an important role in intelligent solutions for multi-sensory systems.

Visual aid can be described in terms of both machine vision and computer vision. The line between the two has become blurred in recent years, given the accelerated development in vision systems as a whole. Traditionally, machine vision is the simpler technique, designed to generate an image from a scene and use low-level processing to analyze and make simple, automated decisions. Machine vision includes the image system in its entirety involving illumination, housing, camera,

2. State of the Art

lens, frame grabber, compression chip and computer software for processing, low-level analysis, and presenting the image. Computer vision involves more in-depth analysis and can identify, predict, and observe trends. Moreover, computer vision focuses solely on the process of interpreting individual images or streams of images, such as object detection and recognition, object and video tracking, motion estimation, image restoration, domain transfer, and scene reconstruction [44]. This means the individual image or video does not necessarily have to be made up of photos from a camera, but can consist of thermal images, sonar images, or images captured by other sensors.

2.1.2 Machine Vision

Machine vision inhabits the attributes of our most trusted sensor, the human eye. It transforms information collected from acoustic sonars, pressure sensors, or other sensors into a medium perceivable by humans, often in the form of visual information. Sonar used in navigation is translated into distances on a 2D map, where humans can relate it to distances seen by the human eye. Acoustic obstacle avoidance is achieved by identifying shapes and nearby obstacles from sonar which is communicated as visual images to the operator. Moreover, machine vision communicates relevant and clean information in its natural state and the spatial and temporal resolution of visual information is unprecedented compared to its acoustic counterpart [10].

Underwater machine vision is a broad field and includes systems with a wide range of specifications. Machine vision can be divided into two main categories, monocular vision and stereo vision. A less common middle ground, referred to as 2.5D vision, is also used to some extent; it reconstructs 3D environmental data from monocular images [91]. Monocular vision uses one camera to generate image data, where the output is 2D data information with a variety of resolutions is dependent on the components involved. Stereo vision, on the other hand, typically involves two or more cameras to generate depth images. Spatial information is extracted using epipolar geometry on two or more disparate images captured by cameras. This includes matching correlated pixels or groups of pixels in the disparate images and calculating the distance between conjugate points. This distance is referred to as the disparity and is given by

$$d = u_L - u_R, \quad (2.1)$$

where u_L and u_R are the projections of a real-world point P in images acquired by the left and right cameras, respectively. In this way, stereo cameras include spatial information in addition to the 2D data, generating disparity maps with 3D data information.

Stereo vision can further be divided into passive stereo vision and active stereo vision [51, 14]. Passive stereo vision has fixed camera positions and locates matching pixels by assuming the object surface reflects identical light intensity to each camera. This assumption is often invalid, especially in the underwater scene where scattering of light is a common phenomenon, thus causing mismatching that leads to flawed spatial information. Active stereo vision applies dynamic motion of the cameras, and often relies on a projected light source to enable matching of correlating pixels in the disparate images. An external projected light source aids the pixel matching; however, mismatching remains a major problem even in active stereo vision. Deep learning has been suggested as a tool to aid in correlation matching for stereo vision. Deep learning techniques, such as object detection, can be used to locate correlating objects in the disparate images locate matching correlations. Using the object detection network EfficientDet [80], Chen et al. [15] were able to match correlating objects and calculate the distance to objects with an error of around 1.88–2.72 mm for distances between 380 mm and 680 mm. This yields a mean distance error of 0.44%.

2.1.3 Computer Vision

Computer vision can be thought of as a more advanced form of machine vision. Methods involving identification, prediction, learning, neural networks, and other computationally expensive operations can be defined within this type of visual aid. Such methods may include classification [38], salient feature detection [17], and object detection [54, 43]. Object detectors built with neural networks can include a large variety of properties depending on the specific application. Vision-based detectors commonly consider edges, color differences, and optical flow. Such properties are frequently considered in combination with one another when building neural networks.

Neural networks can be defined as artificial networks inspired by the natural neural networks in the human brain [2]. Neural networks have successfully been implemented in games [20], handwriting recognition [3], and even explosives detection [89]. Moreover, neural networks provide a method for defining a system too complex to be represented by a simple model, for example in image recognition, object detection, and other systems influenced by uncertainty. As opposed to computing a model capable of detecting specific objects, designing a neural networks requires training on relevant data. In the case of an object detection system, the appropriate training data would be a dataset consisting of images of the relevant object with corresponding labels. Labels serve as a ground truth value that informs the network of the correct answer during training. This form of learning, where the network first predicts an object position within the frame before it is told the true value, is referred to as supervised learning. The objective of this learning algorithm

2. State of the Art

is in general to learn a mapping $y = f(x)$. Other forms of learning algorithm, such as unsupervised learning and reinforcement learning, are also popular. The objective of unsupervised learning is to learn useful properties and structures of a dataset, while reinforcement learning uses a trial-and-error approach by interacting with an environment, as opposed to learning from a dataset.

Object Detection and Tracking

Object detection is commonly separated into two types, region proposal networks and regression-or classification-based methods. Region proposal networks follow the traditional object detection procedure by identifying region proposals and classifying the proposals into object categories. Such methods employ, to some extent, the same methodologies as the human brain, performing an initial scan of a scenario before separating the information into regions of specific interest. The second type follows a classifier-based approach or a regression approach. Both methodologies share the overall objective of learning the mapping f in $y = f(x)$. The regression- and classification-based methods are separated by the type of output. Classification approaches aim to learn the mapping to a discrete or categorical output, while regression approaches aim to learn the mapping to a continuous or numerical output.

Object detection aims to locate objects within a camera frame, and classifies objects by placing bounding boxes surrounding the objects. Object detectors usually operate with different metrics to analyze the performance of the object detection system. Typical metrics are the average precision (AP) and the mean average precision (mAP). The AP score represents how well the detector can recognize an object class, and the mAP score is the mean AP for all classes. These are based on other sub metrics such as intersection over union (IoU), recall rate, and precision score. IoU is demonstrated in Figure 2.1: it indicates the overlap of the predicted bounding box to the ground truth box. The ground truth box is referred to as the label. Recall rate indicates how many objects were detected in a validation set, meaning that a low recall rate represents a detector that overlooked a considerable number of objects. The precision score indicates how many of the predicted objects were in fact, so that a low precision score represents a detector that falsely detects objects that are not there. To determine if a detection is successful, the IoU between the prediction and label needs to be higher than a certain threshold. This threshold varies but is typically 0.5 for normal AP validation.

The well-known You Only Look Once (YOLO) algorithm [56] is one of the most successful real-time object detection algorithms. In contrast to several other object detection methods, YOLO identifies the detection problem as a regression problem as opposed to a classifier-based approach, and the algorithm consist of a single

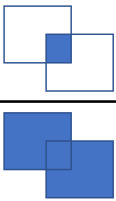
$$\text{IoU} = \frac{\text{area of overlap}}{\text{Area of union}}$$


Figure 2.1: Intersection over union (IoU).

neural network. This specifications provides a framework able to process images in real time at 45 frames per second. In addition, the algorithm generalizes well to different domains, which makes it ideal for new applications. YOLO was first introduced in 2016 by Joseph Redmond, who later published the versions YOLO9000 [57] in 2017 and YOLOv3 [58] in 2018. When introducing the second version, Redmond also introduced Darknet [55], more specifically Darknet-19, as the underlying architecture of his YOLO9000 implementation. Darknet-19 is an open-source convolutional neural network consisting of 19 layers. The third version, YOLOv3, came with an improved backbone of Darknet, the Darknet-53 network. Darknet-53 increased the number of layers to 53 and introduced the use of residuals.

Even though Redmond stopped after three versions of YOLO, other researchers carried on his work and developed YOLOv4 [9], YOLOv5 [30] and PP-YOLO [37]. These are all networks that improved upon the results of YOLOv3. Similarly to other object detectors, the YOLO algorithms predict an object position by placing a bounding box surrounding the object within the image frame. Figure 2.2 illustrates how YOLOv5 is applied to an image and is used to detect a 3D-printed fish object. The object is detected by a bounding box and a corresponding confidence score. The confidence score represents the model's certainty in the detection.

Object detectors can effectively locate relevant objects within a frame. However, detecting objects in individual frames means that temporal information is neglected. Tracking objects over consecutive image frames can be achieved using an object tracker, which has also proven to facilitate robustness against temporary occlusions and to improve stability in detection. Moreover, tracking objects over consecutive images enables identification of individual objects. This improves computer vision capabilities and allows for new features such as counting fish in fish cages, and tracking and following specific pipelines. A popular and easy-to-implement tracker is the DeepSORT tracker developed by [90], which is an extension of the popular Simple Online and Realtime Tracking (SORT) algorithm [5]. Similar to SORT, the DeepSORT tracker uses a Kalman filter and the Hungarian algorithm for the track-

2. State of the Art

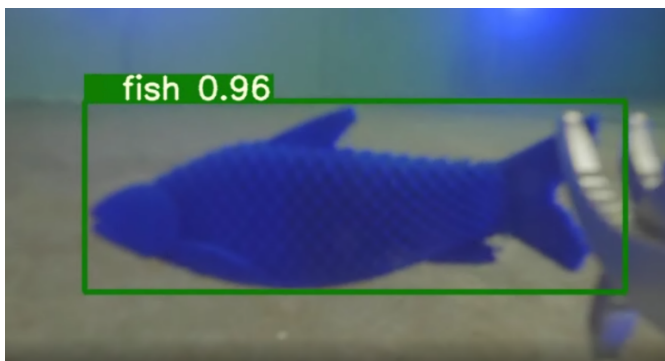


Figure 2.2: A YOLOv5 model applied to detect 3D-printed fish objects. The object is detected with a bounding box and a confidence score.

ing 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.

2.5D Vision

In object detection, the real-time pose of objects is often of interest, thus stereo cameras are often used [65, 75, 92]. Labeling data is often necessary in machine learning applications such as object detection, and 3D images can be complicated to process and labeling such images is difficult. Processing standard monocular camera images and labeling ordinary RGB 2D images is much less complicated. However, this requires alternative methods for extracting spatial features. Underwater object detection using monocular camera has previously been investigated by Chen et al. [18], who used visual features such as color and intensity as well as investigating light transmission information. However, spatial features were not considered. Based on 2D image data, Zhao et al. [94] presented a 3D object detection method. They modified the aforementioned YOLO algorithm to receive input images from different angles and reconstruct a 3D representation with a 3D bounding box locating the object.

Another popular method for generating 3D data from 2D images is the use of geometric relations and mathematical formulations. Using the intrinsic parameters of the camera, such as field of view and lens focal length, the relationship between an image and a real scene is explained by the pinhole model [39]. Figure 2.3 shows the geometric relationship between a box at point (px_y, px_z) with height and width of H_{bb} and W_{bb} in the camera frame, and the corresponding box in 3D space at point (X, Y, Z) , with height H and width W . The coordinates of the box in 3D

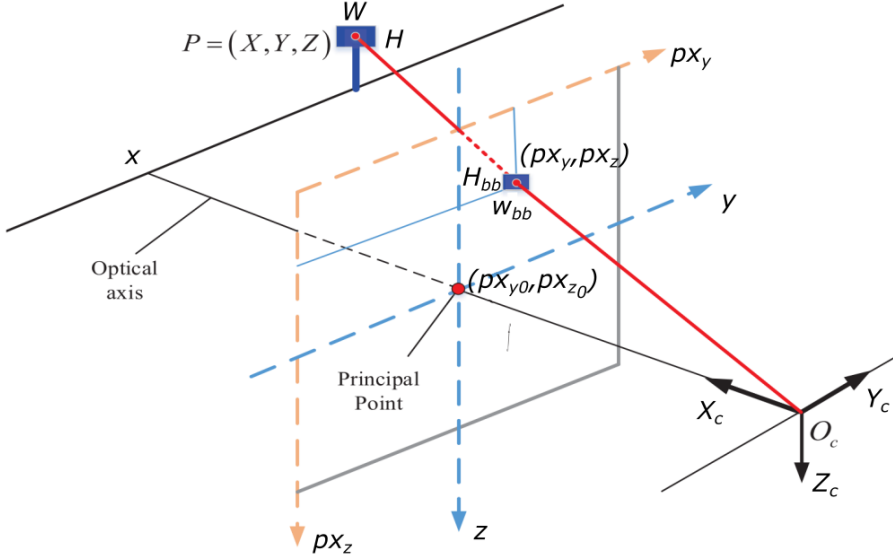


Figure 2.3: Geometric relations between image frame and 3D space [39].

space are given by

$$\begin{aligned}
 X &= (W/w_{bb} + H/h_{bb})f/2 \\
 Y &= X(px_y - px_{y0})dy \\
 Z &= X(px_z - px_{z0})dz,
 \end{aligned} \tag{2.2}$$

where f is the camera lens focal length, and dy and dz are representations of the physical dimensions of each pixel in the y - and z -directions in the image plane, respectively, and are calculated by

$$\begin{aligned}
 dy &= \frac{2X \tan(FOV_y/2)}{PX_y} \\
 dz &= \frac{2X \tan(FOV_z/2)}{PX_z}
 \end{aligned} \tag{2.3}$$

where FOV_y and FOV_z are the field of view of the camera in the horizontal and vertical directions, respectively, and PX_y and PX_z are the total number of pixels in image frame in the y - and z -directions. Relating this approach to object detection, the bounding box around a detected object is used to represent the object size in the image frame.

2. State of the Art

Transfer Learning

Transfer learning is a method within machine learning where the knowledge gained from solving one problem is reapplied to solve a different but related problem. Transfer learning might be seen as a basic ability in the context of human intelligence, but it can be complicated to achieve in machine learning. The idea is to make machines able to transfer knowledge between different domains and execute different but related tasks, or alternatively the same tasks in different environments. Regarding underwater operations, the latter goal is particularly interesting. Training machines in an underwater environment can be extremely time consuming and error-prone due to the harshness of the environment. However, if machines are trained exclusively in simulations, the transfer of knowledge to the real world could also result in failure. Generating robust techniques for transferring the knowledge between domains is therefore of immense interest for operators in this market.

Modeling a photorealistic underwater scene is immensely time-consuming due to the computational cost of raytracing and, more importantly, the vast number of phenomena and effects that need to be modeled. Such phenomena include biofouling, light caustics, and marine snow particles. Instead of focusing on modeling a perfect environment from the ground up, an interesting approach is to introduce transfer learning to learn a general mapping between the modeled domain and the real-world domain. This avoids the considerable work of modeling all visual effects. A breakthrough in transfer learning came in 2014 when generative adversarial networks (GANs) were introduced by Ian Goodfellow[26]. A GAN consists of two networks, a generator and a discriminator, where the generator aims to produce content, while the discriminator determines the level of authenticity of the content. The end-goal of GANs is for the generator network to learn the general mapping between the source domain and the target domain such that the discriminator will be unable to distinguish real samples from fakes. GANs have shown promising results in numerous applications, for example in face generation [70], text-to-image translation [59], image-to-image translation [35], and super resolution [34].

A GAN is based on differentiable generator networks, which transform samples of latent variables z to samples x (or distributions thereof) using a differentiable function $G(z)$. A GAN works on the principle of comparing generator networks with discriminator networks. The generator produces samples $x = G(z)$, and the discriminator attempts to determine if the samples are produced by the generator or if they come directly from the training set. The discriminator produces a probability given by $D(x)$, indicating the probability that x is a real sample rather than a fake sample produced by the generator. The end-goal of a GAN is that the discriminator will be unable to distinguish the real samples from the fakes and so will produce a constant probability of 0.5.

A GAN can in many ways be perceived as a zero-sum game, that is, a game in which the gains and losses of the participants are exactly balanced and sum to zero. Consider the value function $v(G, D)$ that determines the payoff of the discriminator, then the generator receives $-v(G, D)$ as its own payoff. During learning, both participants have a desire to maximize their payoffs, until the learning process reaches convergence,

$$G^* = \arg \min_G \max_D v(G, D). \quad (2.4)$$

In GANs, the value function is typically represented by

$$v(G, D) = \mathbf{E}_{x \sim p_{\text{data}}} \log D(x) + \mathbf{E}_{x \sim p_{\text{model}}} \log(1 - D(x)). \quad (2.5)$$

This function depends on the expected value of $D(x)$ with respect to the training data and the model. The discriminator will focus on learning to correctly classify samples as real or fake, while the generator will simultaneously try to generate samples that look as real as possible to fool the discriminator. The two participants learn in parallel to each other. This model can be highly under-constrained, but there are several published methods and frameworks which help to deal with this problem.

2.2 Manipulator Kinematics and Control

This section is dedicated to manipulators and focuses on defining manipulator kinematics and controls. Manipulators, also referred to as robotic arms, can operate independently or as part of a combined vehicle–manipulator system. The section contributes to Research Objectives 3 and 4, and is relevant to Articles 3, 4, 6, 7, and 10.

2.2.1 Preliminary Discussion

If the purpose of underwater vehicles is to replace the human presence at great depths, underwater manipulators are there to replace humans' intervention capabilities. Hence, manipulators' physical appearance and capabilities tend to mimic the human arm. Disregarding this similarity in appearance, manipulators exist in all sizes and shapes, from small electric manipulators with minimal reach and lifting capacities to large heavy hydraulic manipulators reaching over 2 m and capable of lifting more than half a metric ton. A substantial review of existing underwater manipulators was published by Sivčev et al. [76], in which both hydraulic and electric manipulators of different sizes were considered. The review summarized the state of the art and highlighted the vast number of control approaches and the diversity in manipulator solutions. In addition, the authors identified the great potential in the field. Introducing machine vision combined with underwater manipulation can immensely improve existing sub-sea manipulation systems. Moreover, torque control algorithms and combined vehicle–manipulator systems hold great potential. These topics are highlighted as fields of research that call out for more exploration.

2.2.2 Manipulator Kinematics

A manipulator consists of joints with connecting links, and the number of links determines the DOF of the manipulator. The DOF is used to establish the spatial capabilities of the manipulator and can be obtained from the Chebychev–Grübler–Kutzbach criterion

$$\text{DOF} = \lambda(N - J - 1) + \sum_{i=1}^J F_i, \quad (2.6)$$

where N is the total number of links including the base link, J is the total number of joints connecting two consecutive links, F_i is the DOF at the i^{th} joint, and $\lambda = 6$ for 3D motion or 3 for planar motion [25]. To further determine the geometries of the manipulator, it is necessary to establish a transformation between the joints. This can be achieved by using the Denavit–Hartenberg (DH) convention, which is widely used in the kinematic description of manipulators. The DH convention describes the relations of coordinate frames, and by placing local coordinate frames in each joint, we find the relation between all joints [21]. Using this convention,

we can solve two essential problems of manipulator kinematics—direct and inverse kinematics. Direct kinematics concerns the situation in which joint angles are the input and position and orientation of the last link (or end-effector) are the outputs. Inverse kinematics concerns the opposite problem of finding unknown joint angles when the position and orientation of the last link are known.

Transformation Matrices

Both direct and inverse kinematics concern the configuration of a body frame $\{b\}$ represented in a fixed space frame $\{s\}$ by specifying the position p of the frame $\{b\}$ in $\{s\}$ coordinates. This configuration can be gathered in a single 4×4 matrix T , called the homogeneous transformation matrix,

$$T = \left[\begin{array}{c|c} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \hline \mathbf{0}_{1 \times 3} & 1 \end{array} \right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_1 \\ r_{21} & r_{22} & r_{23} & p_2 \\ r_{31} & r_{32} & r_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.7)$$

where \mathbf{R} is the 3×3 rotational matrix and \mathbf{p} contains the translations in the x , y , and z directions. The bottom row is included to simplify matrix operations. Transformation matrices satisfy properties corresponding with rotation matrices

$$\begin{aligned} \mathbf{T}^T &= \mathbf{T}^{-1}, \\ \det \mathbf{T} &= 1. \end{aligned} \quad (2.8)$$

Transformation matrices have three common uses: (1) representing a rigid body configuration, (2) changing the reference frame of a vector or a frame, and (3) displacing a vector or a frame. Moreover, the manipulator end-effector can be represented in the base frame of the manipulator. Regarding vehicle–manipulator systems, if the manipulator is attached to an ROV, it can further be represented in a body-fixed reference frame of the ROV or a global inertial frame, by expanding the total coordinate frames and corresponding transformation matrices.

The transformation matrix of a manipulator can be established using the DH convention, where the DH parameters determine the parameters in the matrix. The relationship between the transformation matrix and the DH parameters is given by

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.9)$$

where

2. State of the Art

T_i^{i-1} is the transformation matrix between joints $i - 1$ and i ,

d_i is the distance from x_{i-1} to x_i along z_{i-1} ,

θ_i is the angle between x_{i-1} and x_i about z_{i-1} ,

a is the distance from z_{i-1} to z_i along x_i , and

α is the angle between z_{i-1} and z_i around x_i .

Defining n local coordinate systems in the manipulator gives the complete transformation matrix from the base to the end-effector as

$$T_{\text{ec}}^{\text{base}} = T_1^0 T_2^1 T_n^{n-1}. \quad (2.10)$$

Jacobian Matrix

While the homogeneous transformation matrix explains the relationship between Cartesian coordinates and joint positions, the Jacobian matrix explains the relationship between Cartesian velocities and joint velocities. The relations between the velocities are represented by

$$\dot{X} = J\dot{q}, \quad (2.11)$$

where \dot{X} are the end-effector velocities, J is the Jacobian matrix, and \dot{q} are the joint velocities. A 6-DOF manipulator kinematics generates the coherence (expanded from (2.11))

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & J_{66} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} \quad (2.12)$$

In most cases, (2.11) cannot be solved for \dot{q} uniquely. The Jacobian may not be square or invertible, meaning that J^{-1} cannot be computed. Moreover, solving $\dot{q} = J^{-1}\dot{X}$ could cause problems if J is nearly singular [12]. There are several proposed solutions to this problem, including replacing the Jacobian inverse with a transpose Jacobian or pseudoinverse or using damp least squares. The advantage of the pseudoinverse is that it is defined for all matrices, even non-square and rank-deficient matrices. The downside of the pseudoinverse is that it tends to have

2.2. Manipulator Kinematics and Control

stability problems close to singularities. The damped least squares method solves the singularities problem.

To find the pseudoinverse with damping least squares, $\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{X}}$ should be satisfied, where \mathbf{J}^\dagger is the pseudoinverse of the Jacobian. The standard pseudoinverse is computed by finding the best fit solution to (2.11), that is, minimizing the error $\mathbf{e} = \mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{X}}$. Taking the square of this error, the objective is to minimize the velocity error

$$V(\dot{\mathbf{q}}) = \|\mathbf{e}\|^2 = \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{X}}\|^2, \quad (2.13)$$

with respect to the joint velocities $\dot{\mathbf{q}}$. However, to minimize (2.13) requires $\mathbf{J}^T \mathbf{J}$ to be invertible, which is only valid when \mathbf{J} has full rank. This is not the case close to singularities, and thus the standard pseudoinverse will behave poorly around singularities. By introducing damp least squares [88, 33], instead of minimizing the velocity error in (2.13), the objective becomes to minimize

$$V(\dot{\mathbf{q}}) = \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{X}}\|^2 + \lambda \|\dot{\mathbf{q}}\|^2, \quad (2.14)$$

where λ is a non-zero damping coefficient. To solve this, (2.14) can simply be expanded to

$$V(\dot{\mathbf{q}}) = (\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{X}})^T (\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{X}}) + \lambda \dot{\mathbf{q}}^T \dot{\mathbf{q}} \quad (2.15)$$

$$= (\mathbf{J}\dot{\mathbf{q}})^T \mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{X}}^T \mathbf{J}\dot{\mathbf{q}} - (\mathbf{J}\dot{\mathbf{q}})^T \dot{\mathbf{X}} + \dot{\mathbf{X}}^T \dot{\mathbf{X}} + \lambda \dot{\mathbf{q}}^T \dot{\mathbf{q}} \quad (2.16)$$

$$= \dot{\mathbf{q}}^T \mathbf{J}^T \mathbf{J}\dot{\mathbf{q}} - 2\dot{\mathbf{q}}^T \mathbf{J}^T \dot{\mathbf{X}} + \dot{\mathbf{X}}^T \dot{\mathbf{X}} + \lambda \dot{\mathbf{q}}^T \dot{\mathbf{q}} \quad (2.17)$$

To minimize the velocity function with respect to $\dot{\mathbf{q}}$, the partial derivative is set equal to 0,

$$\frac{\partial V(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = 2\mathbf{J}^T \mathbf{J}\dot{\mathbf{q}} - 2\mathbf{J}^T \dot{\mathbf{X}} + 2\lambda \dot{\mathbf{q}} = 0. \quad (2.18)$$

Now $\dot{\mathbf{q}}$ can be isolated through (2.18)

$$\mathbf{J}^T \mathbf{J}\dot{\mathbf{q}} + \lambda \dot{\mathbf{q}} = \mathbf{J}^T \dot{\mathbf{X}} \quad (2.19)$$

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})\dot{\mathbf{q}} = \mathbf{J}^T \dot{\mathbf{X}} \quad (2.20)$$

$$\dot{\mathbf{q}} = \mathbf{J}^T (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1} \dot{\mathbf{X}} \quad (2.21)$$

The pseudoinverse is thus defined as

$$\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1}. \quad (2.22)$$

The pseudoinverse is extensively used in the manipulator control algorithms applied in this thesis.

2.3 Underwater Vehicle Manipulator Systems

This section focuses on system integration and basic control strategies for UVMSs. The section contributes to Research Objective 4, and is relevant to Articles 2, 3, 7, 8, and 10.

2.3.1 Preliminary Discussion

As discussed in Chapter 1, attaching a manipulator to a UUV provides a moving base for the manipulator, and enables autonomous intervention for tasks such as pipelines operations, subsea panel intervention, fixing holes in fish cages, underwater construction work, and collecting organisms such as plants and fish. To control a UVMS, especially when introducing autonomous functionalities, geometrical relations of the system should be clearly defined. Positions and velocities of a body are defined with points and vectors relative to a reference frame. In this section, the North-East-Down (NED) frame is considered as the global reference frame, and a local body-fixed reference frame (known as the body frame) is fixed with respect to the vehicle body. Positions, orientations, and velocities of the vehicle, along with the motion of the body frame, are described relative to the NED frame.

2.3.2 Equations of Motion

The UVMS model presented here is based on that of Schjøberg and Fossen [66], where the statistical properties of the UVMS are described by the position $\boldsymbol{\eta} = [\boldsymbol{\sigma}^T \boldsymbol{\Theta}^T]^T$ and velocity $\boldsymbol{v} = [\boldsymbol{v}^T \boldsymbol{\omega}^T]$. The position $\boldsymbol{\sigma} = [x, y, z]^T$ and the orientation in terms of Euler angles $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$ of the vehicle body frame are expressed with respect to the global NED frame. Moreover, $\boldsymbol{v} = [u, v, w]^T$ represents the linear velocity and $\boldsymbol{\omega} = [p, q, r]^T$ represents the angular velocity of the vehicle body frame. The manipulator states are described by the joint angles $\boldsymbol{q} = [q_1, q_2, q_3]$ and the joint angular velocities $\dot{\boldsymbol{q}} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]$. The manipulator's end-effector pose $\boldsymbol{\eta}_{ee} = [\boldsymbol{\sigma}_{ee}^T \boldsymbol{\Theta}_{ee}^T]^T$, where $\boldsymbol{\sigma}_{ee}^T = [x_{ee}, y_{ee}, z_{ee}]^T$ is the position and $\boldsymbol{\Theta}_{ee} = [\phi_{ee}, \theta_{ee}, \psi_{ee}]^T$ is the orientation in terms of the Euler angles, is defined relative to the vehicle body frame. Moreover, the position and orientation of the end-effector reference frame relative to the vehicle body frame are found using the DH convention as described in Section 2.2.2. The combined system is defined as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_R(\boldsymbol{\eta})\boldsymbol{v} \quad (2.23)$$

$$\dot{\boldsymbol{\eta}}_{ee} = \boldsymbol{J}_e(\boldsymbol{q}, \boldsymbol{\eta})\boldsymbol{\zeta}, \quad (2.24)$$

where $\boldsymbol{\zeta} = [\dot{\boldsymbol{q}}^T \boldsymbol{v}^T]^T$ represents the velocity of both the vehicle and the manipulator. Furthermore, \boldsymbol{J}_R represents the Jacobian of the UVMS matrix and $\boldsymbol{J}_e = [\boldsymbol{J}_1(\boldsymbol{q}) \boldsymbol{J}_2(\boldsymbol{\eta})]$ represents the Jacobian relating the end-effector time derivative to $\boldsymbol{\zeta}$.

2.3.3 Dynamic Positioning of Underwater Vehicle Manipulator Systems

A DP system is expected to keep a vehicle within specified trajectory limits, and for underwater vehicles this can apply to control in all 6 DOFs. This, combined with the fact that the hydrodynamic parameters in a dynamic UVMS model are difficult, or near impossible, to acquire accurately, makes DP of underwater vehicles a challenging task. One suitable method for applying DP to an UVMS is a sliding mode controller (SMC), since this control strategy does not require any prior knowledge of the dynamic model of the system [85]. The SMC controls the states of the vehicle and makes them converge to a sliding manifold s with global exponential stability properties. A sliding manifold is determined based on the control application, and for UVMSs, a common sliding manifold can be defined as

$$s = \begin{bmatrix} \dot{\mathbf{q}}_r - \dot{\mathbf{q}} \\ \mathbf{v}_r - \mathbf{v} \end{bmatrix} + \mathbf{\Lambda} \int_0^t \begin{bmatrix} \dot{\mathbf{q}}_r - \dot{\mathbf{q}} \\ \mathbf{v}_r - \mathbf{v} \end{bmatrix} d\tau = 0, \quad (2.25)$$

where $\mathbf{\Lambda}$ is an integral gain matrix and \mathbf{v}_r and $\dot{\mathbf{q}}_r$ are reference velocities. This sliding manifold is defined to be globally exponentially stable if $\mathbf{\Lambda} > 0$ [77]. The sliding manifold is used in a control law given by

$$\boldsymbol{\tau} = \mathbf{K}_D \mathbf{s} + \hat{\mathbf{g}}(\mathbf{q}, \boldsymbol{\Theta}) + \mathbf{K}_S \text{sat}(\mathbf{s}, \epsilon) \quad (2.26)$$

to keep the vehicle states at the sliding manifold. Here, \mathbf{K}_D and \mathbf{K}_S are positive definite gain matrices, $\mathbf{g}(\mathbf{q}, \boldsymbol{\Theta})$ represents restoring forces and moments, and $\text{sat}(\mathbf{s}, \epsilon)$ is a saturation function of \mathbf{s} with upper and lower limit $\pm\epsilon$. The saturation function is chosen instead of a signum function to avoid chattering [77].

2.3.4 System Integration of Underwater Vehicle Manipulator Systems

The system integration of a UVMS involves the integration of all inherent parts in the system and determines the assembly of the UVMS, the involved hardware solutions, and the involved software solutions. A UVMS can typically be divided into an operator computer and one or more robots, where the manipulator can be recognized as part of the underwater vehicle or as a separate robot. Typically, underwater manipulators require independent actuation systems and sets of sensors and consist of multiple DOFs. From a system integration point of view, it is thus simplest to view the manipulator as an independent robot. The system integration of a UVMS includes three machines: the operator computer, the ROV, and the manipulator. The operator computer typically functions as the operator's control panel with screens including the graphical user interface (GUI) of the system, direct communication to the other machines through tethered connections, and a connected control panel. This allows all machines to communicate with each other.

2. State of the Art

If a standard Ethernet connection is used, this supports 100 Mbps, meaning it can transmit 100 Megabits through the connection each second, which is typically more than sufficient for these systems.

Typically, the operator computer runs as the main machine and manages all heavy computational tasks, such as navigation and guidance strategies, advanced control calculations, camera detection and tracking systems, and other sensor analysis. The two robots normally handle distribution of forces and torques to thrusters and joints, stream the camera feed, measure sonar data, transmit sensor data to the operator computer, and keeps track of the vehicle status including positions and velocities.

Chapter 3

Facilities and Equipment

This chapter outlines the facilities and equipment used for experimental testing in this thesis. The articles that are relevant for the various equipment are stated.

3.1 Facilities: the Marine Cybernetics Laboratory

The Marine Cybernetics Laboratory (MC-lab) is a laboratory located at the Department of Marine Technology, NTNU, in Trondheim [47]. The MC-lab was the main experimental platform for the contributed articles and was used for either data collection or experimental testing for all of them. The laboratory consists of a control room and a towing tank with dimensions $40\text{ m} \times 6.45\text{ m} \times 1.5\text{ m}$, which includes a wave maker, a towing carriage, and a 6-DOF real-time positioning system. The MC-lab is often used for experimental testing of dynamic positioning (DP) systems and is mainly used by Master's students and PhD candidates at the Department of Marine Technology, but is also available for external users. The towing tank and control room are depicted in Figure 3.1. The MC-lab was used for data collection in Articles 1, 2, 3, 4, 5, 6 and 10, and experimental testing was conducted in the laboratory pool in Articles 2, 3, 4, 5, 6 and 10. In addition, for Articles 7 and 10, the MC-lab pool was used as an environmental template for a simulated environmental model in the Gazebo simulator, which is an open-source simulator that offers 3D robot simulation and visualization [31]. The Gazebo simulator focuses on the ability to simulate dynamics and incorporate the accuracy of robot sensors and actuators.

The positioning system in the MC-lab uses Oqus cameras to measure the position of markers both on the surface and underwater. The key components of the system are the Oqus cameras and the Qualisys Track Manager software, which runs on a

3. Facilities and Equipment



Figure 3.1: The MC-lab—towing tank and control room [47].

computer in the laboratory computer room. The Oqus cameras provides precision and real time marker generation inside the cameras and are capable of recording high-speed, high-resolution video citeoqus. Three surface Oqus cameras record motions for surface vessels and six underwater Oqus cameras record motions for underwater vehicles, objects, and structures. For the work conducted in this PhD project, only the underwater motion capture system was used. Using triangulation between the cameras, the Qualisys Track Manager system provides estimates of position and orientation of an object by detecting a set of markers arranged on the object. The accuracy of the system is generally within one centimeter for position and one degree for orientation.

3.2 Equipment

3.2.1 BlueROV2 Underwater Vehicle

The BlueROV2 is a small observation-class ROV produced by BlueRobotics [8] and is a high-performance, low-cost underwater ROV used for both professional and leisure activities. The ROV comes with open-source electronics and software and is highly customizable for use in inspections, research, and recreation. The ROV has six thrusters in standard configuration providing 4 DOFs, and eight thrusters when using the Heavy Configuration Retrofit Kit, which provides full six-DOF control. The BlueROV2 utilized in this research uses the Heavy Configuration Retrofit Kit and is depicted in Figure 3.2. The ROV comes with open-source software and can be controlled using the open-source ArduSub subsea vehicle control firmware. For

this work, new in-house software solutions were developed to replace the open-source software. This was to ensure that the integration of all the components in the ROV were compatible with other equipment used in this research. The software is based on the Robotic Operating System (ROS), enabling safe and fast communication between the involved robots and operators [27]. The embedded software is described more in depth in Section 3.2.3, including both the BlueROV2 and the SeaArm-2. The main specifications of the BlueROV2 are listed in Table 3.1. Note that, in the table, the weight in water is approximately zero, the reason being that this is an adjustable parameter. The vehicle is approximately neutrally buoyant in water and weights can be attached and removed from the vehicle to maintain neutral buoyancy. Moreover, this depends on number of and type of payload that is attached to the ROV, such as a Doppler velocity log, manipulator, or other external sensors. The BlueROV2 underwater vehicle was used for experimental testing in Articles 2, 3, 5 and 10, and as a model template for a digital twin in Articles 7 and 10.



Figure 3.2: The BlueROV2 underwater vehicle.

Table 3.1: BlueROV2 specifications.

Parameter	Value
Weight in air	11.5 kg
Weight in water	~0 kg
L × H × W	457 × 254 × 575 mm
Thrusters	T-200
Battery	14.8 V, 18 Ah
Depth rating	100 m
Camera	Raspberry Pi Camera Module V1
On-board Computer	Raspberry Pi 3B and Navio2

3. Facilities and Equipment

3.2.2 SeaArm-2 Manipulator

The SeaArm-2 manipulator is a modular, fully electric, low-cost manipulator. The modular design of the manipulator provides an adjustable number of DOFs. The manipulator is depicted in Figure 3.3, where it is set up with one base module, three standard modules, and one end module giving it 4 DOFs. The SeaArm-2 manipulator is a successor to the SeaArm manipulator and builds on the same principles for electrical driven and modular design [23]. The module housings of the new design were rounded off to better withstand pressure, slip rings were installed to enable continuous joint rotations, gears were developed to triple the lifting capacity, a camera was integrated in the end module, and the base module was expanded to include an onboard computer. The onboard computer enables distribution of forces and torques, logging of manipulator status, video streaming, and other processes of low computational load that need to be performed on board. Besides the onboard computer, the most important improvement of the manipulator is the integrated camera in the end module that enables recording and live streaming from the manipulator's point of view. The camera is a low-light HD USB camera (based on Sony IMX322) tailored to an environment of limited visible light. The camera is slightly tilted toward the gripper to get a more substantial view of gripper operations. The main specifications of the manipulator are listed in Table 3.2. Note that these specifications are for the manipulator setup depicted in Figure 3.3.

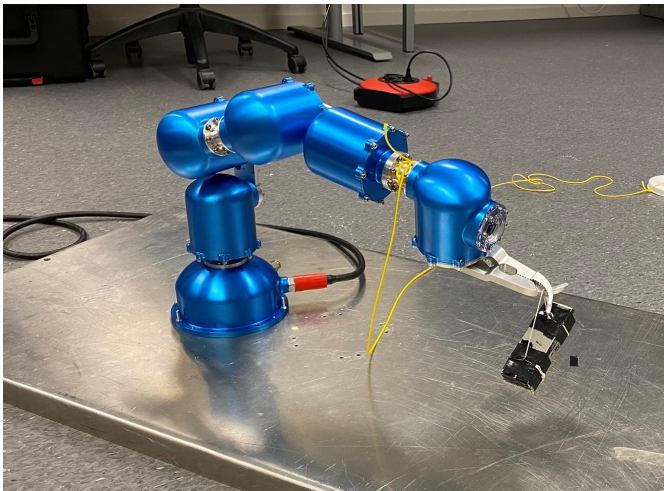


Figure 3.3: The SeaArm-2 manipulator.

The manipulator has a tether that provides communication and power to the onboard computer, servos, and the camera. The manipulator can operate independently, with the tether providing communication to an operator computer and power from

Table 3.2: SeaArm-2 main specifications.

Parameter	Value
Degrees of freedom	4
Weight in air	3.58 kg
Weight in water	0.35 kg
Max reach	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
On-board computer	Raspberry Pi 3B
Communication	RS485 and Ethernet
Camera	Low-light HD USB camera

a topside power supply. The manipulator can also operate as part of a UVMS, in which case communication and power are transferred between the ROV and the manipulator. Communication through the tether is provided with Ethernet connections to a local IP network between the operator computer, the manipulator, and if the system in consideration concerns an UVMS, the ROV. The SeaArm-2 manipulator is used in experimental testing in Articles 4, 6 and 10, and as a model template for digital twin in Articles 7 and 10.

3.2.3 UVMS combining the BlueROV2 and the SeaARM-2

The SeaArm manipulator was originally designed to be equipped to small working class ROVs [23]. The nearly neutral buoyancy and small size of the manipulator makes it convenient to attach to a ROV such as the BlueROV2. Embedded software solutions based on the same principles have been developed for both the SeaArm-2 manipulator and the BlueROV2 underwater vehicle. This way, the two robots can easily be combined into a joint system. When the two robots are controlled individually, they communicate with an operator computer through Ethernet connection. Furthermore, when they operate in a combined system, the two robots and an operator computer are all connected with Ethernet connection through a network switch. The connection setup is illustrated in Figure 3.4. In a combined UVMS, the SeaArm-2 is attached underneath the BlueROV2 as demonstrated in Figure 3.5 and connected through a tether with wet mate connection between the robots. The connection delivers power transferred from the onboard battery in the BlueROV2 and Ethernet communication. The Ethernet communication from the SeaArm-2 manipulator is rerouted in the BlueROV2 through a tethered connection between

3. Facilities and Equipment

the BlueROV2 and an operator computer at topside. This way, the tether from the BlueROV2 delivers Ethernet communication from both robots which eliminates the need for two separate tethers.

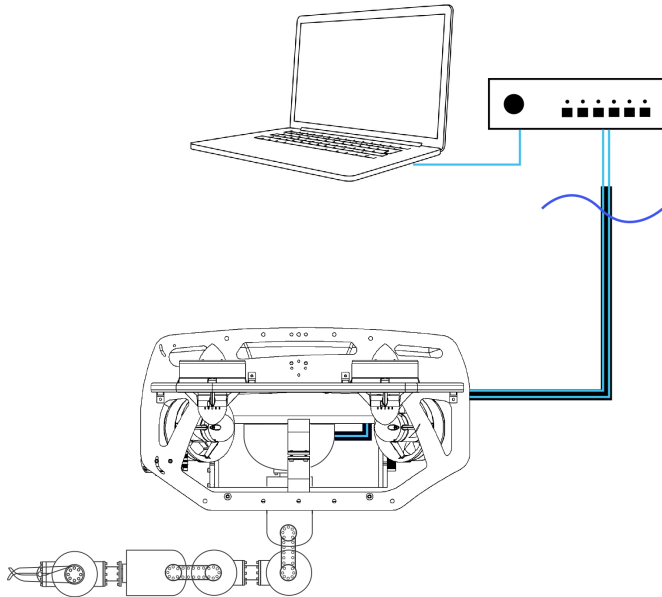


Figure 3.4: Tethered communication of the UVMS.



Figure 3.5: The SeaArm-2 connected underneath the BlueROV2.

Chapter 4

Summary of Results

This chapter summarizes the aims, methodologies and results of the articles enclosed in this thesis.

4.1 Article 1: Transfer Learning in Underwater Operations

Purpose and Novelty

The goal of this article is to research a method of using transfer learning to reduce the reality gap that occurs when applying simulated data in training to vision-based operations in a subsea environment. Furthermore, the large datasets required for machine learning can be expensive and difficult to acquire, and transfer learning could represent a solution. The novel contributions of this article include the collection of two image datasets including rendered and real images of subsea template templates. Moreover, a GAN is trained on the datasets to enable mapping from a simulated environment to the real world, generating photorealistic images from simulated images.

Methodology

Two separate image datasets were collected. The two datasets include unpaired images from two domains: a simulated environment and real images. The first dataset involves real images of a subsea panel placed in the MC-lab pool, and computer-generated images of a simulated model of the same subsea panel. The simulated images were rendered using a computer software called Blender [1]. The second dataset involves another subsea panel located in Trondheimsfjorden and corresponding rendered images taken from a computer-aided design (CAD)

4. Summary of Results

model of the same subsea panel. The dataset from the MC-lab includes 3,065 real images and 5,000 rendered images, and the dataset from Trondheimsfjorden includes 452 real images and 1,986 rendered images.

A neural network based on the CycleGAN framework [95] was adopted to generate a mapping between the two domains. CycleGAN uses the adversarial loss and an additional cycle consistency loss to learn the mapping $x \approx F(G(x))$. This represents the mapping from domain x to domain y , and then the mapping back again to domain x . F and G represent the generator networks that generate images in one domain based on input images from the other domain. The framework uses discriminator networks D_x and D_y to distinguish between images from the actual domain and images created by the generator networks. The discriminator networks penalize bad image generations and ensures the generator networks learn an optimal mapping.

Results

The neural network based on the CycleGAN framework was trained on the two image datasets containing real and rendered images. The image datasets were unpaired, meaning the images from the two domains had no image-to-image correlation. An image-to-image pairing can be a time-consuming processes, since it requires every image in domain x to have a corresponding image in domain y . The CycleGAN learned the general mapping between the domains using unpaired image datasets, and used the cycle consistency loss to compare the input with the reconstructed image. Figure 4.1 presents the mapping $x \approx F(G(x))$ and $y \approx G(F(y))$ for the two datasets. It was shown that the output was a good mapping between the domains and that the reconstruction resembled the input image.

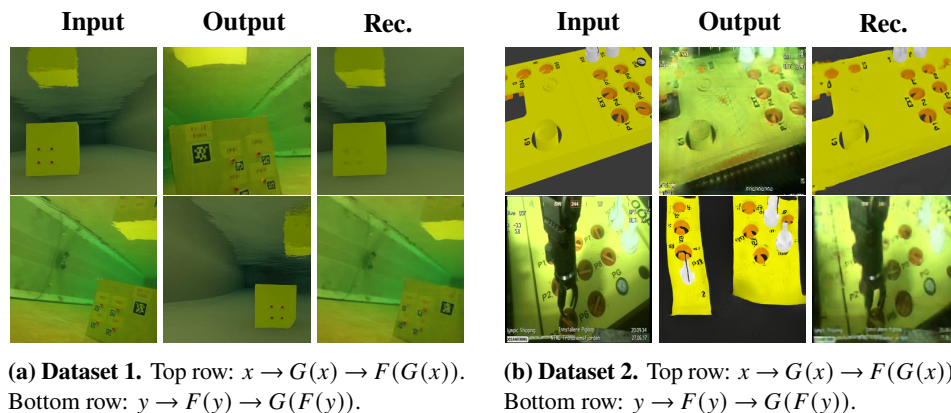


Figure 4.1: Input and output with reconstructed image.

4.1. Article 1: Transfer Learning in Underwater Operations

A selection of the results is depicted in Figures 4.2 and 4.3. Figure 4.2 shows the results for $x \rightarrow G(x)$ for the two datasets, representing the domain transfer from the simulated domain to the real-world domain. Figure 4.3 shows the results for $y \rightarrow f(y)$ for the two datasets, representing the domain transfer from the real-world domain to the simulated domain. Mapping back and forth between both domains was conducted to see if the framework correctly mapped the relevant features between the domains. The framework was able to map features characterized by underwater environments, such as dark images and light reflection. However, some features were not sufficiently mapped. For dataset 1, the relative angle between the camera and structure as well as the spatial features deviated in the output relative to the input. The details on the subsea panel also seemed to be randomly placed on the panel. This suggests that the mapping of some features was insufficient. However, it should be noted that the domains in fact possessed some different features, with, for example, QR codes being neglected in the rendered domain. The results suggested that the framework encountered problems when mapping features that were simply non-existent in one of the two domains.

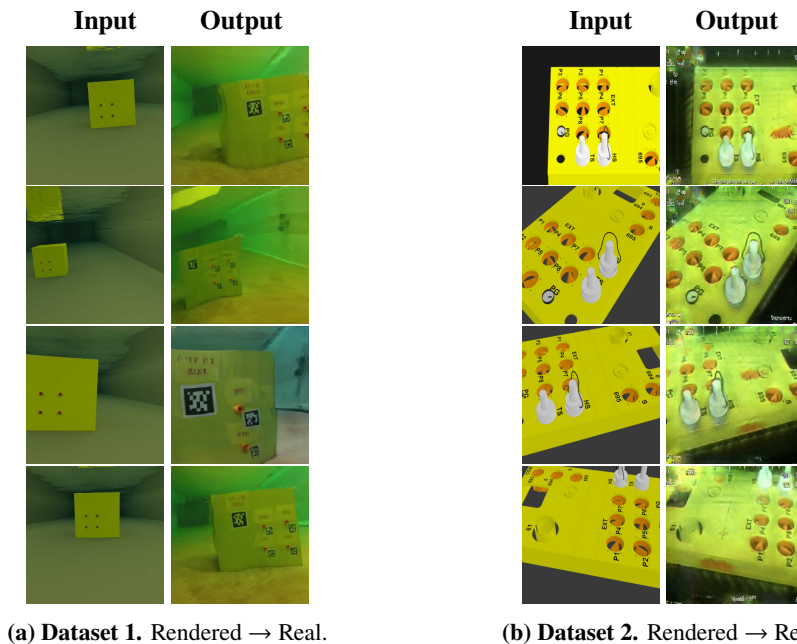


Figure 4.2: Results from generating images in the real domain with rendered images as inputs.

The large datasets required for machine learning can be expensive and difficult to acquire. Applying transfer learning methods to underwater environments can

4. Summary of Results

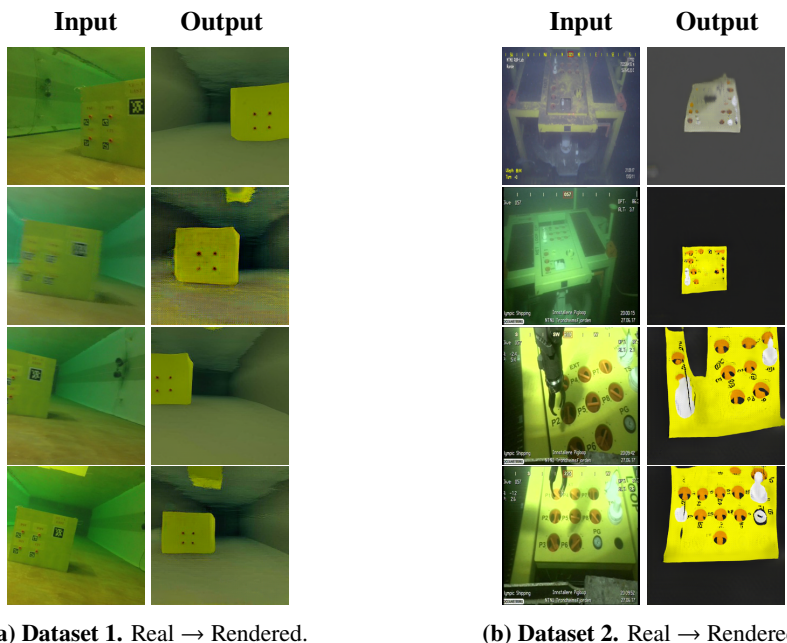


Figure 4.3: Results from generating images in the rendered domain with real images as inputs.

provide an alternative method for cost-effective and simple dataset generation. Regarding dataset generation for future machine learning applications, the results presented in Figure 4.2 would be the most interesting. Proper generation of images in the real-world domain enables the generation of vast datasets from rendered images.

4.2 Article 2: Dynamic Positioning of an Underwater Vehicle using Monocular Vision-Based Object Detection with Machine Learning

Purpose and Novelty

The purpose of this article is to use monocular vision in combination with a scaling function to design a 3D computer vision system. Moreover, the computer vision system is used as the feature extractor on a DP system for an underwater vehicle. The system is verified in experimental testing. The novel contributions of this article include an effective method for collecting and labeling a large image dataset, a scaling function for extracting spatial features for objects detected through object detection, and the design of a DP system on a small underwater vehicle to maintain a relative distance to the detected object.

Methodology

An image dataset of 7,071 images was generated, consisting of images of a relevant object that was later used in DP operation. The images were collected by splitting up videos recorded by the monocular camera in a BlueROV2. The images were then labeled initially by color detection, followed by manual correction of bad or incorrect labels using an interactive GUI. An object detector based on the YOLOv3 algorithm was trained on the dataset and a scaling function was developed to extract the spatial features of the detected 2D images. The scaling function related the relative distance to the object with the size of the detected bounding box around the object in the image frame. The 3D position of the object was then calculated using the camera's intrinsic parameters and simple geometrical relations. The vehicle conducted DP relative to the object, keeping the object in the center of the camera frame by controlling its velocity with an SMC.

Results

The target object was characterized by a clear orange color, which clearly distinguished it from its surroundings. The proposed model yielded an AP score of 97.7% with a classification threshold of 50%, and the object detection system demonstrated excellent detection abilities in the experimental testing. An image depicting the vehicle performing DP relative to the object is shown in Figure 4.4(a), and the vehicle's camera view with successful detection is shown in Figure 4.4(b).

The SMC achieved high velocity tracking performance; the position and velocity errors are plotted in Figure 4.5. The root mean square error (RMSE) for position and velocity are listed in Table 4.1. The position error was below 2.5 cm in all

4. Summary of Results



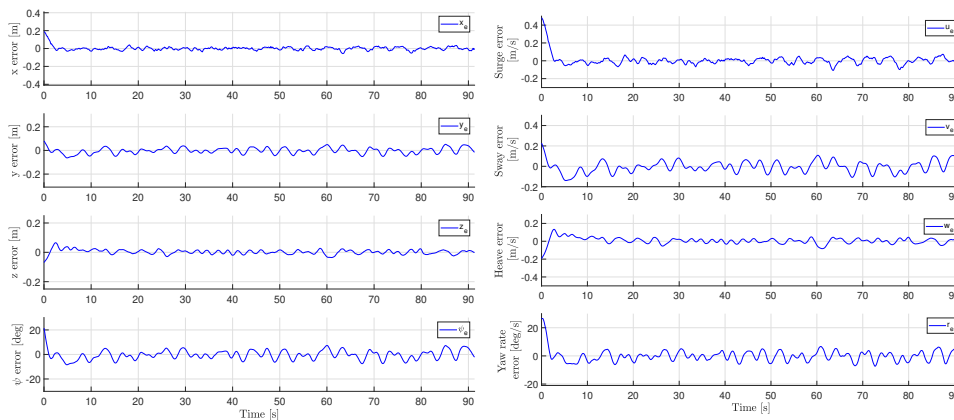
(a) The BlueROV2 performing DP relative to the relevant object in the MC-lab pool.



(b) The BlueROV2 camera view with detection of the relevant object.

Figure 4.4: DP operation with object detection in the MC-lab pool.

directions and the heading angle was below 4 degrees. The system experienced some difficulties maintaining the desired heading, which is likely the result of a combination of slow vehicle dynamics and the delay from image acquisition to the completion of image processing. There was also a small discrepancy of 5–10 degrees between the camera angle and the vehicle’s thrust allocation system. This implies some uncertainty in the estimated position of the object relative to the vehicle.



(a) Relative position error over time.

(b) Relative velocity error over time.

Figure 4.5: Relative position error and velocity error between the vehicle and object over time.

4.2. Article 2: Dynamic Positioning of an Underwater Vehicle using Monocular Vision-Based Object Detection with Machine Learning

Table 4.1: RMSE for position and velocity of the vehicle relative to the object.

RMSE for position	Value	RMSE for velocity	Value
$RMSE_x$	0.024 m	$RMSE_u$	0.060 m/s
$RMSE_y$	0.025 m	$RMSE_v$	0.056 m/s
$RMSE_z$	0.016 m	$RMSE_w$	0.037 m/s
$RMSE_\psi$	3.7 deg	$RMSE_r$	4.2 deg/s

4.3 Article 3: Monocular Vision-Based Gripping of Objects

Purpose and Novelty

The focus of this article is the research of autonomous grasping with a UVMS by using monocular vision and object detection to track an object. Moreover, the object detection is used as guidance for the positioning of the end-effector before grasping. The novel contributions of this article are threefold. First, a computer vision system capable of 3D detection is used to maintain the desired position of the vehicle relative to the object of interest. Second, a positioning procedure for the end-effector based on the same computer vision system is developed. The procedure maneuvers the end-effector to the object location through a developed kinematic control system and allows for object retrieval. Third, two experimental case studies are developed and performed to verify the proposed methodologies.

Methodology

The computer vision system involved a neural network object detector based on the YOLOv3 algorithm and a scaling function extracting spatial information from the image frame. The YOLOv3 model was trained on a labeled image dataset containing the relevant object, and the scaling function related the object's detected bounding box size with the relative distance using an interpolation function in combination with the camera's intrinsic parameters and simple geometric relations. Due to computational requirements, the computer vision system was run on the operator computer at topside using video feed streaming from the vehicle. A Kalman filter estimated the relative velocity of the vehicle based on the estimated position data, and an SMC calculated the desired velocities for the vehicle to keep a desired position relative to the object. Furthermore, the 3-DOF manipulator used kinematic control and inverse kinematics to reach this desired position with its end-effector. The kinematic control applied the pseudo-inverse of the Jacobian with damping least squares to avoid singularities. The grasping of the object was performed by manually commanding the gripper to close when the object was stationed within grasping reach. Experimental testing in the MC-lab pool verified the proposed methodology.

Results

The vehicle tracked the object successfully, demonstrating that the proposed DP system based on a computer vision system works as intended. Even though the DP system was confirmed to work in Article 2, this system utilized a different BlueROV2 in addition to an attached manipulator that substantially changed the

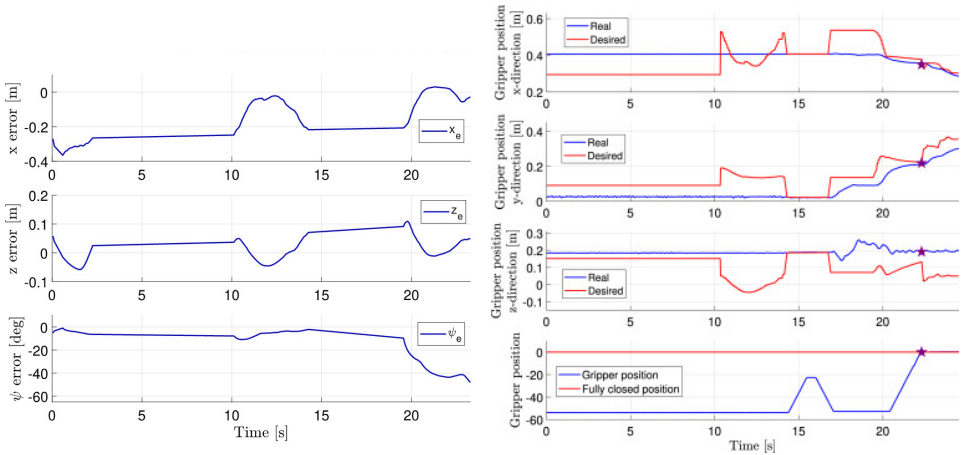
4.3. Article 3: Monocular Vision-Based Gripping of Objects

vehicle dynamics. Furthermore, the same computer vision system was used to guide the manipulator end-effector to the object position for grasping. Figure 4.6 shows a successful grasp of the object, both from a spectator view and the vehicle view. Position tracking performance during one of the experimental tests, both for the vehicle and the manipulator end-effector, is shown in Figure 4.7. The object was grasped after approximately 22 seconds.



(a) The UVMS successfully grasping the object, seen from outside. (b) The UVMS successfully grasping the object, seen from the vehicle's camera.

Figure 4.6: The UVMS successfully grasping the object.



(a) Vehicle error position in the x- and z-directions and heading angle ψ .

(b) Measured (blue) and desired (red) end-effector position in the x-, y-, and z-directions. The star symbolizes a successful grasp of the object.

Figure 4.7: Vehicle and manipulator end-effector positioning performance during a successful object grasping operation.

The article addresses some issues related to the proposed methods, especially regarding having a manipulator with only 3 DOFs and occlusions of the object by the manipulator. Partial occlusion of the object results in an incorrect position estimate,

4. Summary of Results

since this is based on detecting the object through the computer vision system. Potential methods for dealing with occlusions are mentioned in the article. Some suggested methods are path-following or tracking procedures of the manipulator joints, adding occlusion avoidance within a task priority framework, or simply moving the manipulator away from camera. Moreover, a manipulator with more DOFs is preferable, which would allow for a larger work space, lower probability of being close to singularities, and potentially reducing the time taken to grasp the object.

4.4 Article 4: SeaArm-2 - Fully Electric Underwater Manipulator with Integrated End-Effector Camera

Purpose and Novelty

The purpose of this article is to present a new electrically driven underwater manipulator named SeaArm-2, and perform preliminary experimental testing and verification of its capabilities. The novel contribution of this article is the improved design and strength of the manipulator compared with the previous SeaArm manipulator, in addition to the introduction of an onboard computer and an integrated end-effector camera. Proof of concept is provided through experimental testing of the proposed manipulator, where autonomous fixed-base grasping is performed using the integrated camera as the main sensor for guidance.

Methodology

The improved design of the manipulator enables new possibilities for manipulation compared to the previous version. The manipulator kinematics were established through the DH convention, with final reference frames representing both the image frame in the end-effector camera and the gripper position. The integrated monocular camera allowed a computer vision system to be applied to the manipulator system for guidance and navigation. Experimental validation was performed through a fixed-base autonomous grasping procedure, in which the manipulator searched for, tracked, grasped, and retrieved an object of interest. To detect the object, an object detector based on the YOLOv5 neural network was applied and trained on a labeled image dataset of 6,345 images. The relative 3D position between the object and manipulator was estimated using a spatial feature extractor that used a mathematical approach based on the camera's intrinsic parameters and geometric relations with the object size.

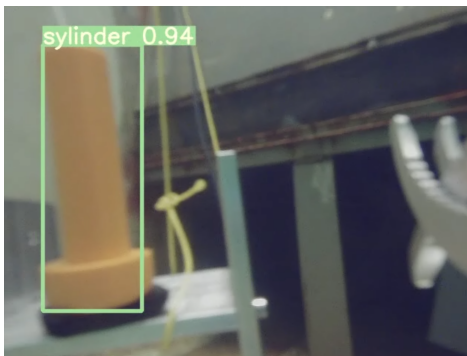
Results

The presented SeaArm-2 manipulator was developed as a small, underwater, low-cost, fully electric, modular manipulator. The modular design of the manipulator allows for varying the number of DOFs based on the modular assembly. In the standard assembly, the manipulator has 4 DOFs with maximum reach of 693.75 mm and an increased lifting capacity to 5 kg at full reach, thanks to new integrated slip rings and gears. The computer vision system provided excellent detection capabilities and the training resulted in an AP score of 99.3% with IoU at 0.5 and an average AP score of 85.4% for IoU from 0.5 to 0.95. The precision score and recall rate are also satisfactory with P=0.962 and R=0.989. High precision indicates a low number of false positives, whereas high recall represents a low

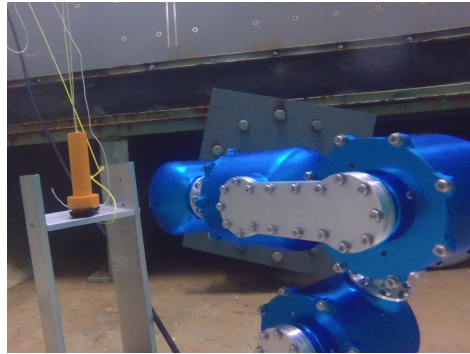
4. Summary of Results

number of false negatives. The obtained precision and recall values indicate low numbers of both false positives and false negatives. The computer vision system is capable of 3D position estimates through a spatial feature extractor that uses a mathematical approach to calculate the relative 3D position between object and manipulator. The spatial feature extractor is based on the known size of the object, the camera's intrinsic parameters, and geometrical relations.

The autonomous fixed-base grasping procedure used the computer vision system for guidance and navigation. Moreover, the manipulator control system is based on kinematic control with inverse kinematics using the manipulator pseudo-inverse with damping least squares to avoid singularities. Figure 4.8(a) shows the manipulator detecting an object and Figure 4.8(b) shows the system from the outside. The autonomous fixed-base grasping procedure was tested in 20 trials, of which 17 were deemed successful. The RMSE values for all 20 experiments are listed in Table 4.2.



(a) Detection of an object, seen from the integrated end-effector camera.



(b) The manipulator and object in tracking mode during an autonomous fixed-base grasping experiment.

Figure 4.8: The manipulator system during experimental testing.

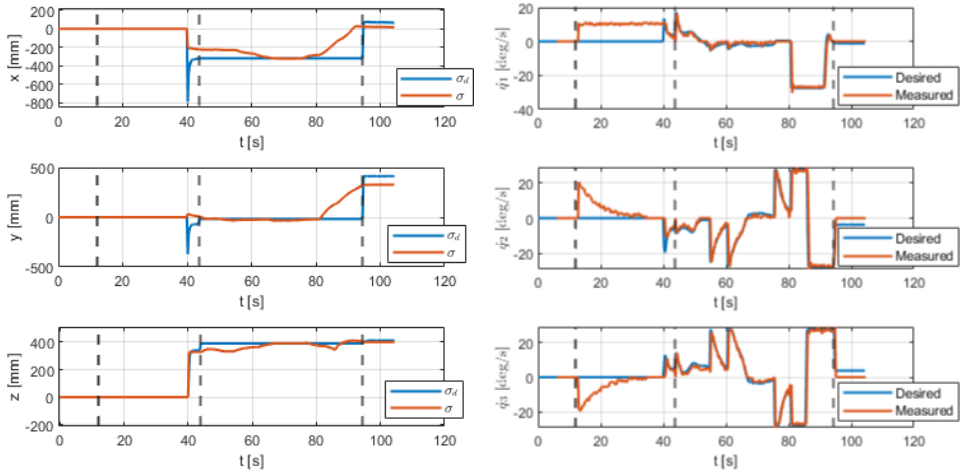
Table 4.2: RMSE for joint velocities (in deg/s) over all 20 trials.

RMSE	\dot{q}_1	\dot{q}_2	\dot{q}_3	Total
Value	2.63	4.15	3.86	3.55

The three unsuccessful runs were caused by either faulty distance estimation or faulty calibration of the manipulator joints. The calibration of the manipulator joints was necessary for the manipulation system to have correct self-awareness of the joint positions and velocities. If this calibration is faulty, the joint position readings will be inaccurate, leading to a misleading end-effector position. The

4.4. Article 4: SeaArm-2 - Fully Electric Underwater Manipulator with Integrated End-Effector Camera

desired and measured position of the end-effector during one of the successful experiments are plotted in Figure 4.9(a). The desired and measured joint velocities for the same experiment are plotted in Figure 4.9(b). The three vertical dotted lines in the plots at approximately 15 s, 45 s, and 95 s represent the times when searching for the object was initialized, when grasping was initialized, and when the retrieval of the object was completed. Note that both desired positions and joint velocities were activated when the object was first detected at about 40 seconds.



(a) End-effector desired (blue) position and measured position (red) for an experiment.

(b) Desired (blue) and measured (red) joint velocities for an experiment.

Figure 4.9: Desired and measured values for one of the 20 experimental trials.

4.5 Article 5: Unsupervised Domain Transfer for Task Automation in Unmanned Underwater Vehicle Intervention Operations

Purpose and Novelty

The research presented in this article consists of the development of a method for producing segmented data that can be used to train neural networks, in situations where acquiring real data is either difficult or the labeling of the data is prohibitively time-consuming. The novel contributions of this article are threefold: first, the acquisition of a novel dataset; second, domain adaption based on the novel dataset with comparison of optimal training cycles; and third, the training of an object detector with experimental validation on a underwater vehicle in a laboratory pool.

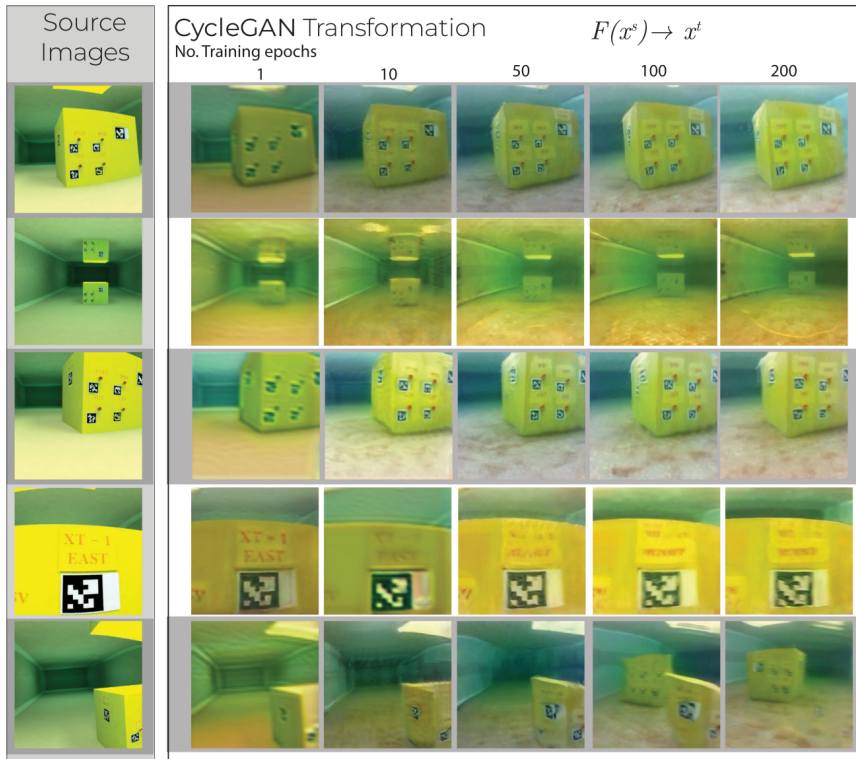
Methodology

The novel dataset was generated to include two sets of images. The first set contains about 5,000 images of a subsea panel in the MC-lab pool. The second set of images includes about 20,000 synthetic images involving the same subsea panel, generated from a synthetic environment created using the Blender software [1]. The domain adaption generated a mapping between the two domains using the CycleGAN framework, and used this image-to-image translation to generate a completely new and unseen dataset. For the experimental testing, an object detector was trained on the new dataset produced by the image-to-image translation. The object detector assisted in the guidance and navigation system of an underwater vehicle that performed simple DP relative to the underwater panel in experimental testing in the MC-lab pool.

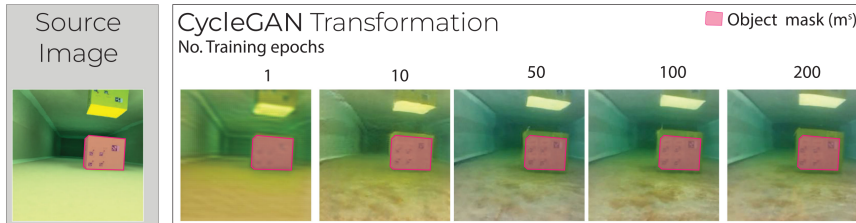
Results

The CycleGAN framework was trained on the novel dataset including images of two domains. Figure 4.10 illustrates image-to-image translation using the CycleGAN network after training for different lengths of time. As can be seen from this figure, the images became more and more realistic as training progressed; however, there were also increasing flaws, which are especially apparent in the final image. After training for 200 epochs, the network started focusing more on generating a random realistic image than on preserving the subsea panel characteristics. The synthetic images have the advantage that they include the segmentation mask adopted from the Blender software. Figure 4.10(b) shows how the segmentation mask matched the image translation in the new domain over different training epochs. With increased training epochs, a displacement of the subsea panel position occurred resulting in an inaccurate segmentation mask.

4.5. Article 5: Unsupervised Domain Transfer for Task Automation in Unmanned Underwater Vehicle Intervention Operations



(a) Image-to-image translation using the CycleGAN framework. Left: Original simulated images. Right: Transformation of the original images after different training lengths (epochs) of the network.



(b) The segmentation masks on input and translated images. Left: Original simulated image with segmentation mask. Right: Transformation of the original image after different training lengths (epochs) of the network with the corresponding segmentation mask.

Figure 4.10: Results of the image-to-image translation.

The displacement of the object in the image-to-image translation created problems in the new generated dataset. This dataset was intended as a labeled image dataset used for an object detection network with the segmentation masks as labels. If the labels are inaccurate, the object detector will be inaccurate. Figure 4.11 shows plots of the AP and mAP scores of an object detector based on the YOLOv3

4. Summary of Results

algorithm. The detector is trained on a generated dataset from the CycleGAN network after different numbers of epochs. The plots show that the object detector achieved its highest mAP score with IoU between 0.5 and 0.95 when training on a dataset generated by the CycleGAN network that trained for 3 epochs. The highest AP score was achieved when training on a dataset generated by the CycleGAN network that trained for 32 epochs. However, the AP score was relatively stable for large parts of the range of CycleGAN training epochs. The deterioration of the mAP scores demonstrates that even though image datasets generated at later epochs achieved a better domain adaption and were more realistic, they achieved lower detection accuracy due to the inaccurate labeling of the dataset.

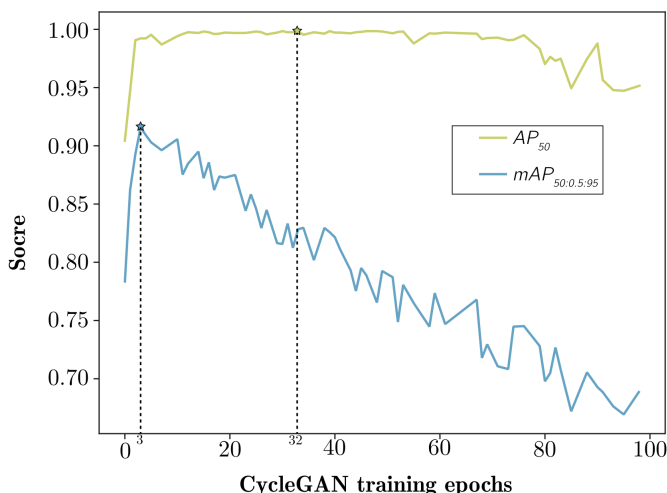


Figure 4.11: Object detector AP score with IoU at 0.5, and mAP score for IoU from 0.5 to 0.95, over training epochs for the CycleGAN network.

The experimental validation of the system is illustrated in Figure 4.12, with a multiframe image in which the underwater vehicle rotates around the subsea panel. The experiments demonstrated that the object detector could reliably detect the subsea panel, and that the system could apply the detection to maintain the panel in the center position of the camera frame.

4.5. Article 5: Unsupervised Domain Transfer for Task Automation in Unmanned Underwater Vehicle Intervention Operations

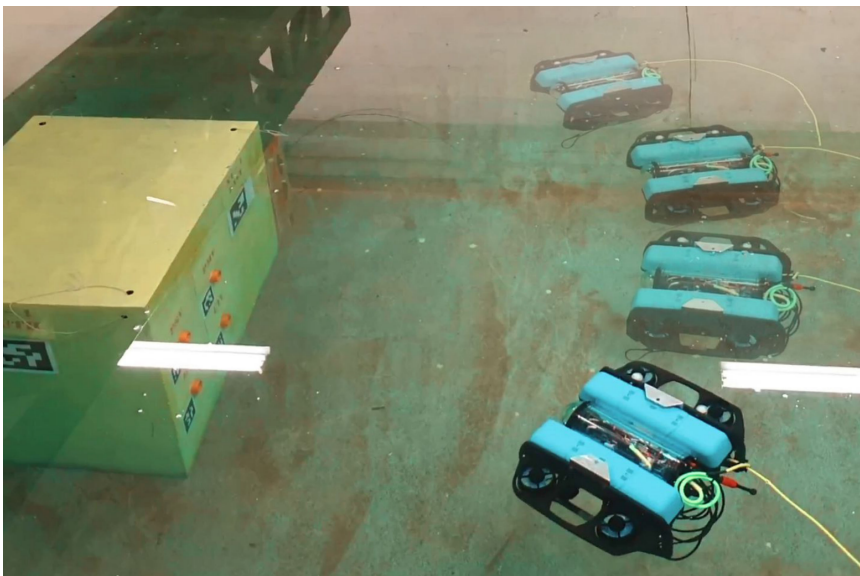


Figure 4.12: Multiframe image of the vehicle, showing how the controller kept the panel at the center of its camera frame, while the vehicle moved sideways.

4.6 Article 6: Autonomous Underwater Grasping using a Novel Vision-Based Distance Estimator

Purpose and Novelty

The goal of this article is the research and introduction of a novel distance estimator using monocular vision and the verification of the method by experimental testing. The novel contribution of the article is the presentation of a new method to estimate distances and the 3D position of an object of unknown size using monocular vision. Furthermore, experimental verification is presented in which the model is tested on an underwater manipulator for autonomous fixed-base grasping.

Methodology

An estimator was developed for robot manipulators with a monocular camera placed near the gripper. The fact that the camera was attached near the gripper made it possible to design a method for capturing images from different positions, as the relative change in position could be measured. The manipulator system used a computer vision system with an object detector based on the YOLOv5 network to locate objects in the camera frame, where the detected bounding box was used in the distance estimator procedure. The detector was trained on a novel dataset including 6,533 images and corresponding labels, with multiple objects that were classified within two different classes. The computer vision system was also enhanced using an object tracker based on the DeepSORT tracker [11] to account for temporal information in the manipulator video stream.

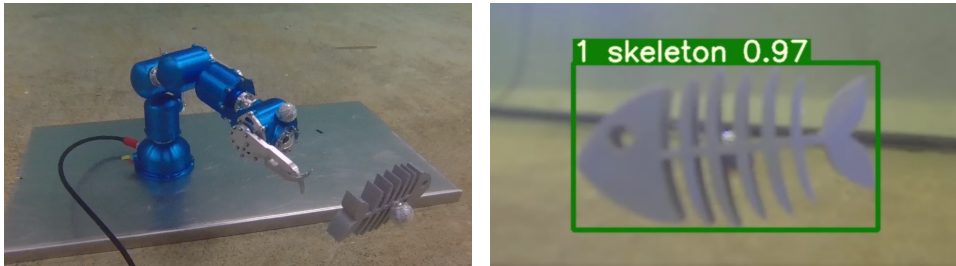
The distance estimator estimated the size of the detected objects based on the camera's intrinsic parameters and geometric relations between multiple captured images from different manipulator reach extensions. The computer vision system including the distance estimator was validated in two case studies with experimental testing on the SeaArm-2 manipulator in the MC-lab pool. The first case study included estimation of the distance to objects in the laboratory pool, and the second case study involved autonomous fixed-base grasping, in which the computer vision system was included in the guidance and navigation of the manipulator system.

Results

The computer vision system detected and estimated the relative distance to an object of unknown size with good precision. The detector achieved an mAP score of 99.4% with IoU at 0.5 and an average mAP score of 94.5% with IoU ranging between 0.5 and 0.95. The precision score and recall rate of the trained detector were 0.976 and 0.997, respectively, which indicates low numbers of both false positive and false negative detections. The DeepSORT tracker further enhanced the

4.6. Article 6: Autonomous Underwater Grasping using a Novel Vision-Based Distance Estimator

computer vision system with improved stability in detection, tracking capabilities, enabling object identification, and allowing for tracking through object occlusions. The manipulator and an object of interest are depicted in Figure 4.13(a), and the detection from the manipulator end-effector camera is depicted in Figure 4.13(b).



(a) The SeaArm-2 Manipulator and a 3D-printed fish object placed in the MC-lab pool, with silver Qualisys motion markers attached.

(b) End-effector camera view. The labels above the detected bounding box represent tracker ID, object class, and detection confidence score.

Figure 4.13: The manipulator system in the MC-lab pool detecting an object, seen from outside and from the manipulator end-effector camera.

The first case involving estimating object distances was intended to offer an understanding of the performance, reliability, and accuracy of the distance estimator to highlight any unfavorable behavior of the system. A set of 20 trials was conducted with two different objects at different positions in the laboratory pool and the resulting RMSE values are listed in Table 4.3. The two object classes were 3D printed fish and 3D printed fish skeletons. This large difference in RMSE values for the two objects resulted from an inaccuracy in the computer vision framework, where a visible Qualisys motion tracker marker on the fish object was falsely detected as a part of the object, which interfered with the distance estimation.

Table 4.3: Distance estimator RMSE for case study 1.

RMSE fish	RMSE skeleton	RMSE Total
124.24 mm	26.81 mm	75.53 mm

The second case study involved autonomous fixed-base grasping of the objects. The computer vision system estimated the relative 3D positions of the object and the manipulator. Furthermore, kinematic control using inverse kinematics enabled the manipulator to grasp the object using the estimated 3D position as the desired end-effector position when grasping. All 12 experiments that were conducted resulted in a successful grasp and retrieval of the object, and the grasping procedure of one of the experiments can be seen in Figure 4.14. The RMSE values for all experiments are listed in Table 4.4. Case study 2 yielded better overall RMSE values, due to

4. Summary of Results

the object being closer since it was within grasping reach of the manipulator. The closer distance to the object helped the detector to better detect the object without mistakenly including the Qualisys marker in the detection.

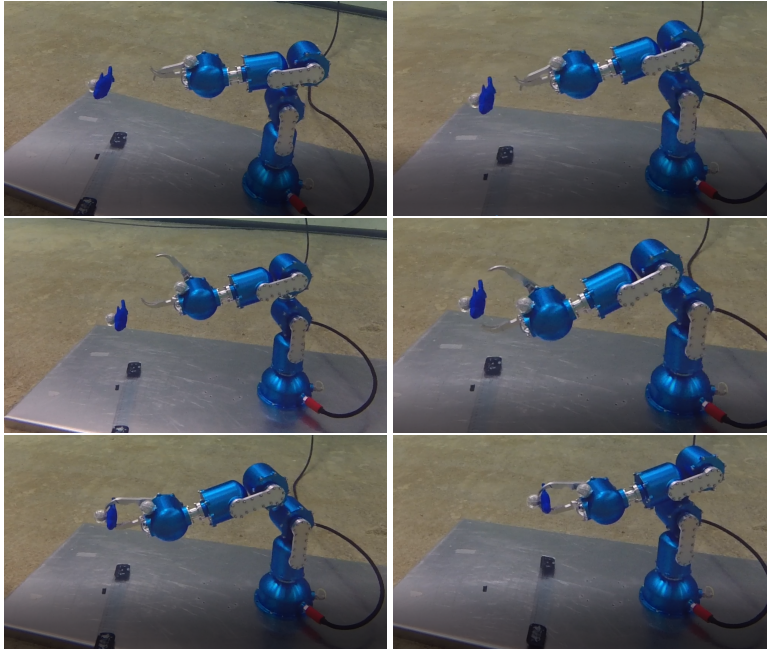


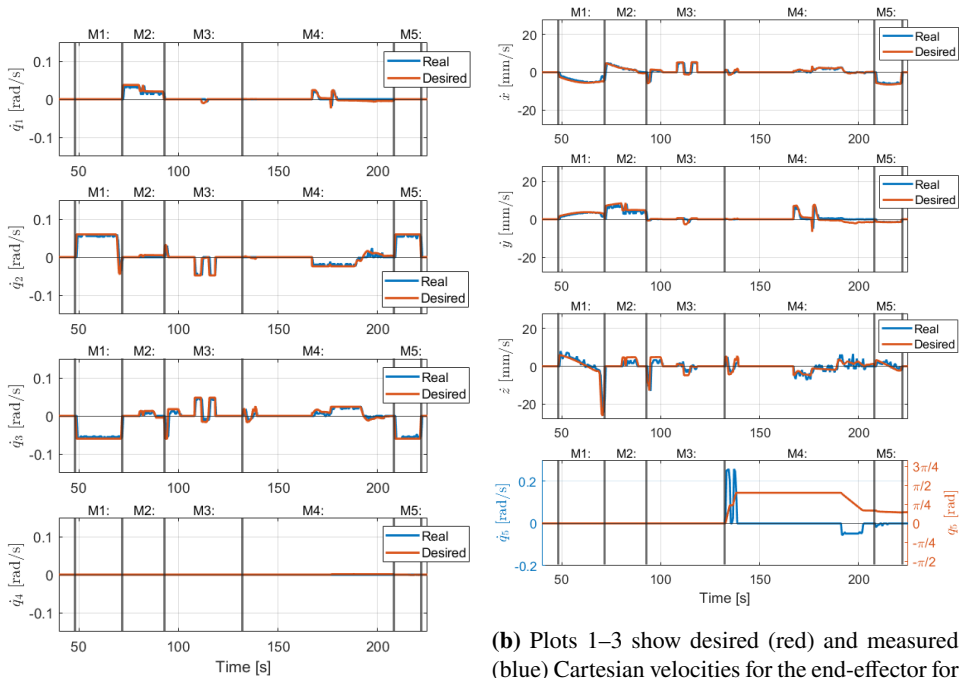
Figure 4.14: Grasping operation for one experiment.

Table 4.4: Distance estimator RMSE values for case study 2.

RMSE fish	RMSE skeleton	RMSE total
8.96 mm	10.47 mm	9.21 mm

The results of one experiment are presented in Figure 4.15, which shows velocity plots of the experiment. Figure 4.15(a) shows plots of the desired and measured joint velocities, and Figure 4.15(b) shows plots of the Cartesian velocities, the gripper angle, and the angular rate. The long time period of zero velocities at the start of the grasping operation (M4 in the plots) is due to the manipulator planning the grasp sequence, which involves calculating gripping rotation angle and gripping approach. The object was successfully grasped after the retrieval of the object (M5 in the plots) was finished at around 220 s.

4.6. Article 6: Autonomous Underwater Grasping using a Novel Vision-Based Distance Estimator



(a) Desired (red) and measured (blue) joint velocities for a grasping experiment for joints 1–4, with time on the x-axis and angular velocity on the y-axis.

(b) Plots 1–3 show desired (red) and measured (blue) Cartesian velocities for the end-effector for a grasping experiment, with time on the x-axis and velocity on the y-axis. Plot 4 shows time on the x-axis and gripper angle in red with the y-axis on the right and gripper angular velocity in blue with the y-axis on the left.

Figure 4.15: Joint velocities and Cartesian velocities for a grasping experiment. The vertical lines and M1–M5 represent the modes in the grasping procedure. The modes are: M1, get into base position; M2, search for object; M3, estimate distance; M4, grasp object; and M5, retrieve object.

4.7 Article 7: Underwater Vehicle Manipulator System (UVMS) with BlueROV2 and SeaArm-2 Manipulator

Purpose and Novelty

The purpose of this article is to develop a digital twin for validation and testing of UVMSs. The novel contribution of this article is the design and implementation of a digital twin for low-cost UVMSs. In addition, a simulated pool environment is designed to form a testing platform for the digital twin. The twin can be applied to low-cost equipment such as the BlueROV2 underwater vehicle and the SeaArm-2 manipulator.

Methodology

The simulator is a mathematical and physical model of a UVMS, designed based on a real system to minimize the complexity of transferring knowledge and software solutions from simulation to real-world experiments. The digital twin in this work was designed based on the BlueROV2 underwater vehicle with the SeaArm-2 manipulator attached underneath. Interactions between the two robots and between the UVMS and operator were communicated through ROS [78]. The digital twin adopted identical communication protocols and software design to the physical system so that it resembled it as closely as possible. For the simulator, an open-source simulator was used, with the environment and the robots included in the UVMS modeled from 3D graphics files. The environment and UVMS were modeled to represent a realistic version of the real system, and to enable systems such as control algorithms and communication protocols to be directly transferable to real world experiments.

The presented system was tested for real-time applications and time delays by measuring and comparing the time slots in which actions were performed in different sections of the system. The time delays between the measurements revealed the time delays present in the system. The system was tested by sending velocity commands and measuring the time delays between the different sections of the system, that is, the processes in the operator computer, the physical UVMS, and the digital twin.

Results

The simulated environment was modeled in the Gazebo simulator. The simulator software was integrated with ROS such that the software running on the operator computer listened to data transmitted from the simulator instead of the hardware. Such software on the operator computer included control systems, computer vision frameworks, and motion planners. In this way, the simulator replaced the hardware

4.7. Article 7: Underwater Vehicle Manipulator System (UVMS) with BlueROV2 and SeaArm-2 Manipulator

and dealt with matters such as thruster forces and torques, manipulator joint velocities, joystick commands, and camera images. The MC-lab environment modeled in the simulator is depicted in Figure 4.16. Gravity, density of water, hard walls and floor, and dimensions equal to the real-world laboratory were modeled to provide a realistic representation of the environment.

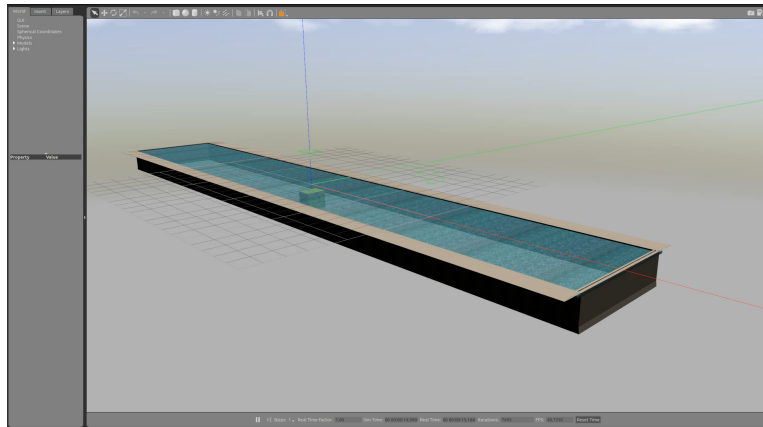
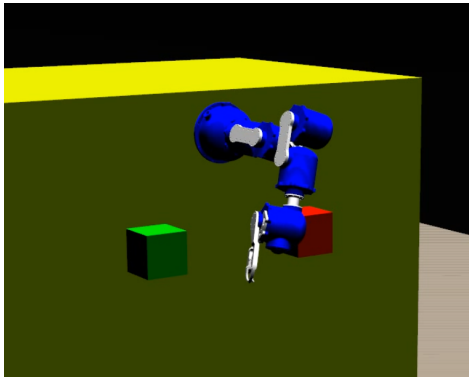


Figure 4.16: The MC-lab pool in the Gazebo simulator.

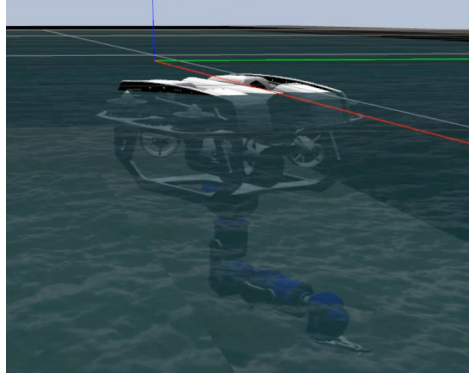
The SeaArm-2 digital twin was assembled based on 3D graphics files for the different modules, links, and the gripper, and is depicted in Figure 4.17(a). Correct rotations and joint limits were included in the assembly to represent the real world as accurately as possible. The digital twins of the SeaArm-2 manipulator and the BlueROV2 were modeled as separate robots, to reflect the real world in which the combined UVMS was built from two individual robots. The two robots could be controlled individually or as a joint system, depending only on the applied control system. The combined UVMS digital twin is depicted in Figure 4.17(b), where it is shown in the modeled MC-lab environment. In this image it can be seen how the UVMS floated in the water surface with the top of the BlueROV2 visible above the waterline, while the rest of the UVMS was submerged.

The time delays between the sections of the system are listed in Table 4.5. The largest time delay occurred between the times when the UVMS demanded action to the thrusters and when the UVMS transmitted the log to the operator computer. This time delay was caused by the UVMS onboard computer collecting and logging all status updates of the vehicle. However, the overall time delays in the system had a negligible impact on its real-time capabilities. There was an average time delay of 0.110 s between a velocity command being initialized on the operator computer and being registered by the digital twin.

4. Summary of Results



(a) The SeaArm-2 manipulator digital twin.



(b) The UVMS digital twin with the BlueROV2 vehicle and the SeaArm-2 manipulator.

Figure 4.17: Digital twins of the SeaArm-2 manipulator and the UVMS applied in the MC-lab pool environment.

Table 4.5: Time delay between different sections of the system.

Section	Time delay
Velocity command input to operator computer	0.000 s
UVMS receive velocity command	0.007 s
UVMS demands action to thrusters	0.007 s
UVMS transmits status log to operator computer	0.096 s
Operator computer logs status	0.107 s
Digital twin receives status log	0.109 s
Digital twin demands action to thrusters	0.110 s

4.8 Article 8: Dynamic Bayesian Networks for Reduced Uncertainty in Underwater Operations

Purpose and Novelty

The purpose of this article is the presentation of a novel framework for modeling dynamic BBNs for online risk assessment in underwater operations. The novel contribution of the article is the design and development of the BBN, in addition to providing an overview of existing BBN frameworks and a discussion about the advantages of the presented BBN over existing solutions.

Methodology

The BBN developed in this work is an open-source python library based on the Bayesian module of the pomegranate library [69]. It extends pomegranate in two ways, by (1) enabling easy calculations and implementations of conditional probability tables (CPTs) and (2) visualizing the Bayesian network in a similar manner to other GUI-based software, thereby incorporating the best of both worlds. Furthermore, a case study is presented in which the BBN was utilized to reduce uncertainties in a UVMS. The case study was an attempt to further improve the methods and results presented in Article 2 and Article 3.

Results

The BBN is referred to as Another Bayesian Belief Analyzer (ABBA). ABBA was designed to circumvent the use of expert judgment by analyzing historic data of the process, while leaving it to the user to decide the causal dependencies to exploit the user's overall knowledge of the process. The library takes a .txt file as input in which all the relevant nodes in the BBN are described, in addition to a .csv file including historic data to calculate the CPTs. The .txt file has the format `name1;name2;scale;limits;parent nodes`, where the attributes are described in further detail in Table 4.6. The .csv file should be in a specific format for the library to be able to read it correctly, where each node has its own column of historic data and where each row represents a measurement. If the historic data are collected from a larger database, they may contain insignificant or irrelevant datapoints. This is of no concern because the relevant datapoints that should be included in the BBN are specified in the input .txt file.

The BBN tool is simple to use and provides good visualization of the network. A built-in generic classifier enables the tool to be directly incorporated in dynamic systems where evidence is derived from measurements. The classifier characterizes measured values within the scales and limits as described in the input .txt file and updates the network based on these measurements. The BBN model is also capable

4. Summary of Results

Table 4.6: Description of the attributes in the input file to ABBA.

Attribute	Description
name1	Name of the node (should not contain spaces)
name2	Name of the attribute in the .cvs file
scale	Scale defining different states in the CPTs
limits	Limits for classification within the relevant scales
parent nodes	Names of all parent nodes (leave blank if none)

of being updated in real time if measurements, time dependencies, or other factors in the overall system require it.

The goal the case study was to investigate if ABBA can be used to formulate a value of belief in the estimated distance derived from the systems presented in Articles 2 and 3. The BBN designed for this case study is presented in Figure 4.18, showing the causal dependencies between the nodes. Each node includes a CPT as depicted in Figure 4.19(a) for the arm’s x-position node, and, as shown in Figure 4.19(b) for the target node, position certainty.

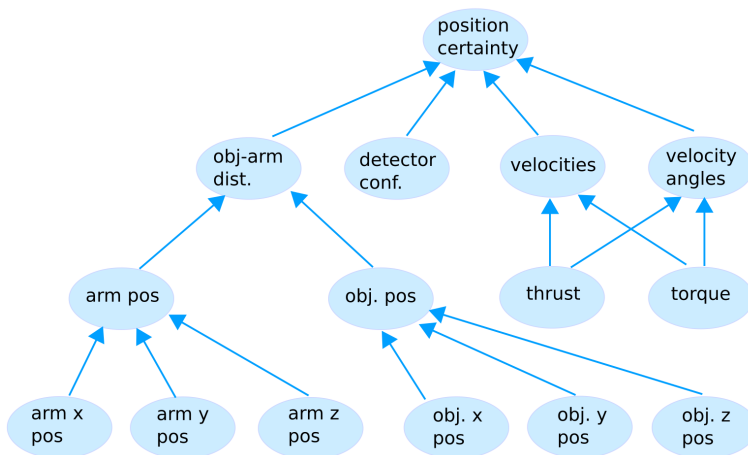
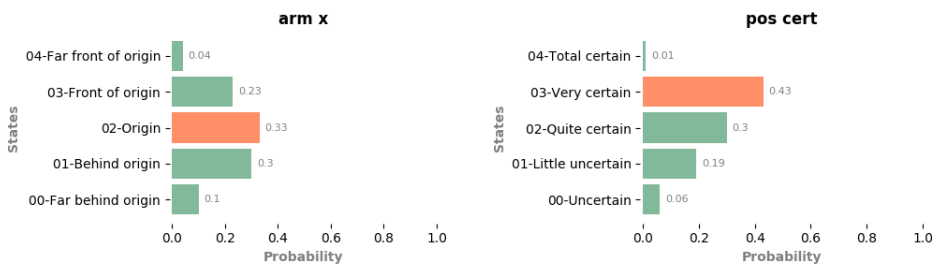


Figure 4.18: BBN structure with causal dependencies between nodes.

4.8. Article 8: Dynamic Bayesian Networks for Reduced Uncertainty in Underwater Operations



(a) CPT for the arm x-position node.

(b) CPT for the target node, position certainty.

Figure 4.19: Visualization of CPTs in the BBN structure, with conceptual data.

4.9 Article 9: Autonomous Subsea Intervention (SEAVENTION)

Article 9 presents the main results and latest developments in a 4-year project called Autonomous Subsea Intervention (SEAVENTION). The article is a collective presentation of the project results combining the methods developed for autonomous IMR in subsea oil and gas operations with UUVs. Furthermore, the article provides an overview of how subsea IMR operations with UUVs are typically performed in the oil and gas industry and the current trends. Since the article is an overview of a collection of different studies, the article itself has no novel or unique results. The article is therefore not discussed further here, but the respective articles for the relevant results included in Article 9 can be found in [42, 61, 7, 93] and in the enclosed Articles 2, 3, and 6.

4.10. Article 10: System Integration of Underwater Vehicle Manipulator System (UVMS) for Autonomous Grasping

4.10 Article 10: System Integration of Underwater Vehicle Manipulator System (UVMS) for Autonomous Grasping

Purpose and Novelty

The purpose of this article is the research and development of a system integration structure for low-cost UVMSs, with implementation on a case study UVMS with experimental testing of software and hardware integration. The novel contributions of the article are threefold. First, the system integration of a low-cost UVMS was discussed and implemented on a case study UVMS. Second, the software integration was verified with simulator experiments. Third, the system integration as a whole was verified, with experimental testing in a laboratory pool using the case study UVMS.

Methodology

A system integration structure was developed for identifying and developing a low-cost UVMS. The system integration included the hardware and software integration on a three-machine UVMS consisting of an operator computer, an underwater vehicle, and a manipulator. All software included in the system utilized ROS as a platform, and the three-machine setup allowed for simple and uncomplicated modifications and/or upgrades on the individual machines without interrupting the other ones. The system was tested in simulation-based experiments with the digital twin of the case study robots. The experiments included autonomous fixed-base grasping using the manipulator digital twin, in which the computer vision system and manipulator controller were adopted from the real UVMS system. Experiments in a laboratory pool were conducted to verify the system integration as a whole, with special consideration given to hardware solutions. Communication in the three-machine setup, computer vision system, and an SMC were verified through a simple DP operation in the MC-lab pool.

Results

Typically, underwater manipulators require independent actuation systems and a set of sensors and consist of multiple DOFs. From a system integration point of view, it is thus simplest to view the manipulator as an independent robot. In this setup, the manipulator operates as a separate machine, and the three machines are all connected to each other through a network switch. This allows all machines to communicate with each other freely and does not require any of the machines to function as a bridge between the other two. Figure 4.20(a) shows the hardware integration of the case study UVMS, assembled with an operator computer, the

4. Summary of Results

BlueROV2 underwater vehicle, and the SeaArm-2 manipulator. Figure 4.20(b) shows the software integration of the same system, in which the nodes represent their respective machines.

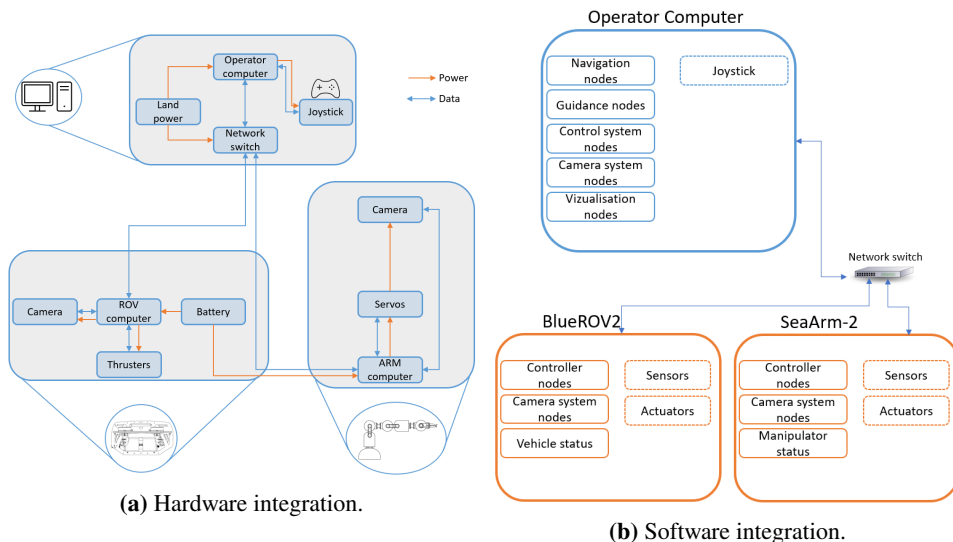


Figure 4.20: Three-machine system integration with ROS on a UVMS including an operator computer, an ROV, and a manipulator (here depicted with the BlueROV2 underwater vehicle and the SeaArm-2 manipulator).

The simulation-based experiments applied the computer vision system from Article 6 and the digital twin from Article 7. The manipulator digital twin can be seen in the simulated MC-lab environment with corresponding end-effector camera view in Figure 4.21.

Figure 4.22(a) shows plots of the joint positions for a single experiment, and Figure 4.22(b) shows plots of the relative distance between the gripper and the object. The dotted red line indicates when the distance estimator was turned on and the solid red line indicates when the distance estimation was terminated. Similarly, the blue dotted line indicates when the grasping procedure started, and the solid blue line corresponds to the time when the procedure was finished and the object was grasped. The joint position changes seen in Figure 4.22(a) before the red dotted line were caused by manual control of the manipulator. The last plot in Figure 4.22(b) shows the absolute relative gripper-object distance—it can be seen that, at approximately 60 s, the gripper maintained a constant distance of about 15 mm from the object. The object was grasped at this moment and the error of 15 mm is believed to derive from inaccuracies in the distance estimator when estimating the size of the object. A position error of 15 mm indicates a favorable

4.10. Article 10: System Integration of Underwater Vehicle Manipulator System (UVMS) for Autonomous Grasping

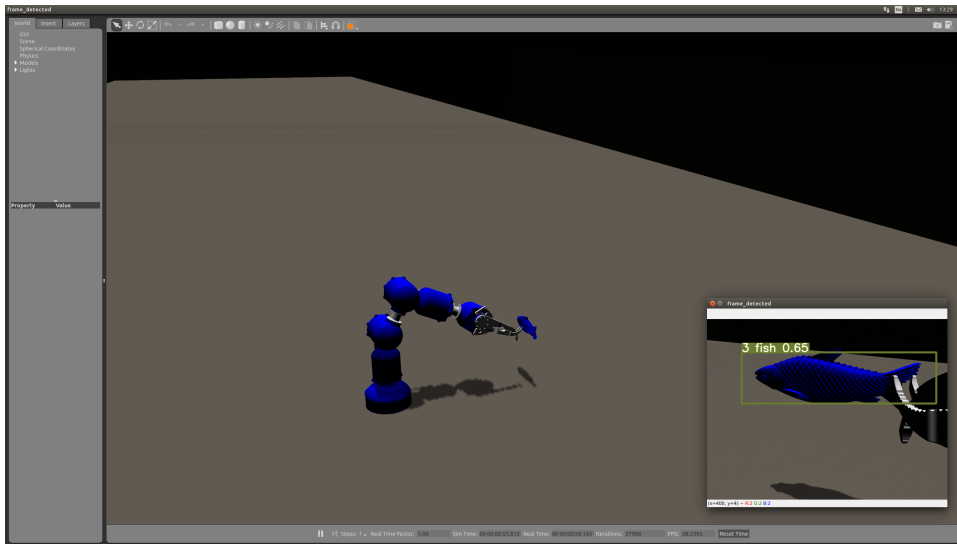


Figure 4.21: The SeaArm-2 manipulator and the fish object with camera display in the MC-lab environment in the Gazebo simulator.

result. The gripper itself had 60 mm gripper fingers and could grasp objects along any of them. The exact gripper position is defined to be in the center of these gripper fingers, meaning there is an acceptable error of ± 30 mm

4. Summary of Results

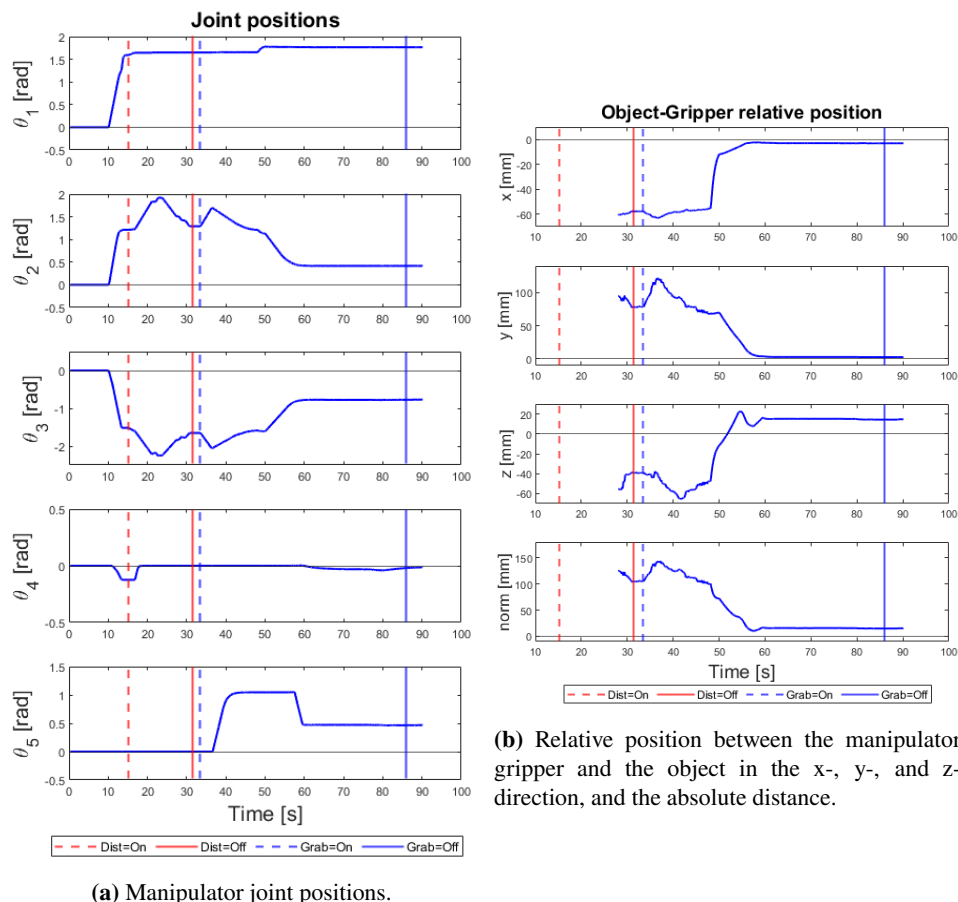


Figure 4.22: Manipulator joint positions and relative gripper-object position for a simulation-based autonomous fixed-base grasping experiment.

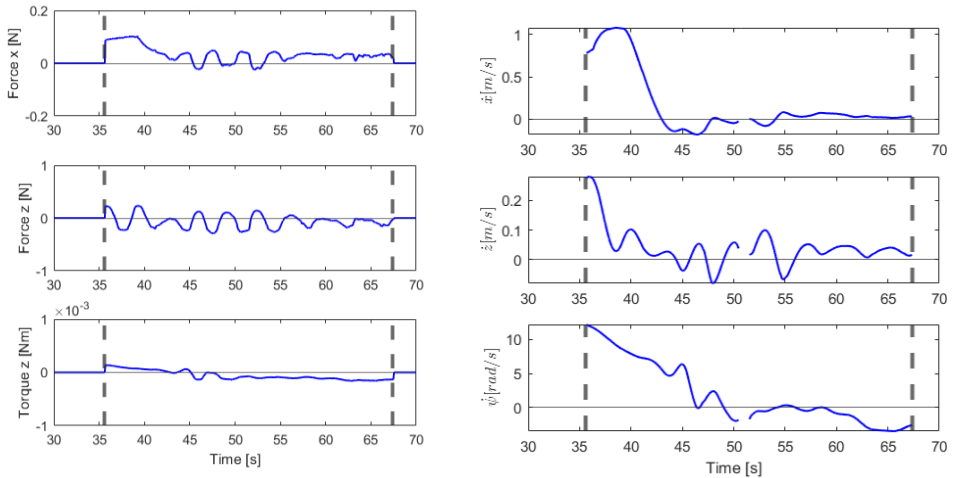
The laboratory experiments adopted the SMC design from Articles 2 and 3 and the computer vision system from Article 6, and consisted of simple DP operations to verify the new system integration setup. The UVMS is depicted during DP operations in Figure 4.23, with the fish object located in front of the vehicle. The results from one of the DP experiments are plotted in Figure 4.24(a), in which the forces and torques commanded by the SMC can be seen, and in Figure 4.24(b), which shows the velocity errors. The velocity errors show some fluctuations, but they remain close to zero. As Figure 4.24(a) illustrates, the controller provided oscillating forces in the x- and z-directions. However, the oscillating behavior decreases toward the end of the plot. The forces oscillate around a slightly positive surge force in the x-direction and a slightly negative heave force in the z-direction. The same behavior is seen in a convergence toward a negative torque in yaw. The

4.10. Article 10: System Integration of Underwater Vehicle Manipulator System (UVMS) for Autonomous Grasping

positive surge force and negative yaw torque are believed to have resulted from counteracting forces originating from the tether. Moreover, the slightly negative heave force originated from counteracting the slightly positive buoyancy of the UVMS.



Figure 4.23: UVMS in DP operations in the MC-lab with relevant object in front of the vehicle.



(a) SMC-commanded forces in the surge and heave directions (in N) and torques in the yaw direction (in Nm).

(b) Velocity error (in m/s) in the surge and heave directions and yaw rate (in deg/s).

Figure 4.24: SMC output and vehicle velocities.

Chapter 5

Conclusions and Further Work

This chapter draws overall conclusions from the thesis and the enclosed articles, and discusses how the work meets the research objectives presented in Section 1.2. Potential future research avenues are also discussed, with a number of possibilities presented on how to continue the work on designing intelligent solutions to increase autonomy in underwater operations.

5.1 Conclusions

The main purpose of this PhD research has been to investigate intelligent solutions for low-cost underwater robotic equipment to improve upon existing methods, and to develop new equipment and methods with increased autonomy. One of the main goals was to design and build the underwater manipulator SeaArm-2, and develop generic intelligent solutions for small electric manipulators, with the SeaArm-2 manipulator being used for laboratory experimental verification of the methods developed. The topics covered in this thesis are related to underwater guidance, navigation, and intervention for small electric vehicle and manipulator systems, with a focus on applications of visual aided machine learning. Moreover, object detection, object tracking, and transfer learning applications have been implemented in the robotic systems to increase their autonomous capabilities, which again was verified in laboratory experiments.

In Articles 1 and 5, transfer learning algorithms were applied in an attempt to reduce the gap between a simulated environment and real-world scenarios. A framework called CycleGAN was applied to datasets containing simulated images and real footage of a subsea template to learn the mapping between the domains. In Article 5, the trained network was used to generate a completely new dataset of

5. Conclusions and Further Work

realistic images containing the subsea template with corresponding labels describing the locations of the template in the image frame. The labels were obtained by transferring the segmentation masks from the simulated input image to the new generated image. The newly generated dataset was utilized to train an object detector that was used in the guidance system of a DP approach. The system was tested in experiments using the BlueROV2 vehicle in the MC-lab pool, which verified that the system was able to efficiently detect the real subsea template after training only on synthetically generated data. The work showed that by using GANs, it is possible to generate synthetic datasets to train other visual aided machine learning applications. This is highly relevant for operations where large datasets may be hard to obtain. In addition, by using the segmented information in the simulated environment, this process generated a labeled dataset, which saved time and costs spent on meticulously labeling each individual image.

Article 4 presented the SeaArm-2 manipulator, a small, electric, modular manipulator with an integrated end-effector camera. The manipulator is designed for independent use and to be mounted on a vehicle in a UVMS system. The end-effector camera provides unique possibilities for autonomous applications, and the manipulator can help increase autonomy in IMR operations. The manipulator's capabilities as a standalone manipulator were verified in experimental testing in Articles 4 and 6, and in Article 6, a novel distance estimator designed for manipulators with end-effector cameras was presented and tested.

Articles 2, 3, and 10 focused on dynamic positioning of small electric underwater vehicles, with the aim to keep a UVMS stationary during intervention operations. The system's guidance system implemented computer vision with object detection algorithms and various distance estimators to station the vehicle relative to a detected object of interest. In Article 3, the UVMS grasped the object with the SeaArm manipulator, while using an SMC for station keeping relative to the object. The UVMS in Article 10 included the newer manipulator version, the SeaArm-2, along with a new system integration setup, and performed station keeping in laboratory experiments, and autonomous fixed-base grasping with the manipulator in a simulated environment.

In Article 7 and 10, a digital twin and system integration structure of low-cost electric UVMSs was presented. The digital twin and experimental validation were based on a UVMS assembled with the BlueROV2 vehicle and the SeaArm-2 manipulator. The digital twin provided an excellent platform for testing and verification of control algorithms, guidance systems, communication protocols, and additional software solutions, before deploying the real system in physical experiments. The digital twin also offered strong visualization capabilities of the system behavior during real operations.

Article 8 introduced a novel framework for modeling dynamic BBNs and presented a conceptual design for the framework to improve estimations in a UVMS performing DP and autonomous intervention. Furthermore, Article 9 presented results of different articles working with autonomy of UUVs aimed toward the oil and gas industry.

Summarized, Articles 1 and 5 attempted to enhance simulation results, while Articles 2, 3, 4, 6, and 10 provided experimental testing of autonomous functionalities. Finally, Articles 7 and 10 presented digital twins and simulation-based testing, Article 8 attempted to improve autonomous accuracy, and Article 9 is a review article presenting results of several articles addressing autonomy of UUVs.

5.2 Addressing Research Objectives

In the following, the research objectives that were stated in Section 1.2 are addressed through the presented articles.

Research Objective 1: “Investigate and research the use of machine learning and intelligent software solutions to increase autonomy in underwater intervention tasks.”

The first objective specifies the development of low-cost solutions for increased autonomy, and this was addressed in Articles 2, 3, 4, 6, 8, 9 and 10. Vision-based object detectors and trackers were implemented to recognize and locate objects of interest in Articles 2, 3, 4, 6 and 10. The object detectors were based on neural networks and trained using machine learning algorithms on custom datasets containing images and corresponding labels of the relevant objects. The detectors provided excellent detection capabilities and were able to locate objects with high precision in underwater scenarios. When combined with different scaling functions and 3D feature extractors, the object detectors also provided 3D position information on objects of interest. In Article 8, a novel BBN was developed for online risk assessment in underwater operations, providing uncomplicated implementation and visualization and verification tools. The BBN uses machine learning algorithms to train CPTs on historic data, and the network can analyze data and produce probabilities in real time. A conceptual design of implementing the BBN to reduce uncertainties in autonomous underwater intervention operations was also presented. Article 9 presented the main results and latest developments in the SEAVENTION project, combining methods developed for autonomous IMR in subsea oil and gas operations with UUVs. The results provide important knowledge about enabling fully autonomous UUV missions in cases where human operator involvement is not possible or desirable.

5. Conclusions and Further Work

Research Objective 2: “Research and develop vision-based machine learning solutions to improve the visual authenticity of simulated underwater environments to improve the transfer of knowledge from simulated environments to real-world applications.”

The second objective was addressed in Articles 1 and 5. The proposed methods were first explored in Article 1, in which algorithms were used to perform a mapping between simulated images and real world footage. The methods demonstrated the possibilities of generating realistic images from basic computer graphics, which can aid in improved simulator graphics or generation of synthetic image datasets that resemble real footage. The methods were further investigated in Article 5, in which they were used to generate datasets with realistic images and corresponding labels of an underwater structure. The dataset was used in training an object detector that was verified in a DP approach of an underwater vehicle in experimental testing. This proved that the synthetically made dataset had high enough quality to be used in training a system for real-life operations.

Objective 3: “Design, develop, and assemble efficient hardware and software for the SeaArm manipulator, and verify the manipulator capabilities with experimental testing.”

The third objective was addressed in Articles 4, 6, 7, and 10. A considerable amount of effort was invested in designing, developing, and assembling the SeaArm-2 manipulator. The finished manipulator was first presented in Article 4, including kinematic properties, kinematic control, and simple server-client communication. Experimental verification of its capabilities was also conducted through autonomous fixed-base grasping experiments using computer vision applied to the monocular camera as guidance. With the positive findings of Article 4, the computer vision system was expanded in Article 6 to perform autonomous fixed-base grasping of objects of unknown size. Furthermore, Articles 7 and 10 improved communication protocols and rebuilt the software based on ROS, in addition to introducing a digital twin of the manipulator. The digital twin holds the same kinematic properties and joint limitations as the real manipulator, is controlled through the same type of controllers, and includes an end-effector camera with identical intrinsic parameters. The digital twin was used for simulation-based experiments in Articles 7 and 10 and has the potential to be used for graphical representation during real-life operations.

Objective 4: “Conduct experimental testing to verify proposed solutions in underwater manipulation as well as vehicle-manipulator systems.”

5.3. Other applications and limitations of the work

The last objective concerns experimental work and was addressed in Articles 2, 3, 4, 5, 6, and 10. An object-relative DP system was developed in Article 2, in which an underwater vehicle performed station keeping relative to an object detected by computer vision. Article 3 built on results from Article 2 by adding an attached manipulator and performing object grasping. Further improvement on the manipulator system led to Article 4, in which the SeaArm-2 manipulator was introduced and used to perform autonomous fixed-base grasping. The work in Article 4 was further improved in Article 6 with an improved camera system and the development of a novel distance estimator, and the Qualisys Motion Tracker system was used for verification of the distance estimator and grasping experiments. Grasping using the integrated end-effector camera yielded good results and robust grasping for different objects in the various studies. Article 5 presented experimental work to verify the object detector that was trained on a synthetic dataset generated by machine learning algorithms. The system integration of a new setup of UVMSs was presented in Article 10, where an underwater vehicle performed object-relative DP with a manipulator attached to verify the hardware and software integration in the system. All experimental work was conducted in the MC-lab and the Qualisys Motion Tracker system provided true position data for post-processing and verification of the methods.

5.3 Other applications and limitations of the work

The work conducted as part of this thesis have been focused on low-cost solutions with experimental testing to verify the proposed theory and methods. The experiments have mainly focused on autonomous detection, localization, and grasping using a manipulator. However, the methods are far from limited to simply grasping and picking up objects. Efficient detection and localization of different objects or features relative to a manipulator can provide essential information in operations such as valve handling, insertion and removal of hot stabs, and repairing holes in aquaculture net pens. Such tasks are represented with defined components and equipment which simplifies automatic, and even autonomous, solutions. Hot stabs with known appearance and size can easily be detected and localized by a computer vision system, and a system including the presented SeaArm-2 manipulator can easily maneuver and operate such tasks. Furthermore, undesirable holes in aquaculture net pens can be detected and localized using computer vision and temporarily fixed by applying patches using a customized patching gripper. The highly customizable manipulator can also be assembled with different numbers of joints, links, DOFs, and grippers to expedite the specific task.

As the thesis has focused on low-cost solutions, the presented methods and equipment have certain limitations compared to other systems. A key limitations that

5. Conclusions and Further Work

involve most of the novelties in this thesis is redundancy. A central part in the low-cost adaptation is to omit redundant equipment and sensors to design a system capable of performing the desired tasks using solely necessary equipment. Such systems are naturally highly dependent on involved equipment and are vulnerable to equipment and method failure. The simplest way to strengthen these limitations is to include more redundancy in the form of extra equipment and sensors. However, this would contradict with the purpose of the thesis to design low-cost solutions. Nonetheless, redundancy is often a design requirement and to avoid failures and down-time of systems it could even be profitable to include more redundancy.

5.4 Further Work

In the future, it is expected that there will be many relevant technological advances allowing for the exploration of new possibilities. Naturally, this will provide opportunities for further development of the work presented in this thesis. The world of visual aid is constantly evolving, and new and improved cameras, lights, and overall camera systems are continuously entering the market. Computer vision is also in constant development and, considering the immense growth in both quantity and quality of solutions in the last decade, it is expected that even more robust and accurate systems will emerge in the future. This continued development will undoubtedly improve detecting and tracking capabilities, and it is likely that systems with entirely new capabilities will be developed as well. The systems developed and tested in this work are highly relevant for the future, and prove that computer vision has a significant role to play in underwater operations. With the presumed development of computer vision and machine vision solutions, the relevant applications for the presented methods and perspectives of this thesis will also increase.

Transfer learning, or other methods for enhancing simulators and improving the reuse of simulator results in real operations, are of interest for operators in underwater environments. Learning algorithms require either data or large amounts of physical trials for training. The approach presented in this work considered domain adaption to learn the mapping between a simulated environment and real footage. This method can be used to generate unlimited amounts of new data, in contexts where data may be hard to acquire. The method also has the potential to increase the authenticity of simulations by mapping an entire simulated environment to realistic scenery. For underwater applications, this is highly relevant given the vast amount of possible phenomena and effects that should be modeled in order to manually create a realistic model of the underwater scene. Further research in this field will support the development of machine learning in underwater operations, by providing an applicable environment where learning algorithms can train.

Electric manipulator systems, especially small, low-cost manipulators, have earned role in underwater intervention in recent years. The presented manipulator, SeaArm-2, offers a versatile research platform for exploration of autonomous intervention, and the modular design, integrated camera, and system integration may be an inspiration for development of future small, electric, low-cost manipulators. Development of hardware solutions is also expected in the future, and stronger and more compact electric motors should create possibilities for even stronger and more effective manipulators. To fulfill the potential of such manipulators, further research in intervention systems should be explored. Effective control systems, and perhaps especially guidance and navigation systems, should be investigated for effective autonomous capabilities. This thesis presented inverse kinematic solutions, where the end-effector camera played an important role in the guidance system. The results yielded good success rates of autonomous intervention capabilities, even though occlusions and the limited frame of view diminished somewhat the full potential of the methods. Further research in underwater vision systems should therefore be explored, since it may play a pivotal role in enhancing future intervention capabilities.

Underwater intervention is highly reliant on UVMSs today, and development in R-ROVs and other underwater vehicles are in great demand. R-ROVs are intended to stay subsea permanently, or semi-permanently, and their internal autonomous capabilities will be essential in the efficiency of the systems. Such systems are today developed to support both autonomous tetherless operations and remote-controlled operations, both with autonomous functionalities for docking, station keeping, transit, and intervention. The methods presented in this thesis explored autonomous intervention, both for single manipulator systems and for UVMSs. The tested methods fit grasping operations of floating or unattached objects that are within the lifting capabilities of the manipulator. Through further research, similar methods can be developed for other intervention operations, such as approaching and fixing holes in the nets of fish cages in aquaculture, welding subsea pipes in the oil and gas industry, and operating subsea control panels. In conclusion, further research within all the presented methods and solutions will aid in the establishment of autonomy in underwater operations, especially regarding autonomous intervention.

References

- [1] Blender. <https://www.blender.org/>. [Online; accessed 29-June-2022].
- [2] B. Academic. Neural network. <https://academic.eb.com/levels/collegiate/article/neural-network/126495>. [Online; accessed 21-November-2018].
- [3] R. Almodfer, S. Xiong, M. Mudhsh, and P. Duan. Multi-column deep neural network for offline arabic handwriting recognition. In A. Lintas, S. Rovetta, P. F. Verschure, and A. E. Villa, editors, *Artificial Neural Networks and Machine Learning – ICANN 2017*, pages 260–267, Cham, 2017. Springer International Publishing.
- [4] G. Antonelli. *Underwater Robotics*. Springer, Cham, 2014.
- [5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016.
- [6] 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 *OCEANS 2015-MTS/IEEE Washington*, pages 1–10. IEEE, 2015.
- [7] M. Bjerkeng, A. Transeth, and T. Myhre. Spatial and temporal autocalibration for underwater vehicles. *IFAC-PapersOnLine*, 52(21):260–264, 2019.
- [8] BlueRobotics. Bluerov2. <https://bluerobotics.com/store/rov/bluerov2/>. [Online; accessed 06-January-2022].
- [9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

REFERENCES

- [10] F. Bonin-Font, G. Oliver, S. Wirth, M. Massot, P. L. Negre, and J.-P. Beltran. Visual sensing for autonomous underwater exploration and intervention tasks. *Ocean Engineering*, 93:25 – 44, 2015.
- [11] M. Broström. Real-time multi-object tracker using yolov5 and deep sort. https://github.com/mikel-brostrom/Yolov5_DeepSort_Pytorch, 2020.
- [12] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, IEEE Journal of Robotics and Automation, 2004.
- [13] Y. Cao, K. Lu, X. Li, and Y. Zang. Accurate numerical methods for computing 2d and 3d robot workspace. *International Journal of Advanced Robotic Systems*, 8(6):76, 2011.
- [14] C. Chen and Y. F. Zheng. Passive and active stereo vision for smooth surface detection of deformed plates. *IEEE Trans. Ind. Electron.*, 42:300–306, 1995.
- [15] M. Chen, Y. Tang, X. Zou, Z. Huang, H. Zhou, and S. Chen. 3d global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and slam. *Computers and Electronics in Agriculture*, 187:106237, 2021.
- [16] 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*, 2019.
- [17] 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*, 2019.
- [18] Z. Chen, Z. Zhang, F. Dai, Y. Bu, and H. Wang. Monocular vision-based underwater object detection. In *Sensors*, 2017.
- [19] R. D. Christ and R. L. Wernli. *The ROV manual: A user guide for remotely operated vehicles*. Butterworth-Heinemann is an imprint of Elsevier, second edition, 2014.
- [20] deepmind.com. Alphago. <https://deepmind.com/research/alphago/>. [Online; accessed 17-November-2018].
- [21] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Trans. ASME E, Journal of Applied Mechanics*, 22:215–221, June 1955.

-
- [22] M. A. Denket, editor. *Frontiers in Robotics Research*. Nova Science Publishers Inc, uk edition, nov 2006.
- [23] O. A. N. Eidsvik, B. O. Arnesen, and I. Schjølberg. Seaarm-a subsea multi-degree of freedom manipulator for small observation class remotely operated vehicles. In *2018 European Control Conference (ECC)*, pages 983–990, 2018.
- [24] J. Gancet, D. Urbina, P. Letier, M. Ilzokvitz, P. Weiss, F. Gauch, G. Antonelli, G. Indiveri, G. Casalino, A. Birk, M. F. Pflingsthor, S. Calinon, A. Tanwani, A. Turetta, C. Walen, and L. Guilpain. Dexrov: Dexterous undersea inspection and maintenance in presence of communication latencies. *IFAC-PapersOnLine*, 48(2):218–223, 2015. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles.
- [25] A. Ghosal. *Manipulator Kinematics*, pages 1777–1808. 09 2015.
- [26] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [27] B. O. A. Haugaløkken, M. B. Skaldebø, and I. Schjølberg. Monocular vision-based gripping of objects. *Robotics and Autonomous Systems*, 131:103589, 2020.
- [28] J. Hegde, I. B. Utne, and I. Schjølberg. Applicability of Current Remotely Operated Vehicle Standards and Guidelines to Autonomous Subsea IMR Operations. volume Volume 7: Ocean Engineering of *International Conference on Offshore Mechanics and Arctic Engineering*, 05 2015.
- [29] 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*, 195:106–644, 2020.
- [30] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, A. Hogan, lorenzomamma, tkianai, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Hatovix, J. Poznanski, L. Yu, changyu98, P. Rai, R. Ferriday, T. Sullivan, W. Xinyu, YuriRibeiro, E. R. Claramunt, hopesala, pritul dave, and yzchen. *ultralytics/yolov5: v3.0*, Aug. 2020.
- [31] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004.

REFERENCES

- [32] B. Lab. Bravo 7. <https://blueprintlab.com/media/docs/75773/Bravo7-Mk2-Datasheet.pdf>. [Online; accessed 27-June-2022].
- [33] C. L. Lawson and H. R. J. *Solving least squares problems*. Englewood Cliffs, N.J., Prentice-Hall, 1974.
- [34] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.
- [35] S. Lin, F. Qin, Y. Li, R. A. Bly, K. S. Moe, and B. Hannaford. Lc-gan: Image-to-image translation based on generative adversarial network for endoscopic images, 2020.
- [36] B. Logic. Subsea docking station (sds). <https://www.bluelogic.no/news-and-media/subsea-docking-station-sds->. [Online; accessed 24-June-2022].
- [37] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding, and S. Wen. Pp-yolo: An effective and efficient implementation of object detector, 2020.
- [38] Madjidi, Hossein and Negahdaripour, Shahriar. On robustness and localization accuracy of optical flow computation for underwater color images. *Computer Vision and Image Understanding*, 104(1):61–76, 2006.
- [39] A. Makarov, V. Lukić, and O. Rahnema. Distance and speed measurements from monocular images. In N. Kehtarnavaz and M. F. Carlsohn, editors, *Real-Time Image and Video Processing 2016*, volume 9897, pages 130 – 140. International Society for Optics and Photonics, SPIE, 2016.
- [40] G. Marani, S. K. Choi, and J. Yuh. Underwater autonomous manipulation for intervention missions auvs. *Ocean Engineering*, 36(1):15–23, 2009.
- [41] MarketResearchFuture.com. Underwater Robotics Market Research Report. Technical report, July 2020.
- [42] A. Mohammed, J. Kvam, J. T. Thielemann, K. H. Haugholt, and P. Risholm. 6D pose estimation for subsea intervention in turbid waters. *Electronics*, 10(19), 2021.
- [43] M. Moniruzzaman, S. M. S. Islam, M. Bennamoun, and P. Lavery. Deep learning on underwater marine object detection: A survey. In J. Blanc-Talon,

- R. Penne, W. Philips, D. Popescu, and P. Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, pages 150–160. Springer International Publishing.
- [44] T. Morris. *Computer Vision and Image Processing*. Palgrave Macmillan Ltd, United Kingdom, 2004.
- [45] National Research Council. *Autonomous Vehicles in Support of Naval Operations*. The National Academies Press, Washington, DC, 2005.
- [46] N. Nhede. Global offshore wind market to hit \$56.8bn by 2026. *Power Eng. Int.*, Aug. 2021.
- [47] NTNU. Marine cybernetics teaching laboratory. <https://www.ntnu.edu/imt/lab/cybernetics>. [Online; accessed 17-June-2022].
- [48] Oceaneering. Freedom autonomous vehicle. <https://www.oceaneering.com/rov-services/next-generation-subsea-vehicles/freedom/>. [Online; accessed 24-June-2022].
- [49] Oceaneering. Liberty e-rov. <https://www.oceaneering.com/rov-services/next-generation-subsea-vehicles/liberty-e-rov/>. [Online; accessed 24-June-2022].
- [50] Oceaneering. Nexxus: Increased roV and bop intervention capabilities. <https://www.oceaneering.com/rov-services/rov-systems/>, 2016. [Online; accessed 23-June-2022].
- [51] A. O’Riordan, T. Newe, G. Dooly, and D. Toal. Stereo vision sensing: Review of existing systems. In *2018 12th International Conference on Sensing Technology (ICST)*, pages 178–184, 2018.
- [52] C. E. Perez. Automating high-level economic thinking using deep learning, December 2017.
- [53] R. W. Proud, J. J. Hart, and R. B. Mrozinski. Methods for determining the level of autonomy to design into a human spaceflight vehicle: A function specific approach, September 2003.
- [54] H. Qin, X. Li, Y. Zhixiong, and M. Shang. When underwater imagery analysis meets deep learning: A solution at the age of big visual data. In *OCEANS 2015 - MTS/IEEE Washington*, pages 1–5.
- [55] J. Redmon. Darknet: Open source neural networks in c. <http://pjreddie.com/darknet/>, 2013–2016.

REFERENCES

- [56] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [57] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger, 2016.
- [58] J. Redmon and A. Farhadi. Yolov3: An incremental improvement, 2018.
- [59] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1060–1069, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [60] H. Reese. Autonomous driving levels 0 to 5: Understanding the differences, January 2016.
- [61] P. Risholm, P. Ø. Ivarsen, K. H. Haugholt, and A. Mohammed. Underwater marker-based pose-estimation with associated uncertainty. In *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV) Workshops*, pages 3713–3721, October 2021.
- [62] B. robotics. Blueye pro. <https://www.blueye.no/produkter/pro>. [Online; accessed 23-June-2022].
- [63] B. robotics. Blueye x3. <https://www.blueye.no/produkter/x3>. [Online; accessed 24-June-2022].
- [64] E. Robotics. Arm 7e. <https://www.ecagroup.com/en/solutions/arm-7e>. [Online; accessed 27-June-2022].
- [65] P. J. Sanz, P. Ridaó, G. Oliver, G. Casalino, Y. Petillot, C. Silvestre, C. Melchiorri, and A. Turetta. Trident an european project targeted to increase the autonomy levels for underwater intervention missions. In *2013 OCEANS - San Diego*, pages 1–10, Sep. 2013.
- [66] I. Schjølberg and T. I. Fossen. Modelling and control of underwater vehicle-manipulator systems. In *in Proc. rd Conf. on Marine Craft maneuvering and control*, pages 45–57, 1994.
- [67] I. Schjølberg, T. B. Gjersvik, A. A. Transeth, and I. B. Utne. Next generation subsea inspection, maintenance and repair operations. *IFAC-PapersOnLine*, 49(23):434 – 439, 2016. 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016.

-
- [68] I. Schjølberg and I. B. Utne. Towards autonomy in roV operations. *IFAC-PapersOnLine*, 48(2):183–188, 2015. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV 2015.
- [69] J. Schreiber. Pomegranate: fast and flexible probabilistic modeling in python. *Journal of Machine Learning Research*, 18(164):1–6, 2018.
- [70] Y. Shen, P. Luo, P. Luo, J. Yan, X. Wang, and X. Tang. Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2018.
- [71] E. Simetti. Autonomous underwater intervention. *Current Robotics Reports*, 1:117–122, 2020.
- [72] E. Simetti, R. Campos, D. D. Vito, J. Quintana, G. Antonelli, R. Garcia, and A. Turetta. Sea mining exploration with an uvms: Experimental validation of the control and perception framework. *IEEE/ASME Transactions on Mechatronics*, 26(3):1635–1645, 2021.
- [73] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta. Floating underwater manipulation: Developed control methodology and experimental validation within the trident project. *Journal of Field Robotics*, 31(3):364–385, 2014.
- [74] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia. Autonomous underwater intervention: Experimental results of the maris project. *IEEE J Oceanic Eng.*, 43(3):620–639, 2018.
- [75] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia. Autonomous underwater intervention: Experimental results of the maris project. *IEEE Journal of Oceanic Engineering*, 43(3):620–639, 2018.
- [76] S. Sivčev, J. Coleman, E. Omerdić, G. Dooly, and D. Toal. Underwater manipulators: A review. *Ocean Engineering*, 163:431 – 450, 2018.
- [77] J.-J. E. Slotine, W. Li, et al. *Applied nonlinear control*, volume 199. Prentice-Hall Englewood Cliffs, NJ, 1991.
- [78] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.
- [79] E. subsea intervention. Eelume. <https://eelume.com/>. [Online; accessed 24-June-2022].

REFERENCES

- [80] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [81] G. Tech. Uma 1500. <https://www.graaltech.com/products/uma-1500/>. [Online; accessed 27-June-2022].
- [82] K. TeleRobotics. Raptor: Force feedback manipulator. <http://www.krafttelerobotics.com/products/raptor.htm>. [Online; accessed 27-June-2022].
- [83] E. Topini, A. Bucci, J. Gelli, A. Topini, A. Ridolfi, and B. Allotta. Advanced underwater manipulation systems: an overview of the suono project. In *OCEANS 2021: San Diego – Porto*, pages 1–7, 2021.
- [84] N. U-102:2020. Remotely operated vehicle (rov) services. Standard, NOR-SOK, Nov. 2020.
- [85] V. Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control*, 22(2):212–222, 1977.
- [86] I. B. Utne, I. Schjøberg, and E. Roe. High reliability management and control operator risks in autonomous marine systems and operations. *Ocean Engineering*, 171:399 – 416, 2019.
- [87] I. B. Utne, A. J. Sørensen, and I. Schjøberg. Risk Management of Autonomous Marine Systems and Operations. volume Volume 3B: Structures, Safety and Reliability of *International Conference on Offshore Mechanics and Arctic Engineering*, 06 2017.
- [88] I. Wampler, C. W. and L. J. Leifer. Applications of Damped Least-Squares Methods to Resolved-Rate and Resolved-Acceleration Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 110(1):31–38, 03 1988.
- [89] H. Wang, Y. Li, Y. Yang, S. Hu, B. Chen, and W. Gao. Study of artificial neural network on explosive detection with pftna method. In *IEEE Nuclear Science Symposium Conference Record, 2005*, volume 1, pages 471–473, Oct 2005.
- [90] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric, 2017.

- [91] Q. Xi, T. Rauschenbach, and L. Daoliang. Review of underwater machine vision technology and its applications. *Marine Technology Society Journal*, 51(1):75–97, 2017.
- [92] J. Xu and H.-S. Yoon. Vision-based estimation of excavator manipulator pose for automated grading control. *Automation in Construction*, 98:122 – 131, 2019.
- [93] L. Xue and A. M. Lekkas. Comparison of ai planning frameworks for underwater intervention drones. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pages 1–9, 2020.
- [94] X. Zhao, H. Jia, and Y. Ni. A novel three-dimensional object detection with the modified you only look once method. *International Journal of Advanced Robotic Systems*, 15:172988141876550, 03 2018.
- [95] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

Part II

Collection of Articles

Article 1

Transfer Learning in Underwater Operations

Martin Skaldebø, Albert Sans-Muntadas, Ingrid Schjølberg
Oceans, 2019, Marseille

doi: [10.1109/OCEANSE.2019.8867288](https://doi.org/10.1109/OCEANSE.2019.8867288)

In reference to IEEE copyrighted material, which is used with permission in this thesis, the IEEE does not endorse any of NTNU's products or services, Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to <https://www.ieee.org/publications/rights/rights-link.html> to learn how to obtain a License from RightsLink

Transfer Learning in Underwater Operations

Martin Skaldebø
Dept. of Marine Technology
NTNU
Trondheim, Norway
martin.b.skaldebo@ntnu.no

Albert Sans Muntadas
Dept. of Marine Technology
NTNU
Trondheim, Norway
albert.sans@ntnu.no

Ingrid Schjølberg
Dept. of Marine Technology
NTNU
Trondheim, Norway
ingrid.schjolberg@ntnu.no

Abstract—This paper investigates a method for reducing the reality gap that occurs when applying simulated data in training for vision-based operations in a subsea environment. The distinction in knowledge in the simulated and real domains is denoted the *reality gap*. The objective of the presented work is to adapt and test a method for transferring knowledge obtained in a simulated environment into the real environment. The main method in focus is the machine learning framework CycleGAN, mapping desired features in order to recreate environments. The overall goal is to enable a framework trained in a simulated environment to recognize the desired features when applied in the real world. The performance of the learning transfer is measured by the ability to recreate the different environments from new test data. The obtained results demonstrates that the CycleGAN framework is able to map features characteristic for an underwater environment presented with the unlabeled datasets. Evaluation metrics, such as Average precision (AP) or FCN-score can be used to further evaluate the results. Moreover, this requires labeled data, which provides additional development of the current datasets.

Index Terms—Underwater robotics, transfer learning, autonomy, CycleGAN

I. INTRODUCTION

Today, underwater operations experience a shift towards use of more autonomous systems, where machine learning is believed to play a central role. Especially, regarding the ability to transfer knowledge between operator and system. Human brains are experts at knowledge transfer. This might be perceived as a basic trait of the human intelligence, but is in fact extremely complicated to establish as a computational ability. The main idea is to enable machines to transfer knowledge between different domains and execute different related tasks. An overall goal is to be able to train in a simulated domain and then execute the same tasks in the real world. Regarding underwater operations the latter one is of particular interest as the deep sea is less accessible, operations are costly and challenging.

Training machines in an underwater environment is extremely time consuming and error-prone due to the harsh environment. Moreover, if machines are trained exclusively in simulations the transfer of knowledge to the real world could also generate failure. This is referred to as the reality gap [1]. Generating robust techniques for transferring the knowledge between domains is therefore of immediate interest for operators in this market. There exist several published methods dealing with this problem, however only for specific

domains. This paper will investigate one such method for the use in the underwater domain. One of the most promising frameworks, CycleGAN, will be tested on two different datasets considering underwater environments. The datasets include real and rendered vision based pictures of subsea panels.

A. Motivation

The underwater robotic market size is claimed to reach USD 6.74 Billion by 2025 [2]. This corresponds to a Compound Annual Growth Rate (CAGR) of 13.5%. By comparison, Apple Inc.'s 5 year CAGR is, per April 2019, 9.2% [3]. The same report predicts that autonomous underwater vehicles (AUV) will account for USD 1.48 billion by 2025. The Norwegian Government is investing in the ocean space when designing the concept *Ocean Space Center*. The concept has a planned investment of 4.7 billion NOK [4].

Activity, interest and economic growth within the ocean space is in other words unquestionable, and with growth advancement in the technology is forthcoming. In the last years, machine learning has experienced a substantial growth in both media coverage and technological applications. One specific area is within vision-based navigation for autonomous systems. The wide interest and willingness to achieve progress that is shown today generates motivation for further investments in the field. Machine learning is believed to play a significant role in the shift towards autonomy.

B. Background

Underwater operations today are highly dependant on human operators. Operations previously executed by human divers are now mostly transferred to remotely operated vehicles (ROVs). Moreover, the industry is today experiencing a new shift towards more autonomous operations where ROVs becomes more independent of human operators. Increasing the level of autonomy and optimize the human-robot interaction in these operations can potentially reduce costs and increase safety [5]. A higher level of autonomy leads to new requirements and increasing the autonomous complexity. Moreover, autonomous underwater vehicles (AUVs) require higher level of autonomy than ROVs. Since global navigation satellite system (GNSS) measurements are not applicable underwater, vehicle operation in this domain lacks localization measurements and are prone to accumulation of error. Today,

the most common measurements and signal data arrives from acoustic sensors. Such signals are prone to data loss due to transmission losses, acoustic noise in thrusters, signal reflections on different surfaces, absorption loss and more. Feature extraction using camera vision are rarely used, but improvements within artificial neural networks (ANN), especially convolutional neural networks (CNN), shows promising results. In the presented work, systems using visual aided navigation will be investigated. This is mostly motivated by the rapid advance withing CNN and other computer vision frameworks building on CNN.

Although it is in the last few years CNN has been given recognition for its good results, it can be traced back to 1980 and Neocognitron [6]. He proposed a hierarchical multilayered neural network performing robust visual pattern recognition. Such networks can be defined as

“Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.” [7]

A neural network can be defined as a computer program that is inspired by the natural neural networks in the human brain [8]. Such artificial neural networks are designed to perform cognitive functions as problem solving and machine learning. Neural networks have successfully been implemented in games [9], handwriting recognition [10] and even explosive detection [11]. Neural networks provides a method for defining a system too complex to be defined by a simple model, e.g. image recognition and other systems influenced by uncertainty.

A really important parameter concerning the overall capability of the neural network is it's architecture. The architecture concerns number of layers, number of neurons in each layer, connections between neurons etc. A nematode worm possesses only 302 neurons in total [12]. Still this presumably unintelligent worm is capable of performing complex tasks super computers today have troubles with. This is due to the complexity of the yet unknown inner mechanisms and architecture of a worm's biological neural network. As stated before, ANNs are inspired from the biological networks in human brains. However, the extremely complex brain is still not fully understood even by scientists who have devoted large part of their professional life investigating the human brain. ANNs architecture are therefore just a mere sketch of the complex biological version. Still, through the 4th industrial revolution we are experiencing today, new methods, algorithms and frameworks emerge rapidly [13].

Machine learning applications have achieved state-of-the-art performances in multiple disciplines using ANN. Google's *AlphaGo* has beaten the worlds best human Go player, and is arguably the strongest Go player in history [9]. *InnerEye* by Microsoft uses machine learning to develop image diagnostic tools in order to detect tumors etc. [14]. Machine learning approaches are also believed to have a dramatic impact in the fields of economics [15] in the short future. Thus, it is safe to say that machine learning will, at some extent, impact the majority of the modern generation.

C. Contributions

This paper investigates a method for transfer learning in underwater domains. Existing methods have not to a large extent been tested for use in underwater domains. In the presented work, experiments are conducted for two different datasets obtained in an underwater environment. Large datasets required for machine learning applications can be expensive and difficult to acquire. Applying transfer learning methods for underwater environments can provide an alternative method for cost-effective and simple dataset generation. This paper provides a collective overview of state-of-the-art frameworks targeting transfer learning topics. Moreover, suggests solutions for reduction of the reality gap in the learning process of machines. The main contribution of the work is the application of a transfer learning framework to vision-based underwater operations.

The outline of the paper follows with Sec. II describing investigated methods involving transfer learning. Sec. III describes the experiment setup and datasets as well as conducted simulations, before the results are presented and discussed in Sec. IV. Lastly conclusions and recommendations regarding further work are presented in Sec. V.

II. RELATED WORK

Transfer learning is a substantial problem in machine learning. A robotic arm can be trained to sort red and yellow cubes. However, such training algorithms often run into problems if the color of the cubes change to blue and green. Or, if the shape changes to triangles, or simply the lightning setting changes. Algorithms trained in a simulated environment often experience a problem when they are applied to real world data. This is referred to as the *reality gap*. Different approaches have been developed to reduce this gap between a simulated environment and the real world. A suggested solution is to train on a variation of simulated environment data. [16] developed an object detector that trained using only simulated data. The paper focused on a robotic arm that would grasp desired objects in a cluttered environment. They found it possible to train the detector to 1.5cm accuracy. The simulator they utilized consisted of randomly rendered images with variation in camera position, lighting conditions, object positions and non-realistic-textures. The objective was to perceive the real environment as just another variation. They demonstrated how their object detector could achieve high enough accuracy when tested in real life even though it only had been trained on in a simulated environment.

A breakthrough within the transfer learning topic arguably came in 2014 when Generative Adversarial Networks (GAN) was introduced [7]. The network consist of a combination of two networks, a generator and a discriminator. The generator aims to produce content, while the discriminator determines the level of authenticity of the content. They learn simultaneously and compete against each other, in what can be described as a zero-sum game. The generator produces samples $x = G(z)$, and the discriminator attempts to determine if the samples are produced by the generator or if they come directly from the

training set. The discriminator produces a probability given by $D(x)$, indicating the probability that x is a real sample rather than a fake sample produced by the generator. The end-goal of GAN is that the discriminator will be unable to distinguish the real samples from the fake and produce a constant probability of 0.5. The discriminator will focus on learning to correctly classify samples as real or fake, while the generator will simultaneously try to generate as real looking samples as possible to fool the discriminator. This model can be highly under-constrained, but there exist several published methods and frameworks solving this.

Coupled Generative Adversarial Network (CoGAN) is a framework for learning joint distributions between individual domains [17]. The model aims to obtain a learning based on the joint distributions between domains rather than learning from corresponding images in different domains. This simplifies the requirements of the datasets, because CoGAN doesn't require corresponding images in the different domains. The framework discovers the joint distribution instead. CoGAN has been applied for color and depth images, as well as on face images with different attributes and demonstrated successfully image transformations between domains.

Based on the CoGAN framework, [18] illustrates a method for unsupervised image-to-image translation. The method learns a joint distribution between individual domains, by assuming there exists a shared-latent space. The shared-latent space assumption assumes a pair of corresponding images in different domains can be mapped in the same latent domain. The authors demonstrated image-to-image translation between two domains without any corresponding images in the training datasets. Moreover, a limitation of the presented translation is a unimodal model due to the Gaussian latent space assumption. A unimodal model means there exist only one peak, i.e. one right answer. Another limitation is possible unstable training due to the saddle point searching problem.

pix2pix uses conditional GAN (cGAN) to learn the translation between domains [19]. Since the release of the framework, a large number of different experiments has been conducted by different people. The framework shows promising results. The downside of pix2pix is the need for correlating image pairs in the source and target domain. A modified version of GAN, CycleGAN, is a method to perform image-to-image translation between domains without paired images in each domain [20]. The independence from paired images as well as wide range of domains CycleGAN has been applied to, are the main reasons why CycleGAN is the contemplated framework for this paper.

A. CycleGAN

A thorough description of the CycleGAN framework can be found in [20]. Moreover, an overall description of the framework and how the cycle consistency is implemented in the framework is summarized here. The image-to-image translation is achieved by adding an additional generator and discriminator. The framework attempts to learn the mapping $y = G(x)$ and $x \approx F(G(x))$, where G and F are two different generators. CycleGAN is one of the recent most successful

approaches to the image domain transformation topic. Introducing $x \approx F(G(x))$ provides an additional loss function, the cycle consistency loss, in addition to the adversarial loss. The adversarial loss is defined with

$$L_{GAN}(G, D_Y, X, Y) = \mathbf{E}_{y \sim p_{data}} \log D_Y(y) + \mathbf{E}_{x \sim p_{data}} \log (1 - D_Y(G(x))), \quad (1)$$

where G is the mapping function attempting to generate images $G(x)$ similar to images in domain Y . D_Y attempts to distinguish between the generated images, $G(x)$, and the real images y .

In order to implement a desired cycle consistent mapping, the cycle consistency loss is added, (2). This loss ensures that for each image, x or y , the original image is reconstructed after the image translation cycle, i.e. $X \approx F(G(X))$ and $Y \approx G(F(Y))$, as previously mentioned.

$$L_{cyc}(G, F) = \mathbf{E}_{x \sim p_{data}} \|F(G(x)) - x\|_1 + \mathbf{E}_{y \sim p_{data}} \|G(F(y)) - y\|_1 \quad (2)$$

The objective in CycleGAN will consequently be a sum of the adversarial loss and the cycle consistency loss, represented with the final loss function

$$L_{CycleGAN}(G, F, D_x, D_y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(G, D_X, Y, X) + \lambda L_{cyc}(G, F). \quad (3)$$

λ determines the relative importance of the two objectives. Notice that the final loss function is represented with two functions for adversarial loss. This is to ensure the losses for mapping between both domains are accounted for. Considering the loss function given by (3), the objective of CycleGAN will be to solve

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} L_{CycleGAN}(G, F, D_x, D_y). \quad (4)$$

As mentioned, CycleGAN offers unpaired image-to-image translation. Regarding datasets, this provide a great advantage, because datasets can be extracted from already existing data in the industry. The framework can also provide the translation with unlabeled dataset, which means time spent on labeling each element in vast amounts of data can then be avoided.

III. EXPERIMENTAL SETUP

In this section the two contemplated datasets will be introduced. Parameters regarding the training and testing of the CycleGAN framework will also be presented.

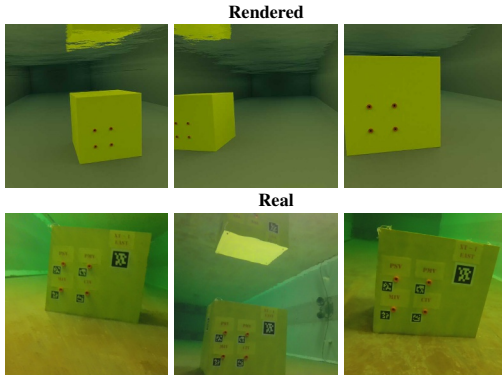


Fig. 1: Dataset 1.

A. Dataset

The datasets that will be used for simulations are two sets containing real and rendered images of a subsea panel. Subsea panels are installed on oil and gas templates on the Norwegian Continental Shelf. The panels are accessed by ROVs for e.g. valve operations and the ROVs operators are totally dependent of good images. In case of autonomous valve operations, automatic systems based on machine learning techniques and the CycleGAN framework is one solution for image characterization. The datasets contains no corresponding images in the training sets, meaning there exist no specific image for one domain corresponding to another image in the other domain. The datasets are also unlabeled. The framework is therefore required to map the features between the domains without being told the correspondence between them. Dataset 1 contains images of a subsea panel placed in the marine cybernetics laboratory (MC-lab) at NTNU [21], as well as rendered images of the same environment. This dataset contains four different directories.

- **trainA**: Containing 4868 rendered .jpg images of the subsea panel at the bottom of the MC-lab.
- **trainB**: Containing 2947 .jpg real images of the subsea panel at the bottom of the MC-lab
- **testA**: Containing 132 rendered .jpg images of the subsea panel at the bottom of the MC-lab
- **testB**: Containing 118 .jpg real images of the subsea panel at the bottom of the MC-lab

The images are taken from a videostream filming the subsea panel at different angles, while the rendered images are rendered using the software blender. Fig. 1 represents image examples taken from the dataset.

Dataset 2 contains real and rendered images of a subsea panel placed in the fjord outside Trondheim. These are images taken at a more realistic setting, which naturally contains more noise than the images from the laboratory. The rendered images are taken from a computer aided design (CAD) model

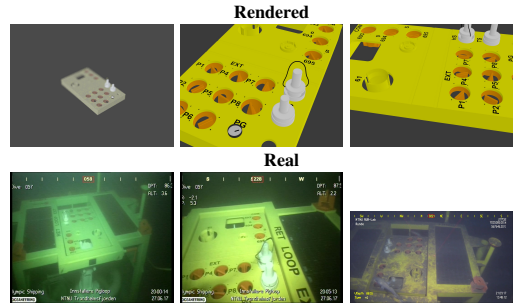


Fig. 2: Dataset 2.

where angles, distance and different noise patterns are altered to ensure the dataset contains variance. The images can be seen in Fig. 2. The dataset is split into 4 directories with

- **trainA**: Containing 1786 rendered .jpg images of the subsea panel.
- **trainB**: Containing 406 .jpg real images of the subsea panel at the bottom the fjord.
- **testA**: Containing 200 rendered .jpg images of the subsea panel.
- **testB**: Containing 46 .jpg real images of the subsea panel at the bottom of the fjord.

Both datasets represents an underwater environment, however at different extent. Dataset 1 is from a laboratory and the images are characterized by clear water and light conditions, not unlike a surface environment. Dataset 2 are more characterized by a typical underwater environment. The environment is dark, reflection from light source occurs and fouling are present at a representative amount of the images. Another reoccurring issue in underwater environments is marine snow which leads to occluded images. Moreover, the images in this dataset are taken from inside a fjord, which results in marine snow being almost non-existent with relative clear images due to calm water.

B. Framework

Dataset 1 discussed above includes images with size 256×256 . When training on this network, both load size and crop size of the framework are set to 256×256 in order to maintain the resolution of the input images. Dataset 2 includes images with different image sizes. The rendered images includes images in sizes 1080×980 and 1080×800 . The real images varies between 1920×1080 and 720×576 . When training on this dataset, the load size is set to 286×286 and crop size 256×256 . The different sizes of input images are therefore coped with by loading all images into the framework with equal size before they are cropped.

The model trains for 200 epochs, which represents going through the entire dataset 200 times. The training is conducted with a constant learning rate of 0.0002 for the first 100 epochs, before decaying towards 0 for the last 100 epochs. For all

simulations, λ from equation 3 is set to $\lambda = 10$. The model is saved every 3000 iterations. In order to keep track of the progress during training, examples of the current state are generated every time the model is saved. The discriminator network architectures are 70×70 PatchGAN networks. The regular GAN discriminator maps from a 256×256 input to a scalar output to determine real or fake. In comparison, the PatchGAN discriminator maps from a 256×256 input to a $N \times N$ network of X outputs, where X_{ij} signifies whether the patch ij in the input image is real or fake. The architecture of the generator networks contains two 2-stride convolutions, nine residual blocks and two fractionally-strided convolutions with stride $1/2$. Further, the framework is trained with a batch size of 3 with batch normalization. The batch normalization ensures that the loss is calculated over the batch and not for each instance. The framework is trained on a *Nvidia GeForce RTX 2080 Ti/PCIe/SSE2* graphics card.

IV. RESULTS AND DISCUSSION

In this section the results from the simulations will be presented. Both datasets have been tested on transfer learning between the two domains and the results vary. The varying results will also be discussed and suggested improvements and solutions will be presented.

A. Results

The results are obtained by testing the framework on the test directories with the trained weights. A part of the obtained results are depicted in Fig. 3. The figures includes six original input images from the rendered domain and the corresponding generated output. For both datasets. Testing has been conducted between both domains, in order to see if the framework has correctly mapped the relevant features in the two domains. However, regarding the task of generating datasets for future machine learning applications, the results presented in Fig. 3 would be the most interesting. Proper generation of images in the real world domain enables generation of vast datasets from rendered images. If large and decent datasets are hard to obtain, this method can provide a more cost-effective alternative.

The results are most satisfying for dataset 2. It can be seen from the figures that for dataset 2, the framework manages to transfer the subsea panel into the other domain in a good manner. The overall structure from the input is kept while the domain changes towards the real domain. Regarding dataset 1 the results are not as satisfying. The output drastically deviates from the input. The relative angle between the camera and structure as well as the spatial features are changed in the output relative to the input. The details on the subsea panel also seems to be randomly placed on the panel. This suggests that the mapping between the domains has been unsuccessful. The domains possess some different features, e.g. QR-codes are neglected in the rendered domain. The framework might encounter issues mapping features that is simply non-existent in one of the two domains.

Fig. 4 also illustrates an insufficient mapping between the domains. For both datasets. This figure depicts the results of applying the domain transfer on the real domain and transferring the input images to the rendered domain. Dataset 1 demonstrates the same issues as for the opposite domain transfer, where angles and spatial features of the input images are changed in the domain transfer. This strengthens the theory that the features of the domains has not been properly mapped due to the low level of details in the rendered images. The domain transfer for dataset 2 seems to encounter much of the same issues. The input images includes parts of the structure that are not included in the rendered domain. The additional structure circumventing the subsea panel as well as the robotic manipulator in the images are unknown to the rendered domain. Consequently, the framework have problems transferring these features into the rendered domain where they are completely absent.

As previously stated, CycleGAN compares the original image to a reconstructed image in order to calculate the cycle consistency loss. This is depicted in Fig. 5. The figure depicts the input image fed to the framework and the output is the corresponding image in the other domain. The reconstructed image is generated by taking the output image as input and then transferred back to the first domain. The reconstructed images represents the input very well. Notice that for the second image line, the four orange dots are almost gone in the reconstructed image. This demonstrates that the framework perceives these features as non-important. The orange dots are the only features on the rendered subsea panel with information about where the QR-codes should be placed. If these features are seen as non-important it could explain why the generated subsea panels from Fig. 3a are often flipped.

B. Discussion

The results are promising and illustrates a decent mapping between the domains, especially for dataset 2. However, less satisfactory results were also obtained, which indicates that the feature mapping may not be as robust as desired. It should be noted that the level of details on the CAD models could be a reason. The CAD model in dataset 1 illustrates a yellow box with four orange dots. The different QR-codes that are present on the real model are neglected in the CAD model. This may confuse the framework when it attempts to map these exact features between the two domains. This might also be a reason why the generated images of the subsea panel often is flipped for this dataset. The largest QR-code are placed in the upper right corner of the real model. See Fig. 1. However, for the constructed images, it seems randomly placed in either of the upper corners. Placing QR-codes on the rendered model could help ensure that the placement is perceived as a more important feature to the framework. This dataset is obtained in the MC-Lab at NTNU, and the images are not characterized by the dark underwater environment. It is therefore believed that the issues of mapping the features between the domains, is due to the lack of details in the CAD model rather than the fact that the domains are underwater. Moreover, we do

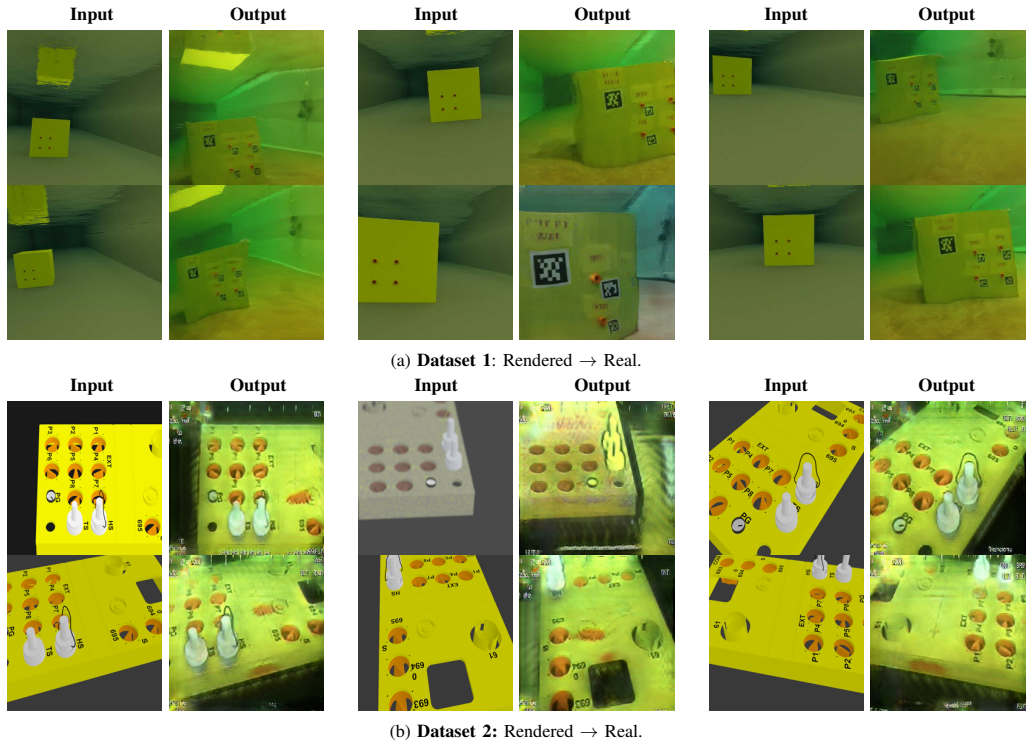


Fig. 3: Results from generating images in the real domain with rendered images as input.

not know exactly how the features are mapped in the neural network, which makes it difficult to determine how much such changes would improve the framework. It is also unknown if they would improve the results at all.

The level of details are increased for dataset 2. However, even though the CAD model is relatively detailed, it only includes the panel itself. The real images shows a panel placed on a larger subsea structure with a manipulator often occurring in the images as well. The additional subsea structure around the panel is non-existent in the CAD-model, which causes some mapping issues. Constructing images of the panel from a distance cause varying results due to this missing structure in the CAD-model. When constructing close up images of the panel on the other hand, the framework performs very well. On the close up images, the level of details are quite similar for the CAD-model and the real images. This provides good circumstances for the framework to map the features between the domains. This dataset is also much more characterized by an underwater environment. When the panel is seen from a distance it is perceived blurry, and the lighting becomes a strong feature. Due to the dark environment, a source of light is necessary in order to light up the subsea panel. Fouling on

the structure, distance of the camera, occluding of camera or light source an other factors provides different reflections on the structure and provides a challenging domain to map. Still, the framework is able to represent this light reflection in a good manner.

Overall, the framework performs well. Limitations that occurs in the domain transfers are believed to arrive from different features not being present in both domains, rather than features characterized by underwater environments. Since the framework is able to comprehend with such circumstantial features, the obtained limitations should be possible to improve similarly to surface limitations.

A limitation with the dataset is the absent of marine snow in the images. The environments on the Norwegian Continental Shelf are deeper, darker and more demanding than the datasets presented in this paper. Another limitation is that the results are hard to evaluate with a metric, due to the dataset being unlabeled. A labeled dataset provides a ground truth, which the results can be compared to. Two popular evaluation metrics for the results presented in this paper are the Average precision (AP), which is often used when measuring accuracy of classifiers [22], and the FCN-score used in [19].

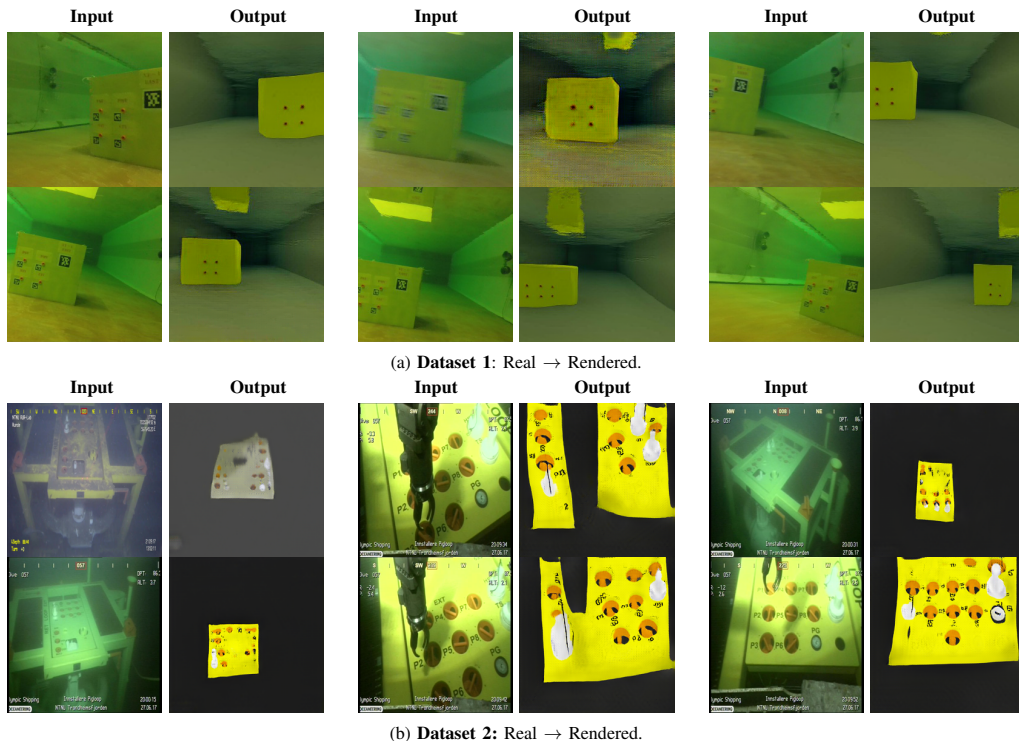


Fig. 4: Results from generating images in the rendered domain with real images as input.

V. CONCLUSIONS AND FUTURE WORK

This paper investigate methods for reducing the reality gap for vision based systems in the underwater segment. Simulations have been conducted on two different underwater datasets in order to apply existing methods at underwater environments. CycleGAN has been used as the contemplated framework. The datasets consist of rendered and real images of two different subsea panels. The framework was trained for 200 epochs on the two different datasets and the results demonstrated a partially successful mapping between the domains. Some results were satisfactory, but less satisfactory results revealed a less robust feature mapping. The framework proved to be able to map features characterized by underwater environments, such as dark images and light reflection. It is therefore believed that increasing the level of details on the CAD models could provide a solution for increasing the robustness of the feature mapping. Moreover, using labeled datasets can also provide possibilities for using evaluation metrics. These issues should be addressed in future work.

ACKNOWLEDGMENTS

Special thanks to MSc. student Daniel Harper for providing the CAD model utilized in dataset 2. Also thanks to MSC. student Fabian Skjølvsvik for editing the images in dataset 2 and setting it up as a functioning dataset.

REFERENCES

- [1] K. Bousmalis and S. Levine, "Closing the simulation-to-reality gap for deep robotic learning." Available at <https://ai.googleblog.com/2017/10/closing-simulation-to-reality-gap-for.html>, October 2017.
- [2] ResearchAndMarkets.com, *Underwater Robotics Market Size, Share Trends Analysis Report By Type (ROV, AUV), By Application (Commercial Exploration, Defense Security, Scientific Research), By Region, And Segment Forecasts, 2018 - 2025*. ResearchAndMarkets.com, 2018.
- [3] finbox.io, "Revenue cagr (5y) for apple inc." Available at https://finbox.io/AAPL/explorer/revenue_cagr_5y, November 2018.
- [4] sintef.no, "Ocean space centre får grønt lys fra kvalitetssikrer." Available at <http://www.oceanspacecentre.no/wp-content/uploads/2018/04/Supplerende-analyse-Ocean-Space-Centre-forkortet.pdf>, October 2018.
- [5] I. Schjølberg, T. B. Gjersvik, A. A. Transteth, and I. B. Utne, "Next generation subsea inspection, maintenance and repair operations," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 434 – 439, 2016, 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016.

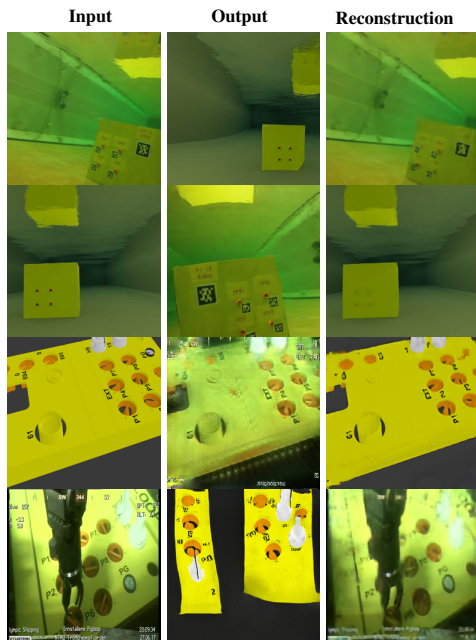


Fig. 5: Input and output with reconstructed image.

ulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 23–30.

[17] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 469–477.

[18] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *CoRR*, vol. abs/1703.00848, 2017.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arxiv*, 2016.

[20] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR*, vol. abs/1703.10593, 2017.

[21] "Marine cybernetics teaching laboratory," Available at <https://www.ntnu.edu/imt/lab/cybernetics>.

[22] J. Hui, "map (mean average precision) for object detection," Available at https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173, March 2018.

[6] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, April 1980.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.

[8] V. Zwass, "Neral network," Available at <https://academic.eb.com/levels/collegiate/article/neural-network/126495>, November 2018.

[9] "AlphaGo," Available at <https://deepmind.com/research/alphago/>, November 2018.

[10] R. Almodfer, S. Xiong, M. Mudhsh, and P. Duan, "Multi-column deep neural network for offline arabic handwriting recognition," in *Artificial Neural Networks and Machine Learning – ICANN 2017*, A. Lintas, S. Rovetta, P. F. Verschure, and A. E. Villa, Eds. Cham: Springer International Publishing, 2017, pp. 260–267.

[11] H. Wang, Y. Li, Y. Yang, S. Hu, B. Chen, and W. Gao, "Study of artificial neural network on explosive detection with pftna method," in *IEEE Nuclear Science Symposium Conference Record, 2005*, vol. 1, Oct 2005, pp. 471–473.

[12] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode *Caenorhabditis elegans*," *Philosophical transactions of the Royal Society of London*, vol. 314, no. 1165, pp. 1–340, 1986.

[13] B. Marr, "What everyone must know about industry 4.0," Available at <https://www.forbes.com/sites/bernardmarr/2016/06/20/what-everyone-must-know-about-industry-4-0/2941b856795f>, June 2016.

[14] "Project innereye – medical imaging ai to empower clinicians," Available at <https://www.microsoft.com/en-us/research/project/medical-image-analysis/>, 2018.

[15] S. Athey, "The impact of machine learning on economics," Available at <https://www.nber.org/chapters/c14009.pdf>, January 2018.

[16] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from sim-

Article 2

Dynamic Positioning of an Underwater Vehicle using Monocular Vision-Based Object Detection with Machine Learning

Martin Skaldebø, Bent O. A. Haugaløkken, Ingrid Schjøberg
Oceans, Seattle, 2019.

doi: [10.23919/OCEANS40490.2019.8962583](https://doi.org/10.23919/OCEANS40490.2019.8962583)

In reference to IEEE copyrighted material, which is used with permission in this thesis, the IEEE does not endorse any of NTNU's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to <https://www.ieee.org/publications/rights/rights-link.html> to learn how to obtain a License from RightsLink

Dynamic Positioning of an Underwater Vehicle using Monocular Vision-Based Object Detection with Machine Learning

Martin Skaldebø
Dept. of Marine Technology
NTNU
Trondheim, Norway
martin.b.skaldebo@ntnu.no

Bent Oddvar Arnesen Haugaløkken
Dept. of Marine Technology
NTNU
Trondheim, Norway
bent.o.arnesen@ntnu.no

Ingrid Schjølberg
Dept. of Marine Technology
NTNU
Trondheim, Norway
ingrid.schjolberg@ntnu.no

Abstract—Sonar and camera are two widely used sensors in the underwater segment. Moreover, optical based systems provide higher spatial and temporal resolution than their acoustic counterpart. In this paper, a dynamic positioning system for a small-class ROV relative to an object detected by a monocular camera will be presented. The object detection will be performed by the state-of-the-art object detector YOLOv3 trained on a dataset representing the relevant known object. In order to perform dynamic positioning based on 2D images, a scaling will be used to extract the spatial features from the images. The entire system is able to perform in real-time, which is essential for a dynamic positioning procedure.

Index Terms—Underwater robotics, object detection, autonomy, data augmentation, dynamic positioning

I. INTRODUCTION

Subsea inspection, maintenance and repair (IMR) operations are often identified with the offshore oil and gas industry, but lately also highly relevant for deep sea mining and aquaculture. IMR operations are commonly executed by underwater vehicle-manipulator systems (UVMS), which today rely heavily on humans. Increasing the level of autonomy and optimizing the human-robot interaction in these operations can potentially reduce costs and increase safety [1].

In teleoperated systems, the human operator is aided by visual and sensory feedback in order to assess the situation, make decisions and remotely execute tasks. The same conditions apply for autonomous systems, only then the system itself needs to conduct the operations without a human in the loop, increasing the demands regarding the sensory systems and implemented software. In the object detection aspect of autonomous operations, sonars and cameras are two widely used sensors [2]–[4]. Acoustic sonars have for a long time been a preferable sensor in underwater systems. However, recent technological advance within camera systems and the use of visual aid proves that camera systems have potential for short range navigation. Moreover, visual aided systems may provide systems with higher spatial and temporal resolutions than the acoustic counterpart [5].

It is not straightforward to use camera systems in underwater environments, especially when paired with robotic

systems during semi- or fully autonomous operations. The underwater scene is one of the most difficult environments to perform optical detection and recognition of objects and patterns. This is because underwater image quality heavily depends on absorption and scattering of light [6] [7]. With regards to the operation at hand, underwater object detection is typically performed with a specific object in mind, which might be fully visible, or either partially or fully obscured by other objects. Recent technological advances and lowered costs of graphical processing units and cameras have made object detection possible both quickly and reliably, hence making the method suitable in conjunction with autonomous control applications. Simultaneously, the same type of development has been seen in commercial underwater vehicle products, such as the BlueROV2, which allows customization and testing of new hardware with user-made software. Incorporating object detection methodologies for underwater vehicles enables more autonomous functionality in underwater robotics, such as tracking of objects during IMR or visually aided manipulation operations, whether it is used in exploration operations, within the marine oil and gas or the aquaculture industry.

This paper will investigate a method for efficient labelling of large datasets and training a model based on the state-of-the-art object detection framework YOLOv3 [8]. The object detection will be used to locate a known object with a monocular camera mounted in a BlueROV2 underwater vehicle. Moreover, spatial features will be extracted from the object detection in order to estimate the relative distance between object and vehicle. A dynamic positioning (DP) system is designed to keep the vehicle at a desired position and orientation relative to the object. The main contributions of this paper are as follows:

- Collect a large dataset of the object and provide underwater object detection using machine learning
- A method for efficiently labeling of large datasets
- Extraction of spatial information from monocular camera underwater images
- A DP-system where the vehicle has a desired position and orientation relative to a known object, where position of

the object is extracted from monocular camera

- Demonstration of the proposed methods and concepts in a laboratory pool

The paper is structured as follows. Section II provides some background information on visually aided control using underwater vehicles and presents related work on object detection topics. Section III presents the dataset and how the labeling and scaling procedures are conducted as well as how the detector works. Furthermore, Section IV describes the motion control system, while the experimental setup is explained along with results in Section V. Lastly, the results are discussed in Section VI and conclusions and suggestions for further work are provided in Section VII.

II. RELATED WORK

Optic-based underwater object detection is a hot topic today, especially within the research community and various subsea industries, but also for hobbyists, where a consistent series of new contributions have surfaced in the last few years. The most common vision-based techniques utilize either monocular (2D) or stereo (3D) vision, while 2.5D methods also have been proposed, mainly consisting of projection algorithms from the 2D image plane to reconstruct 3D environment features [6]. Detection methods range from on edge [9] [10] or color [11] detection, optical flow [12], and techniques relying more on machine learning approaches such as classification [13], salient feature detection [14] or object detection [15] [16]. Color detection simply finds and draws contours around neighboring colored pixels in an image, while edge detection denotes the boundaries of objects. Optical flow is used to track individual pixels or pixel areas, and can be used in combination with the aforementioned methods.

Within machine learning, there is a wide array of various detection methodologies, mainly identified as two different types. These are the region proposals based detectors and the regression and classification based detectors. They both originate from generic object detection as illustrated in Fig. 1. The first follows the traditional object detection procedure, identifying region proposals and classifying the proposals into object categories, also known as Region Proposals Networks (RPN). Such methods consider, at some extent, the same methodologies as the human brain, and are based on an initial scan of the entire scenario before the scanned image is separated into regions of specific interest. The second type follows a classifier-based approach or a regression problem. These methodologies both arrive from the supervised learning branch of machine learning. The branches of machine learning are shown in Fig. 2. Thus, they share the overall objective of supervised learning, which is to learn the mapping from input x to output y , i.e. learn the mapping function $f(\cdot)$ in $y = f(x)$. The main difference is that while classification approaches aim to learn the mapping to a discrete or categorical output, regression approaches aim to learn the mapping to a continuous or numerical output.

A collected review of the most essential methods of both RPN and regression/classification based approaches can be

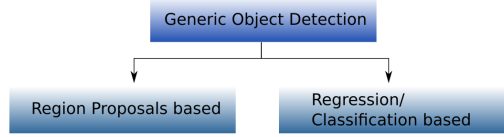


Fig. 1: Object Detection Methodologies

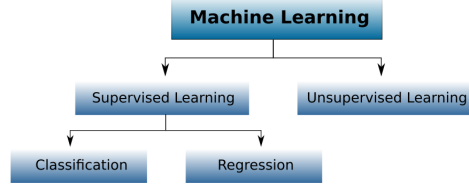


Fig. 2: Machine Learning methodologies

seen in [17]. Some methods worth mentioning are the RPNs Region proposals CNN (R-CNN), Fast R-CNN and Faster R-CNN and the regression/classification based methods Single Shot Detector (SSD) and You Only Look Once (YOLO) versions 1, 2 and 3.

R-CNN is region proposals combined with convolutional neural networks, hence the name. This algorithm proposes 2000 regions per image. The algorithm then works with these regions and attempts to classify them in order to locate the object. 2000 regions per image is a lot to process and is computationally expensive. The algorithm uses about 47 seconds per image, which makes it irrelevant regarding real-time detection. A modified version, Fast R-CNN, uses a slightly different approach. Instead of proposing 2000 regions per image, the modified version feeds the input image to a CNN and outputs a convolutional feature map. The proposed regions from the feature map is fed through a regions of interest (RoI) pooling layer and a fully connected layer. Then a softmax layer is used in order to predict the class of the object as well as the regions for the bounding boxes. As for R-CNN, Fast R-CNN uses selective search, which is a slow and time-consuming process. Faster R-CNN is a further modification of the algorithm to circumvent the selective search algorithm. Faster R-CNN uses a separate network instead of selective search in order to predict the region proposals. This results in a much faster network. Faster R-CNN is the only network out of these three that is fast enough to be applicable for real-time object detection [18]–[20].

SSD uses a classification/regression based approach and, hence, does not require object proposals, and encapsulates all computation in a single network [21]. Consequently, the algorithm is fast and suitable as a real-time object detector. SSD also proved to be more accurate than the YOLO versions available when SSD was first introduced in late 2015. However, new versions of YOLO has emerged since then. The

version YOLOv3 has proven to be just as accurate as SSD, however three times faster [8]. Still YOLOv3 is not as fast as its predecessor, where YOLOv2 could run on a Titan X at 45 frames per second (FPS) while YOLOv3 is limited to about 30 FPS. This is due to the increased complexity of the underlying network called Darknet. YOLOv2 used Darknet-19 consisting of a total of 30 layers, consisting of a 19-layer network with additional 11 layers for object detection. YOLOv3's network Darknet-53 contains 53 layers trained on ImageNet with 53 more layers for object detection. This gives a network with 106 layers. This increased complexity is the reason for decreased speed as well as increased accuracy. Darknet-53 incorporates some new important elements compared to Darknet-19, such as residual blocks, skip connections and upsampling. In standard neural networks, layers are connected strictly from layer to layer, but introducing residual blocks or skip connections enables layers 2-3 steps away to be directly connected. Upsampling resolve the task of bringing back the resolution to the resolution of the previous layer. See Table I for the Darknet-53 layers. Another issue YOLOv3 has addressed is the fact that previous versions have struggled with detecting small objects. In YOLOv3, predictions are made at three different scales, and after each detection, layers are upsampled. The upsampling helps the network learn fine-grained features, which are advantageous for detecting small objects.

As mentioned previously, pairing visual tracking methodologies of objects with underwater vehicles has a tremendous potential in autonomous underwater operations. One of the most famous and successful combinations of this sort was demonstrated by the underwater vehicle named SAUVIM [22]. The object detection phase, which they considered to be the most difficult part of the project, was threefold. Image sonar and DIDSON sonar were used at long- and mid

range detection, while for the actual manipulation tasks, video cameras were used in collaboration with ultrasonic motion trackers. Further development of the SAUVIM project was later presented by the TRIDENT and MARIS projects [23], [24]. The TRIDENT project demonstrated the first multi-purpose object search and recovery strategy. Similar to the TRIDENT project, the MARIS project stands out as one of the recent most promising projects regarding autonomous underwater manipulation. Compared to the TRIDENT project, the MARIS project improved the vision system. Both projects employed stereo cameras where the MARIS project improved the detection algorithm to cope with partial occlusions of the object.

Some of the other work related to vision-aided robotic control applications can be found in [25]–[29]. The focus reported by these articles are either solely related to underwater object detection, positioning using vehicles and manipulation of detected objects using manipulator arms, or a combination of these. A more in-depth review article on this topic is presented in [6], and recent developments in machine learning methodologies can be seen in [17].

III. DATA AUGMENTATION

In the presented work, the dataset applied represents a known object in an underwater environment in the Marine Cybernetics Laboratory (MC-Lab) at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU). The dataset includes images retrieved in two stages. In the first stage, the camera recorded a video of the object at close range with a monocular camera attached to a robotic manipulator. In the second stage, a remotely operated vehicle (ROV) was used to manually record the environment with the object from various angles. For both of these cases, a Raspberry Pi Camera V2.1 was used. Specifications regarding pixel size and field of view of the camera are given in Table II. The two recordings were split into image sets to generate the dataset.

A. Labeling

The collected dataset contains 7071 images. Such a vast dataset provides comprehensive work regarding labeling the images. Therefore, methods for efficiently labeling the images were investigated. A popular method for managing large datasets is crowdsourcing, where a task is distributed to numerous participants for analyses. A recurring challenge is the unknown reliability of the participants [30] [31]. Our labeling scheme enables safe labeling of a large dataset with over 7000 images in mere hours, where every label is verified by the user in order to ensure the liability of the labels. The labeling process is split in two steps.

- Initial labeling of images using color detection.
- Correcting of wrong/bad labels.

The considered object is characterized by a clear orange color. This made it possible to separate the relevant colors of the object in order to detect the object. The contemplated

TABLE I: Darknet-53

	Type	Filters	Size	Output
1x	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3 / 2	128 x 128
	Convolutional	32	1 x 1	128 x 128
	Convolutional	64	1 x 1	
	Residual			
	Convolutional	128	3 x 3 / 2	64 x 64
2x	Convolutional	64	1 x 1	64 x 64
	Convolutional	128	3 x 3	
	Residual			
	Convolutional	256	3 x 3 / 2	32 x 32
	Convolutional	128	1 x 1	32 x 32
	Convolutional	256	3 x 3	
8x	Residual			
	Convolutional	512	3 x 3 / 2	16 x 16
	Convolutional	256	1 x 1	16 x 16
	Convolutional	512	3 x 3	
	Residual			
	Convolutional	1024	3 x 3 / 2	8 x 8
4x	Convolutional	512	1 x 1	8 x 8
	Convolutional	1024	3 x 3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

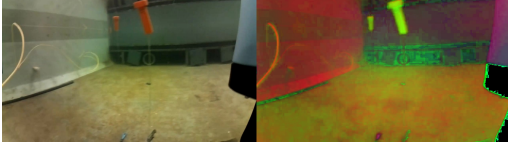


Fig. 3: RGB to HSV transformation of the underwater image

images were transformed from the RGB space to the hue-saturation-brightness value (HSV) in order to generate a more straightforward color map. Such a transformation can be seen in Fig. 3. After the initial label generation using the color scheme, the labels were regulated by the user operating an interactive interface. Utilizing HSV images ensures that the object becomes apparent in the image and simplifies the color separation. The values characterizing HSV images also provide a simpler spectrum to analyze compared to RGB values.

B. Spatial scaling

In object detection schemes, real-time pose of the object is generally of interest, thus stereo cameras are often used [23] [24] [32]. Stereo camera systems employ two or more lenses in order to simulate humans binocular vision, thus capturing spatial features in images to get a 3D representation. Moreover, in machine learning applications where labeled data is needed, 3D images could be complicated to process and labeling of such images are difficult. Processing standard monocular camera images and labeling ordinary RGB 2D images is a much more tractable problem, but requires alternative methods for extracting spatial features.

A scaling function is designed in consideration of the spatial features in the system. The spatial features are important in order to estimate the position of the object relative to the ROV. The scaling function involves scaling the area of the bounding boxes to the corresponding distance to the object. The scaling is performed by manually measuring different distances to the object and registering the area of the detection bounding box. The final scaling function is generated by using piecewise cubic hermite interpolation polynomials (PCHIP) at the registered values. A visual representation of the scaling is depicted in Fig. 4

The distance to the object corresponds to the x-position of the vehicle relative to the object. Values for (y,z) position of the object are also extracted as pixel values from the image frame, and refer to lateral and vertical position relative to the object, respectively. Furthermore the values are transformed into distances by (6) and (7), where the angles ψ and θ are retrieved from (3) and (4). The remaining variables in the equations are explained in Table II.

C. Object Detector

The object detector is based on the YOLOv3 algorithm discussed in Section II. The algorithm trains for 5000 iterations with a batch size of 64 and subdivision set to 16. An entire

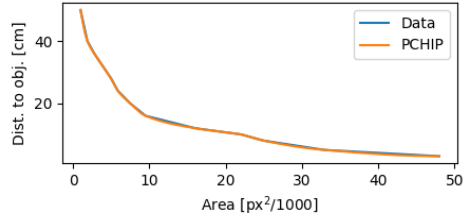


Fig. 4: Visual representation of the scaling function. Area in $\text{px}^2/1000$ is plotted against the corresponding distance to the object in cm. The original measured data is visualized with a blue line and the final scaling using PCHIP is visualized in orange.

TABLE II: Specifications of the Raspberry Pi Camera V2.1

Parameter	Definition	Value
FOV_w	Field of view in horizontal plane	62.6 [deg]
FOV_h	Field of view in vertical plane	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]

batch is considered between every update of the weights in the neural network. After training on the contemplated dataset of the object in the MC-lab, the object detector performs well and achieves an average precision (AP) of 97.9%, calculated with a threshold of 50%. The AP value is incredibly high and proves that the detection of the object is successful. However, the high AP has to be considered with some constraints. The algorithm was validated on a subset of the entire dataset, meaning the validation set embodies almost identical features as the training set. Consequently, a very high AP can potentially point to a overfitted model just as well as a good one.

The final updated weights from the training procedure are further used in the detection procedure. An algorithm is designed with an image frame as input and coordinates and size of the detected object as output. An outline of the detection algorithm is depicted in Fig 5, where (y, z) represents the pixel value of bounding box center position. The pixel origin is located in the top left corner where the pixel values are (0, 0). Numeric values for (w, h) represents width and height of the bounding box. All values are given in pixels. Total pixel dimensions of the camera frame can be found in Table II. Six parameters are defined in the table, representing horizontal and vertical field of view (FOV) of the camera, total pixel height and width of the camera images and the allowed pixel position of the object within the camera image. The algorithm assumes that there exists maximum one object per frame. However, the detector itself can detect several objects per frame, meaning that it is easy to modify the algorithm if tracking multiple objects is desired in the future. The image frames are sent from the camera feed and transformed into OpenCV image objects, before they are sent as input to the detection algorithm.

The OpenCV image objects are used by OpenCV, which is a powerful open source computer vision framework that allows us to perform various operations to the image, e.g. scaling, blurring, color detection, and so on. In the experiments, the system runs on a HP Laptop with Intel Core i7-7700HQ, 16 GB RAM and an NVIDIA GeForce GTX 1060 (6 GB GDDR5 dedicated) graphics card. With this setup, the detection can perform at approximately 30-40 fps for 1080x720 resolution video, which provides real-time compatibility.

IV. MOTION CONTROL SYSTEM

This section describes the motion control system (MCS), i.e. the navigation, guidance and control system, that has been developed for the vehicle used in the experimental testing. The objective is to keep a fixed position and heading angle of an underwater vehicle through dynamic positioning (DP) relative to an object. It is important to note that the MCS only performs during successful detection of the object, and that the vehicle should use a lower level MCS when the object cannot be detected. A simple Kalman filter is used for estimating the vehicle's velocity, and a sliding mode controller based on the velocity is responsible for controlling the vehicle. Automatic pitch and roll proportional-integral-derivative controllers stabilize the vehicle in roll and pitch. Consequently, roll and pitch motions are handled by a lower level inertial navigation system, and are omitted in the MCS presented here. All forces and moments are handled by a thrust allocation system in order for the thrusters to produce the desired amount of torque.

A. Navigation System

The navigation system only considers the kinematic model of the underwater vehicle relative to the detected object. The kinematic model represents the vehicle's states, which are defined by its position and orientation (pose) $\eta = [p^T \theta^T]^T$ and velocity $\nu = [v^T \omega^T]^T$. Vehicle position and Euler angle orientation are described by the vectors $p = [x, y, z]^T$ and $\theta = [\psi]^T$, expressed in the object frame. Linear and angular velocity are defined as $v = [u, v, w]^T$ and $\omega = [r]^T$, expressed in the vehicle's body-frame. Due to the nature of monocular cameras, the position and velocity of the vehicle are only valid when the object is detected, and are defined as

$$\eta = [x \ y \ z \ \psi]^T + w_s^T \quad (1)$$

$$\nu = [u \ v \ w \ r]^T, \quad (2)$$

where the term w_s represents Gaussian distributed white noise. In order to find the values for η , we first calculate

$$\psi = \frac{FOV_w}{C_w/2} \cdot P_{obj,w} - FOV_w \quad (3)$$

$$\theta = \frac{FOV_h}{C_h/2} \cdot P_{obj,h} - FOV_h, \quad (4)$$

where ψ is the heading angle and θ is the pitch angle relative to the object, and where the rest of the parameters have been

defined in Table II. Next, the vehicle's distance to the object x is retrieved from the scaling function estimated through the PCHIP function $S(A)$, depicted in Fig. 4. The y - and z -distance relative to the object are then found by exploiting the geometrical relations through x and (3)-(4). The position estimation procedure can then be written as follows

$$x = S(A) \quad (5)$$

$$y = x \sin(\psi) \quad (6)$$

$$z = x \sin(\theta). \quad (7)$$

As previously mentioned, a Kalman filter is implemented to estimate the vehicle's velocity ν .

B. Guidance System

The guidance system generates appropriate references for the control system, and is here used to calculate reference velocities. First, the desired pose must be defined, and the velocity reference can be derived from this by a simple technique, similar to [33]. The desired position η_d is written as

$$\eta_d = [x_d \ y_d \ z_d \ \psi_d]^T, \quad (8)$$

where x_d, y_d, z_d, ψ_d correspond to the desired distance to the object in x -, y - and z -direction and heading. The reference velocity ν_r is used as a feedback to increase the convergence to the desired position and heading angle of the vehicle, and is calculated as

$$\nu_r = -\gamma(\eta_d - \eta), \quad (9)$$

where γ in (9) is the task reference gain matrix.

C. Control System

The control system utilizes a sliding mode controller (SMC) in order to make the states of the vehicle converge to the desired values by controlling them to a sliding manifold with global exponential stability properties [34]. Similar to [33], the manifold is chosen as

$$s = (\nu_r - \nu) + \Lambda \int_0^t (\nu_r - \nu) d\tau, \quad (10)$$

where Λ is the integral gain matrix, and where s is globally exponentially stable if $\Lambda > 0$. Finally, the control law [35] [33] is then given as

$$\tau = K_D s + \hat{g}(\Theta) + K_S \text{sat}(s, \epsilon). \quad (11)$$

In (11), $K_D > 0$ and $K_S > 0$ are gain matrices. By assuming that the vehicle is neutrally buoyant and that velocities will be small, restoring forces and moments represented by $\hat{g}(\Theta)$ can be omitted. Furthermore, the function $\text{sat}(s, \epsilon)$ refers to a saturation function of s with lower and upper bound of $\pm\epsilon$, and replaces the signum function to avoid chattering [33] [34].

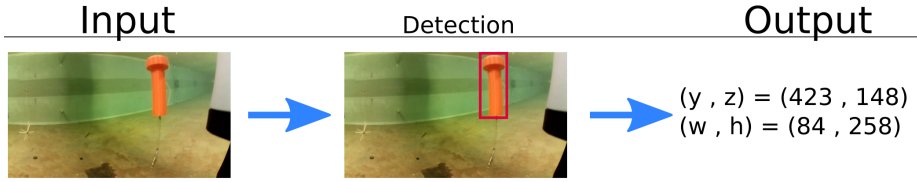


Fig. 5: Outline of the detection algorithm

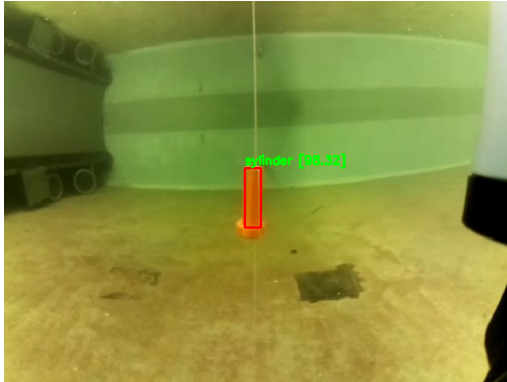


Fig. 6: An image from the camera of the vehicle, showing a successfully detected cylinder by the object detection algorithms, represented by a bounding box (red) and a confidence of 98.32% (green).



Fig. 7: The BlueROV2 underwater vehicle



Fig. 8: Water tank in the Marine Cybernetics Laboratory. Dimensions: L x B x D = 40m x 6.45m x 1.5m.

V. EXPERIMENTAL TESTING

In the experimental testing, a BlueROV2 has been used, which is a small-sized ROV. The vehicle is neutrally buoyant, has six degree of freedom and runs the robotic operating system (ROS) framework for message communication. It is equipped with a monocular camera in the front of the vehicle, where the direction of which it points coincides with the heading angle of the vehicle. A small information scheme representing some of the main features of the vehicle can be seen in Table III, and the vehicle itself is depicted in Fig. 7. Experiments have been conducted in the Marine Cybernetics Laboratory (MC-lab) at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. The water tank in the MC-lab where the experiments have been conducted is depicted in Fig. 8.

The experiment shows that the vehicle is able to perform DP relative to the object, controlled by a sliding mode controller (SMC) based on velocity estimates by a Kalman filter (KF). The KF incorporates the estimated position data of the object through a trained model of the object, as described in Section III. A successful detection during the dynamic positioning can be seen in Fig. 6, and the BlueROV2 and object during the

experiment is depicted in Fig. 9. This experiment was run for approximately 90 seconds, and the results from this test can be seen in Figs. 10 and 11. Position and heading errors can be seen in Fig. 10, while velocity and angular velocity errors are represented in Fig. 11.

The resulting plots in Fig. 11 show that the velocity is tracked with small errors. This gives fast convergence to the desired position and heading angle, as can be seen in Fig. 10. Root mean square error (RMSE) is presented in Table IV, denoted by a subscript for the respective state. This table shows an RMSE of around 2.5 [cm] in x- and y-direction, 1.6 [cm] in z-direction and 3.7 [deg] for the heading angle.

The gains of the guidance and control system have been

TABLE III: BlueROV2 specifications

Parameter	Value
L x H x W	457 [mm] x 254 [mm] x 575 [mm]
Weight in air	11.5 [kg]
Weight submerged	0 [kg]
Thrusters	T-200
Battery	14.8 [V], 10 [Ah]
Depth rating	100 [m]
Camera	Raspberry Pi Camera V2.1
Onboard Computer	Raspberry Pi 3B and Navio2

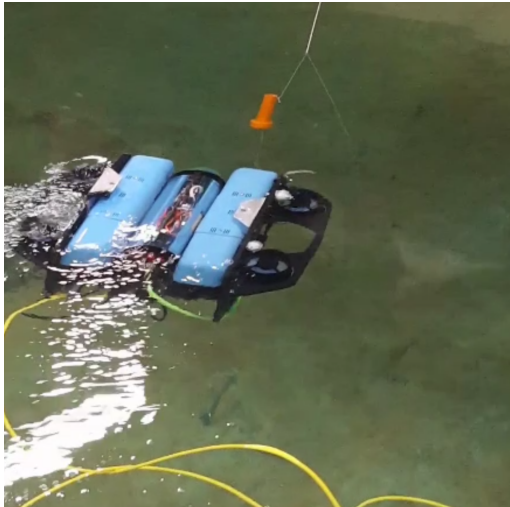


Fig. 9: The BlueROV2 performing dynamic positioning during an experiment in the pool in the Marine Cybernetics Laboratory.

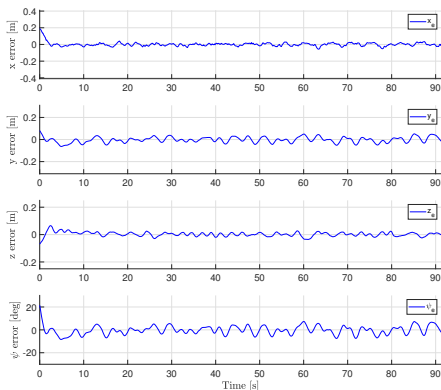


Fig. 10: Vehicle error position (blue) in x -, y -, z -direction [m] and heading angle ψ [deg] from top to bottom, respectively.

tuned to give a more aggressive steering in x -, y - and z -direction through γ in (9). The integral effect in (10) is slightly larger in z -direction and for the heading ψ compared to the x - and y -direction, in order to compensate for drag forces from the tether. Increased integral effect could have been applied for control in x - and y -direction as well, but the values used here were found to be suitable for following the desired reference velocities. The gain matrices \mathbf{K}_D and \mathbf{K}_S were chosen to be quite low. Furthermore, the unit of the reference velocity calculated in [cm/s], which might be the

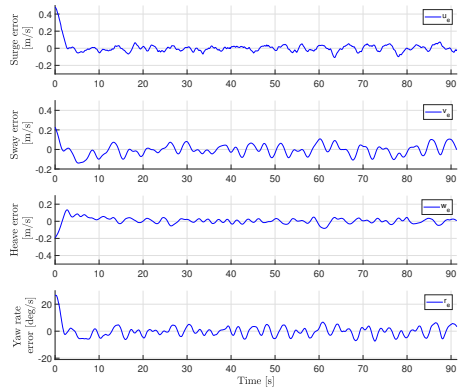


Fig. 11: Vehicle velocity error in x -, y -, z -direction [m/s] and heading angle ψ [deg/s] from top to bottom, respectively.

TABLE IV: Root Mean Square Error for position and velocity of the vehicle relative to the object

RMSE for position	Value	RMSE for velocity	Value
RMSE _{x}	0.024 [m]	RMSE _{u}	0.060 [m/s]
RMSE _{y}	0.025 [m]	RMSE _{v}	0.056 [m/s]
RMSE _{z}	0.016 [m]	RMSE _{w}	0.037 [m/s]
RMSE _{ψ}	3.7 [deg]	RMSE _{$\dot{\psi}$}	4.2 [deg/s]

reason for the seemingly small gain parameters in (9). The gains are presented in Table V.

VI. DISCUSSION

As previously described the object detector was verified on a subset of the entire dataset. The associated value of AP of 97.9% should therefore be considered with constraints. Several experiments should be conducted where new datasets can be retrieved for verification of the model. Optimally, several datasets should be retrieved over several experiments to the final training dataset as well, in order to ensure robustness of the model.

If retrieving more data proves difficult in the future, it is also possible to improve the verification with the currently obtained dataset. The dataset can be divided chronologically into groups of 4. The dataset can then first be trained on the parts 1, 2 and 3 and verified on part 4. Then a new training process should be conducted on parts 1, 2 and 4 and verification on part 3. The process can be repeated until the model has been verified on all parts and the new AP can be calculated for all

TABLE V: Gains and parameters used for reference velocity generation and tuning the sliding mode controller.

Parameter	Value
γ	[2.5 2.5 2.5 0.3]
Λ	[0.02 0.02 0.12 0.2]
\mathbf{K}_D	[$7.2 \cdot 10^{-4}$ $7.2 \cdot 10^{-7}$ $4.5 \cdot 10^{-3}$ $3.6 \cdot 10^{-6}$]
\mathbf{K}_S	[$3 \cdot 10^{-3}$ $3 \cdot 10^{-6}$ $1.9 \cdot 10^{-2}$ $1.5 \cdot 10^{-5}$]

the processes. This will give a better representation of the AP of the model. If the dataset is large, it can also be divided into more groups than 4. However increasing the number of groups will increase the time consumption of the whole process.

In the experimental testing, the MCS for the object tracking mission starts automatically after the object is first detected. The experimental results presented in Figs. 10-11 and Table IV show both good convergence rate and capability in maintaining the desired states, with errors below 2.5 [cm] in all directions and a heading angle error below 4 [deg].

The vehicle does experience some difficulties maintaining the desired heading angle, which can be seen from the oscillating behavior in Fig. 10. This is best explained by the combination of a delay that arises between image acquisition and when the image is ready for processing, and the time it takes for the vehicle's thrusters to propel the vehicle in the desired direction. Less oscillations were achieved by making the controller slower and to slightly decrease the effect of the integral effect. There is also a slight mismatch between the camera angle and the thrust allocation system, as the camera's angle is tilted approximately 5 to 10 degrees in the direction of a positive heading angle for the vehicle, as can be seen from Fig. 6. This may have contributed some to the error in the heading angle, in addition to the x- and y-position. Another factor that may have contributed to the errors and the somewhat oscillating behavior in Figs. 10-11 can be explained by the refraction of light caused by the dome in front of the camera in Fig. 7, as seen in Fig. 6. Yet, the detection algorithm had no trouble detecting the image, and the ROV nicely converged towards and maintained the desired states, and the effect was found to have negligible impact on the performance of the system. Future work may take the refraction into account by calibrating the camera's intrinsic parameters.

Experiments were also conducted for a moving object, although not documented. The experiments demonstrated that the vehicle managed to maintain the desired relative position and followed the object's movement. The experiments were conducted in a laboratory pool with still water and thus not tested for conditions with constant or alternating currents. However, it is believed that the presented MCS will be able to cope with currents, given proper tuning of gains, particularly the integral terms.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents a dynamic positioning procedure relative to an object of interest using a small-class fully actuated underwater vehicle. The object is detected based on a trained model of a large image dataset that contains the object of interest in an underwater environment, using a monocular camera. Furthermore, the paper presents a powerful labeling procedure of the object within the dataset, and a model trained on these images. Experimental testing results prove the effectiveness of the proposed methods, where a small underwater vehicle performs DP relative to the object with small velocity tracking and position errors. Further work involves adapting

the proposed methods to an underwater vehicle-manipulator system for simultaneous DP on the vehicle and gripping with the manipulator arm, and may look into environments with constant or varying disturbances such as currents.

REFERENCES

- [1] I. Schjølberg, T. B. Gjersvik, A. A. Transeth, and I. B. Utne, "Next generation subsea inspection, maintenance and repair operations," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 434 – 439, 2016, 10th IFAC Conference on Control Applications in Marine Systems CAMS 2016.
- [2] Z. Chen, Z. Zhang, F. Dai, Y. Bu, and H. Wang, "Monocular vision-based underwater object detection," in *Sensors*, 2017.
- [3] Z. Chen, Z. Zhang, Y. Bu, F. Dai, T. Fan, and H. Wang, "Underwater object segmentation based on optical features," *Sensors (Basel, Switzerland)*, vol. 18, 01 2018.
- [4] H. Cho, J. Gu, H. Joe, A. Asada, and S.-C. Yu, "Acoustic beam profile-based rapid underwater object detection for an imaging sonar," *Journal of Marine Science and Technology*, vol. 20, no. 1, pp. 180–197, Mar 2015.
- [5] F. Bonin-Font, G. Oliver, S. Wirth, M. Massot, P. L. Negre, and J.-P. Beltran, "Visual sensing for autonomous underwater exploration and intervention tasks," *Ocean Engineering*, vol. 93, pp. 25 – 44, 2015.
- [6] Q. Xi, T. Rauschenbach, and L. Daojiang, "Review of underwater machine vision technology and its applications," *Marine Technology Society Journal*, vol. 51, no. 1, pp. 75–97, 2017. [Online]. Available: <https://www.ingentaconnect.com/content/mts/mts/2017/00000051/00000001/art00009> <https://doi.org/10.4031/MTSJ.51.1.8>
- [7] 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*, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231219304230>
- [8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.
- [9] Y. He, B. Zheng, Y. Ding, and H. Yang, "Underwater image edge detection based on k-means algorithm," in *2014 Oceans - St. John's*, Conference Proceedings, pp. 1–4.
- [10] M. Narimani, S. Nazem, and M. Loueipour, "Robotics vision-based system for an underwater pipeline and cable tracker," in *OCEANS 2009-EUROPE*, Conference Proceedings, pp. 1–6.
- [11] Z. Chen, Z. Zhang, F. Dai, Y. Bu, and H. Wang, "Monocular vision-based underwater object detection," *Sensors (Basel, Switzerland)*, vol. 17, no. 8, p. 1784, 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pubmed/28771194> <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5580077/>
- [12] H. Madjidi and S. Negahdaripour, "On robustness and localization accuracy of optical flow computation for underwater color images," *Computer Vision and Image Understanding*, vol. 104, no. 1, pp. 61–76, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S107731420600083X>
- [13] Madjidi, Hossein and Negahdaripour, Shahriar, "On robustness and localization accuracy of optical flow computation for underwater color images," *Computer Vision and Image Understanding*, vol. 104, no. 1, pp. 61–76, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S107731420600083X>
- [14] 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*, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231219304230>
- [15] H. Qin, X. Li, Y. Zhixiong, and M. Shang, "When underwater imagery analysis meets deep learning: A solution at the age of big visual data," in *OCEANS 2015 - MTS/IEEE Washington*, Conference Proceedings, pp. 1–5.
- [16] M. Moniruzzaman, S. M. S. Islam, M. Bennamoun, and P. Lavery, "Deep learning on underwater marine object detection: A survey," in *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, R. Penne, W. Philips, D. Popescu, and P. Scheunders, Eds. Springer International Publishing, Conference Proceedings, pp. 150–160.
- [17] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2019.
- [18] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

- [19] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.
- [20] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.
- [21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.
- [22] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions auvs," *Ocean Engineering*, vol. 36, no. 1, pp. 15 – 23, 2009, autonomous Underwater Vehicles.
- [23] P. J. Sanz, P. Ridaou, G. Oliver, G. Casalino, Y. Petillot, C. Silvestre, C. Melchiorri, and A. Turetta, "Trident an european project targeted to increase the autonomy levels for underwater intervention missions," in *2013 OCEANS - San Diego*, Sep. 2013, pp. 1–10.
- [24] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the maris project," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 3, pp. 620–639, July 2018.
- [25] A. Huster and S. M. Rock, "Relative Position Estimation for Intervention-Capable AUVs by Fusing Vision and Inertial Measurements," in *Proceedings of the 12th International Symposium on Unmanned Untethered Submersible Technology*, Autonomous Undersea Systems Institute. Durham, NH: Autonomous Undersea Systems Institute, August 2001. [Online]. Available: http://www.stanford.edu/group/ar/cgibin/drupal/sites/default/files/public/publications/HusterR_2001a.pdf
- [26] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the maris project," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 3, pp. 620–639, 2018.
- [27] F. Bonin-Font, G. Oliver, S. Wirth, M. Massot, P. Lluís Nègre, and J.-P. Beltran, "Visual sensing for autonomous underwater exploration and intervention tasks," *Ocean Engineering*, vol. 93, pp. 25–44, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0029801814004090>
- [28] R. Dario Lodi, K. Fabjan, O. Fabio, and C. Stefano, "Investigation of vision-based underwater object detection with multiple datasets," *International Journal of Advanced Robotic Systems*, vol. 12, no. 6, p. 77, 2015.
- [29] A. B. Labao and P. C. Naval, "Cascaded deep network systems with linked ensemble components for underwater fish detection in the wild," *Ecological Informatics*, vol. 52, pp. 103–121, 2019.
- [30] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi, "Aggregating crowdsourced binary ratings," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: ACM, 2013, pp. 285–294.
- [31] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 1953–1961.
- [32] J. Xu and H.-S. Yoon, "Vision-based estimation of excavator manipulator pose for automated grading control," *Automation in Construction*, vol. 98, pp. 122 – 131, 2019.
- [33] B. O. A. Haugaløkken, E. K. Jørgensen, and I. Schjøllberg, "Experimental validation of end-effector stabilization for underwater vehicle-manipulator systems in subsea operations," *Robotics and Autonomous Systems*, vol. 109, pp. 1 – 12, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889018300952>
- [34] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. Prentice-Hall Englewood Cliffs, NJ, 1991, vol. 199.
- [35] G. Antonelli, *Underwater Robots*. Springer, 2014.

Article 3

Monocular vision-based gripping of objects

Bent. O. A. Haugaløkken, **Martin B. Skaldebø**, Ingrid Schjølberg
Robotics and Autonomous Systems
[doi: /10.1016/j.robot.2020.103589](https://doi.org/10.1016/j.robot.2020.103589)



Monocular vision-based gripping of objects

Bent Oddvar Arnesen Haugaløkken*, Martin Breivik Skaldebo, 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.

© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

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. Skaldebo), ingrid.schjolberg@ntnu.no (I. Schjølberg).

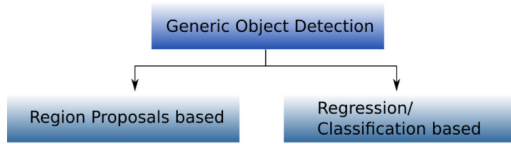


Fig. 1. Object detection methodologies.

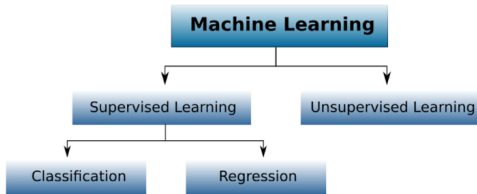


Fig. 2. Machine learning methodologies.

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 of 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

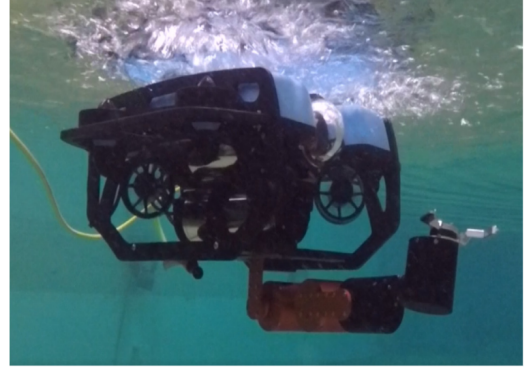


Fig. 3. The BlueROV2 underwater vehicle equipped with the SeaArm manipulator arm.

Table 2
Denavit–Hartenberg parameters.

i	α_{i-1} [rad]	a_{i-1} [mm]	d_i [mm]	θ_i [rad]
1	0	0	55.3	θ_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

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.

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

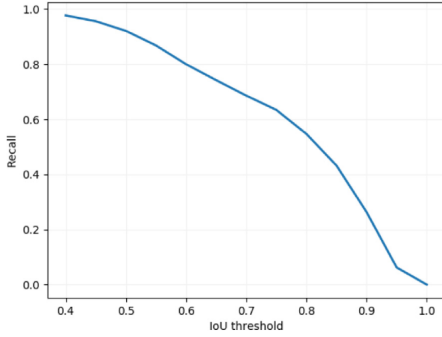


Fig. 4. Recall vs. IoU threshold values.

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

$$\dot{\eta} = J_R(\eta)v \quad (1)$$

$$\dot{\eta}_{ee} = J_e(q, \eta)\zeta, \quad (2)$$

where $J_e = [J_1(q) \ J_2(\eta)]$, and $\zeta = [\dot{q}^T \ v^T]^T$ represents the velocity of both the UVMS body and the manipulator. J_R denotes the Jacobian for the UVMS and J_e is the Jacobian relating the end-effector time derivative to ζ , 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

$$x = S_A \quad (3)$$

$$y = x \sin(\psi) \quad (4)$$

$$z = x \sin(\theta). \quad (5)$$

where ψ is the heading angle and θ 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

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

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

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 ψ and pitch angle θ 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 v based on the relative position and angle estimates in (3)–(7).

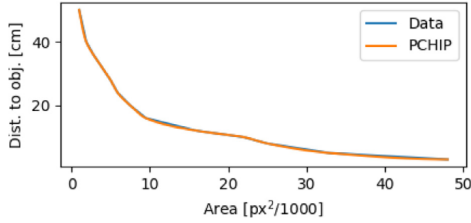


Fig. 5. Visual representation of the scaling function. The object area [px²/1000] is plotted along the x -axis with the corresponding distance to the object [cm] on the y -axis. The original measured data is shown in blue and the area scaling function based on the PCHIP is visualized in orange [31]. (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 $\mathbf{Q} = \text{diag}(3^2, 4^2, 5^2)$ and measurement noise matrix $\mathbf{R} = \text{diag}(0.5, 0.5, 0.5)$, 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 χ can be defined through a generic variable as

$$\sigma_\chi = \sigma_\chi(\eta, \mathbf{q}) \quad (8)$$

and the task-specific Jacobian

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

where the value χ 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 ψ , 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

$$\sigma_1 = [x \quad z \quad \psi]^T \quad (10)$$

$$\mathbf{J}_1 = \mathbf{I}_{3 \times 3}, \quad (11)$$

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

$$\sigma_2 = [x_{ee} \quad y_{ee} \quad z_{ee}]^T, \quad (12)$$

where the task Jacobian \mathbf{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

$$\zeta_r = \mathbf{J}_\chi^\dagger(\eta)\dot{\sigma}_{\chi,r}. \quad (13)$$

The pseudo-inverse is given as

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

Note that the notation for (η, \mathbf{q}) is now omitted for enhanced readability. The parameter $\dot{\sigma}_{\chi,r}$ 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

$$\dot{\sigma}_{\chi,r} = -\mathbf{K}_\chi \tilde{\sigma}_\chi, \quad (15)$$

where \mathbf{K}_χ is a gain matrix. Furthermore, the task error is chosen as $\tilde{\sigma}_\chi = \sigma_{\chi,d} - \sigma_\chi$, where $\sigma_{\chi,d}$ represents the desired values for task χ . The part of $\sigma_{\chi,d}$ 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 \mathbf{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

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

where λ 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 λ 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 ζ_r for the vehicle is given as

$$\zeta_r = -\mathbf{J}_1^\dagger \mathbf{K}_1 \tilde{\sigma}_1, \quad (17)$$

where \mathbf{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

$$\mathbf{s} = (\mathbf{v}_r - \mathbf{v}) + \Lambda \int_0^t (\mathbf{v}_r - \mathbf{v}) dt, \quad (18)$$

where \mathbf{v}_r is the reference velocities and Λ is an integral gain matrix. Finally, the control law [7] is then given as

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

In (19), \mathbf{K}_D and \mathbf{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 $\hat{\mathbf{g}}(\Theta)$ can be omitted. Furthermore, the function $\text{sat}(\mathbf{s}, \epsilon)$ refers to a saturation function of \mathbf{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)

$$V = \frac{1}{2} \mathbf{s}^T \mathbf{s} > 0, \quad \forall \mathbf{s} \neq \mathbf{0}, \quad (20)$$

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

$$\dot{V} = \mathbf{s}^T \dot{\mathbf{s}} \quad (21)$$

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

$$\mathbf{s} = \tilde{\mathbf{v}} + \Lambda \int_0^t \tilde{\mathbf{v}} dt \quad (22)$$

$$= \mathbf{J}^\dagger \dot{\tilde{\sigma}} + \Lambda \mathbf{J}^\dagger \tilde{\sigma} dt. \quad (23)$$

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

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

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

$$\dot{\mathbf{s}} = \dot{\tilde{\mathbf{v}}} + \frac{d}{dt} \left(\Lambda \int_0^t \tilde{\mathbf{v}} dt \right). \quad (25)$$

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

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

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

$$\dot{\mathbf{s}} = \mathbf{J}^\dagger \mathbf{K} \dot{\sigma} - \Lambda \mathbf{J}^\dagger \mathbf{K} \tilde{\sigma} - \Lambda \mathbf{J}^\dagger \dot{\sigma}. \quad (27)$$

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

$$\dot{\mathbf{s}} = -\mathbf{J}^\dagger \mathbf{K}^2 \tilde{\sigma}. \quad (28)$$

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

$$\dot{V} = \mathbf{s}^T \dot{\mathbf{s}} \quad (29)$$

$$= (\mathbf{J}^\dagger (\mathbf{K} + \Lambda) \tilde{\sigma})^T (-\mathbf{J}^\dagger \mathbf{K}^2 \tilde{\sigma}) \quad (30)$$

$$= -\tilde{\sigma}^T \mathbf{M} \tilde{\sigma}, \quad (31)$$

where $\mathbf{M} = (\mathbf{K}^2 (\mathbf{K} + \Lambda)^T)$ 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 $\mathbf{s} = \mathbf{0}$ given that $\lim_{t \rightarrow \infty} \tilde{\sigma} = \mathbf{0}$. To prove that the latter condition holds, consider the following the CLF

$$V^* = \frac{1}{2} \tilde{\sigma}^T \tilde{\sigma}. \quad (32)$$

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

$$\dot{V}^* = \tilde{\sigma}^T \dot{\tilde{\sigma}} \quad (33)$$

$$= -\tilde{\sigma}^T \dot{\sigma}. \quad (34)$$

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

$$\dot{V}^* = -\tilde{\sigma}^T \mathbf{K} \tilde{\sigma} \quad (35)$$

Hence, with \mathbf{K} chosen as positive definite, \dot{V}^* is negative definite, and $\tilde{\sigma} = \mathbf{0}$ is exponentially stable. It follows that the sliding surface converges asymptotically to zero. Note that the results only hold locally, i.e. if σ belongs to a compact set, as $\tilde{\sigma}$ 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 σ (and σ_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 $[x_d \ z_d \ \psi_d] = [0.18 \ \text{m} \ -0.05 \ \text{m} \ -5^\circ]$ 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.

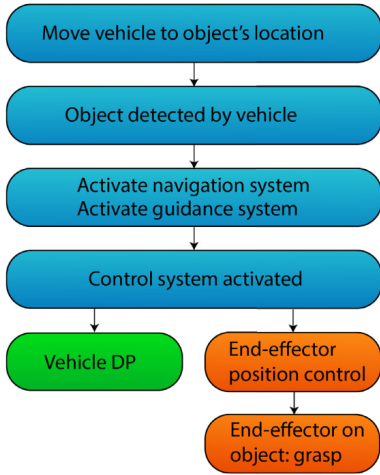


Fig. 6. A flow chart showing the experimental testing procedure.

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

Parameter	Value
K_1	[2.5 2.5 0.3]
Λ	[0.02 0.12 0.2]
K_D	[$7.2 \cdot 10^{-4}$ $4.5 \cdot 10^{-3}$ $3.6 \cdot 10^{-6}$]
K_S	[$3 \cdot 10^{-3}$ $1.9 \cdot 10^{-2}$ $1.5 \cdot 10^{-5}$]

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

Position	Value	Velocity	Value
RMSE _x	0.025 [m]	RMSE _{v_x}	0.027 [m/s]
RMSE _z	0.025 [m]	RMSE _{v_z}	0.024 [m/s]
RMSE _ψ	6.7 [deg]	RMSE _{v_ψ}	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-

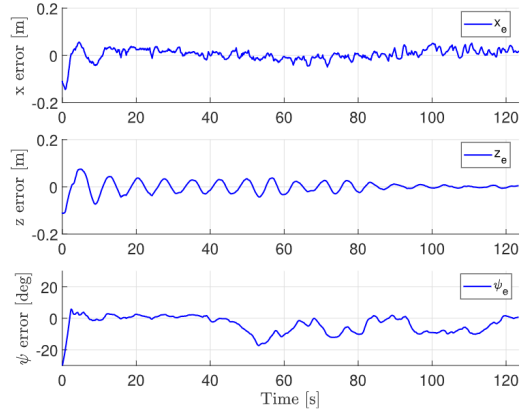


Fig. 7. Vehicle error position in x- and z-direction [m] and heading angle ψ [deg] during the vehicle DP operation.

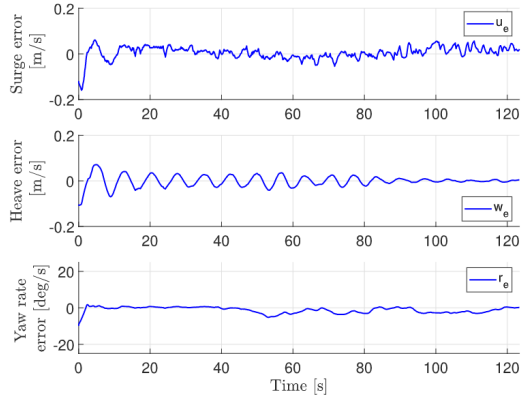


Fig. 8. Vehicle velocity error in surge and heave [m/s] and yaw rate ψ [deg/s] during the vehicle DP operation.

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

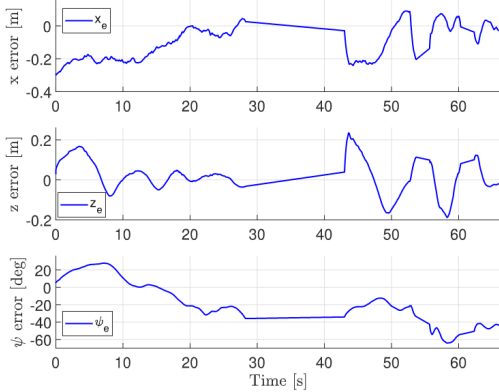


Fig. 9. Vehicle error position in x- and z-direction [m] and heading angle ψ [deg] during the first grasping operation.

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 _ψ	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.

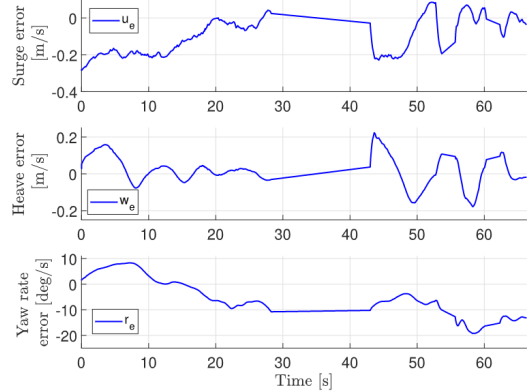


Fig. 10. Vehicle velocity error in surge and heave [m/s] and yaw rate ψ [deg/s] during the first grasping operation.

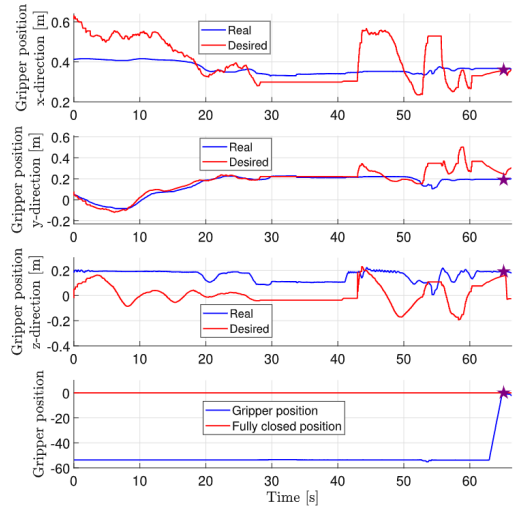


Fig. 11. The real end-effector (blue) and desired (red) position in x-, y-, z-direction during the first grasping operation. The star-symbol marks the point in time where the successful grasp of the object took place. (For interpretation of the references to color in this figure legend, the reader is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

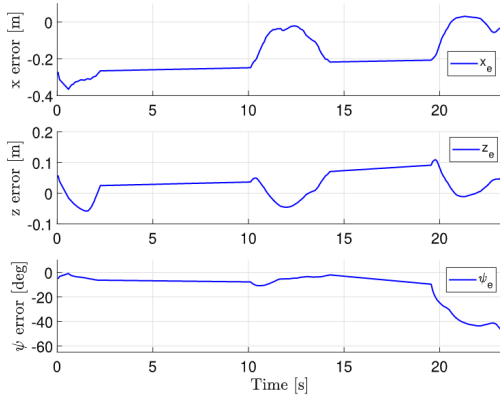


Fig. 12. Vehicle error position in x- and z-direction [m] and heading angle ψ [deg] during the second grasping operation.

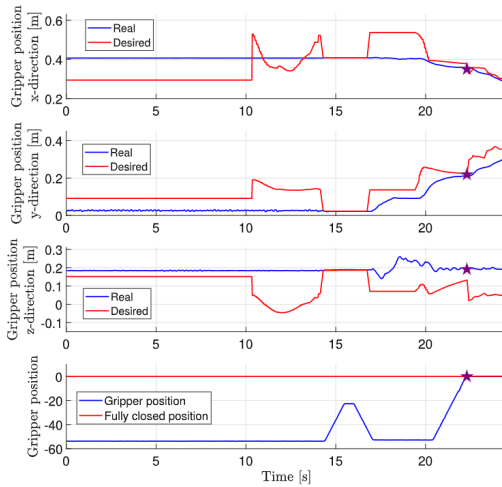


Fig. 13. The real end-effector (blue) and desired (red) position in x-, y- and z-direction during the second grasping operation. The star-symbol marks the point in time where the successful grasp of the object took place. (For interpretation of the references to color in this figure legend, the reader 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
$RMSE_{e_x}$	0.15 [m]	$RMSE_{\dot{e}_v}$	0.17 [m/s]	$RMSE_{ee,x}$	0.09 [m]
$RMSE_{e_z}$	0.04 [m]	$RMSE_{\dot{e}_w}$	0.04 [m/s]	$RMSE_{ee,y}$	0.07 [m]
$RMSE_{\psi}$	27 [deg]	$RMSE_{\dot{e}_r}$	8 [deg/s]	$RMSE_{ee,z}$	0.11 [m]



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

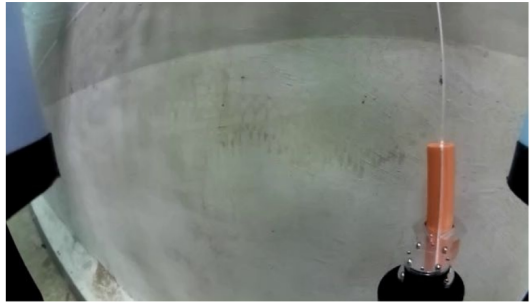


Fig. 15. The UVMS successfully grasping the object seen from the vehicle's camera during the second grasping operation.

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 $[x_d \ z_d \ \psi_d] = [0.18 \text{ m} \ -0.05 \text{ m} \ -5^\circ]$. 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 is 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 $0.027 \frac{\text{m}}{\text{s}}$, $0.024 \frac{\text{m}}{\text{s}}$ and $2 \frac{\text{deg}}{\text{s}}$ 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 λ 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 sub-sea 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. Scarica, Autonomous underwater intervention: Experimental results of the maris project, *IEEE J. Ocean. Eng.* 43 (3) (2018) 620–639.
- [26] P. Ridaou, 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. Ilzkovitz, P. Weiss, F. Gauch, G. Antonelli, G. Indiveri, G. Casalino, A. Birk, M.F. Pflingsthor, S. Calinon, A. Tanwani, A. Turetta, C. Walen, L. Guilpain, Dextror: 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, Dextror 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.

Article 4

SeaArm-2 - Fully electric underwater manipulator with integrated end-effector camera

Martin B. Skaldebø, Bent O. A. Haugaløkken, Ingrid Schjølbørg
European Control Conference (ECC), 2021
[doi: 10.23919/ECC54610.2021.9655121](https://doi.org/10.23919/ECC54610.2021.9655121)

In reference to IEEE copyrighted material, which is used with permission in this thesis, the IEEE does not endorse any of NTNU's products or services, Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to <https://www.ieee.org/publications/rights/rights-link.html> to learn how to obtain a License from RightsLink

SeaArm-2 - Fully electric underwater manipulator with integrated end-effector camera

Martin B Skaldebø¹ Bent O. A. Haugaløkken² and Ingrid Schjølberg³

Abstract—This paper presents an electrically driven underwater manipulator named SeaArm-2 and considers the kinematic control approach to perform autonomous fixed-base grasping of an object of known size and shape at unknown positions underwater. SeaArm-2 has an integrated monocular camera near the gripper, which enables perception and object detection capabilities without the need for an external system. Furthermore, this work presents autonomous fixed-base seabed grasping operations utilizing the integrated camera and object detection functionality through a novel open source machine learning framework. A total of 20 trials have been conducted where the object was placed within reach but out of sight of the manipulator. The results from the experimental testing showed that the manipulator was able to detect the object in all trials, and successfully grasp the object in 17/20 trials.

Robot manipulators, autonomy, underwater technology, computer vision, monocular camera, machine learning, kinematic control

I. INTRODUCTION

There is a need for increased levels of autonomy in subsea inspection, maintenance, and repair (IMR) operations. Intervention is related to maintenance and repair, and require tools such as a robot manipulator in order to interact with the environment. Manipulators are often mounted on an unmanned underwater vehicle (UUV), which serves as platforms that move the manipulator base. Autonomous UUVs with manipulation capabilities are also referred to as underwater vehicle manipulator systems (UVMS). These systems are of increasing interest to the maritime industry because they may increase safety and reduce operational costs significantly [1], [2].

Underwater manipulators have been studied since the beginning of 1990, both driven by hydraulic and electric actuators. Hydraulic actuators typically have high force/torque output, however at the cost of low position accuracy which means they are not well suited for intervention operations that require accurate force/torque control. Electric actuators on the other hand are capable of providing motions with high accuracy, but due to the nature of electric actuators in terms of size and power, the force/torque output is severely limited compared to its hydraulic counterpart. Developing smaller arms and at a lower cost is another advantage of the

electrically driven manipulators, and the maintenance cost of such systems is significantly lower compared to hydraulic systems. Electric manipulators can also be connected to the vehicle's power system and do not depend on additional infrastructure as is the case for hydraulic systems [3].

Autonomous or semi-autonomous grasping of objects in an underwater environment is challenging because equipment is expensive or difficult to develop, operations in the underwater environment are highly complex and high accuracy is required. Several systems need to work together in real time, such as cameras, object detection and/or scene reconstruction methods, object pose estimation, manipulator or UVMS control, grasping pose and control of gripper pose for achieving the grasp. Underwater object grasping has received increasing attention in the recent years. In [4], computer vision was used to detect a marker next to the object and a combination of color and edge (Hough transform) detection was adopted to identify the object. [5] used a multi-view laser reconstruction method to map the scene with high precision using a camera and a laser emitter near the end effector. The largest object in the scene was chosen as the subject for grasping, and a grasp planner estimated a good end effector pose and grasping points. The aforementioned works made use of manipulators with a non-moving base, but manipulators with moving bases, e.g. when placed on underwater vehicles, are also of interest. [6], [7], [8], [9] presented successful autonomous object retrieval with UVMS. Furthermore, and more recently, [10] presented successful sea trials where grasping of marine species was achieved, however without causing damage, using an UVMS. According to [2], grasping of objects on the seabed is among the most studied topics within underwater manipulation, both in pools and at sea, but there is still a need for perception of such objects and autonomous grasping. For the purpose of perceiving objects, underwater object detection using cameras has great potential within autonomous intervention operations. Detection of reflected light, i.e. images and video, can be extracted by utilizing monocular (2D) or stereo (3D) camera systems, in addition to "2.5D" solutions that project 2D images to reconstruct 3D environment features [11].

In this work, an electrically driven underwater manipulator named SeaArm-2 is presented, which is the second version of the small and modular electric underwater manipulator first presented in [12]. This manipulator has integrated monocular camera near the gripper, which enables perception and object detection capabilities without the need for an external system. The manipulator is designed to be waterproof up to 500 meters. A microcontroller in the base of the manipulator

¹Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway, martin.b.skaldebo@ntnu.no

²Department of Seafood Technology, SINTEF Ocean, Trondheim, Norway, bent.haugalokken@sintef.no

³Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Trondheim, Norway, ingrid.schjolberg@ntnu.no

processes data and commands manipulator motions based on the manipulator state and camera input. Furthermore, this work presents fully autonomous fixed-base grasping utilizing the integrated camera and object detection functionality through a novel open source machine learning and computer vision framework. The contributions of the article are listed below

- 1) Presentation of a new small, modular, flexible and low cost electric underwater manipulator
- 2) Autonomous detection, movement, and grasping of known objects in unknown position
- 3) Experimental validation of the functionality of the manipulator and the autonomous fixed-base grasping operation

The article is structured as follows: Section II presents the SeaArm-2 manipulator design, the micro-controller, camera, and the communication system; Section III deals with position and velocity control of the manipulator; Section IV contains the computer vision, machine learning systems and training of the model; Section V describes the experimental setup and the testing procedure, while Section VI presents the results from the experimental tests. Finally, Section VII is dedicated to a discussion of the results and the concluding remarks from this work.

II. MANIPULATOR

A. Design

The presented manipulator is developed based on the SeaArm manipulator presented in [12]. The most fundamental properties are kept from the previous version while others have been updated. The manipulator is still completely modular with identical modules except from the module at the end which now includes housing for camera and a transparent dome.

The SeaArm-2 manipulator is depicted in Fig. 1. It is designed as a set of modular housings, and is comprised of a base, four standard housing modules, and an end effector module containing camera and end effector tool. The mechanical design of the modules and internal mechanical features are designed to enable the manipulator to withstand up to 500m of water pressure. Furthermore, the design allows for continuous rotation, as depicted in Fig. 2. Modules can be removed or added, which enables customization of the number of degrees of freedom (DOFs), manipulator size, maximum reach length and work space. The main specifications of SeaArm-2 are presented in Table I. The camera within the end effector module is a Low-Light HD USB Camera (based on Sony IMX322), and is suitable for use in an underwater environment with limited visible light. The camera is slightly tilted towards the gripper to get a more substantial view of the gripper operation.

B. Connection & Communication

The manipulator is connected to the topside unit through an Ethernet cable. This cable is responsible for enabling fast and reliable communication and enables power delivery

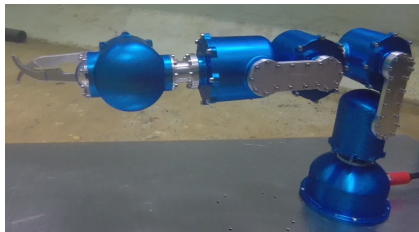


Fig. 1. SeaArm-2 manipulator.

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)	693.75mm
Number of servos	5
Stall torque at 12.0 V	25.2 Nm
Full reach lift	5 Kg
Depth rating	500m
Gear ratio	3 : 1
On-board computer	Raspberry Pi 3B
Communication	RS485 and Ethernet
Camera	Low Light HD USB Camera

through a power supply. Furthermore, the connection enables both additional computational power and operational supervision. Communication is set up in a client-server architecture with Ethernet connection, where a Raspberry Pi (RPi) located in the manipulator base represents the server and the topside computer system is the client. The RPi controls the servos using an RS485 interface and receives camera data through a USB connection. The RPi performs calculations and distributes forces and torques to the servos in order to achieve the desired movement of the manipulator. Operations that involve a significant computational load, such as object detection or matrix calculations for inverse kinematics, are calculated by the client computer before appropriate reference signals are sent to the manipulator. To facilitate this workflow, the status of the servos and raw camera video stream are continuously transmitted to the topside client, hence making this information available for

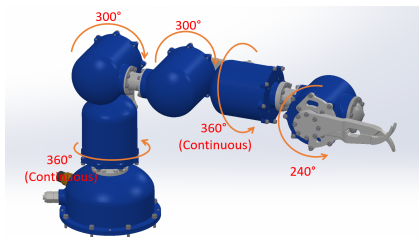


Fig. 2. SeaArm-2 manipulator joint revolutions.

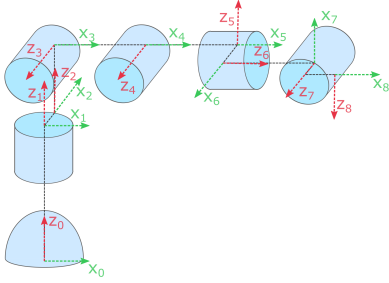


Fig. 3. Coordinate axis system of the manipulator

the client computer. The client computer performs calculations, analyze camera data and transmits commands to the manipulator at a rate of 15 Hz, however this rate can be increased or decreased if preferred.

III. KINEMATIC CONTROL

This section presents the control framework for the SeaArm-2 manipulator. The manipulator is controlled through a kinematic control framework, i.e. geometrical relations, as opposed to kinetics or dynamic control that relates the motions to forces and torques.

The Denavit-Hartenberg (DH) convention is used in order to represent the transformation matrix for the manipulator [13]. Fig. 3 illustrates the coordinate axis system used to calculate the DH parameters, and these are listed for the manipulator in Table II. Note that these parameters are the representation of the manipulator configuration as assembled in Fig. 1. The coordinate frames are chosen with the camera in mind, as frame 8 is placed in the position of the camera and has the same coordinate configuration as the frame produced by the camera. The manipulator has 5 servos, however only four applicable theta angles exist in Tab. II, where the last servo is dedicated to gripper actuation.

TABLE II
DENAVIT-HARTENBERG PARAMETERS.

i	d_i [mm]	θ_i [rad]	a_i [mm]	α_i [rad]
1	145.4	θ_1	0	0
2	0	$\pi/2$	50.1	0
3	80.0	$-\pi/2$	0	$\pi/2$
4	0	θ_2	112.0	0
5	0	θ_3	80.0	$-\pi/2$
6	0	$-\pi/2$	50.1	$-\pi/2$
7	143.0	$-\pi/2 + \theta_4$	0	$-\pi/2$
8	25.0	$-\pi/2$	60.0	$\pi/2$

The coherence between the transformation matrix and the DH-parameters is represented by

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & ac\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & as\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

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

$$\mathbf{T}_{ee}^{\text{base}} = \mathbf{T}_1^0 \mathbf{T}_2^1 \mathbf{T}_3^2 \mathbf{T}_4^3 \mathbf{T}_5^4 \mathbf{T}_6^5 \mathbf{T}_7^6 \mathbf{T}_8^7. \quad (2)$$

This results in a very comprehensive matrix and due to space constraints it is not shown here. The transformation matrix is used in both the forward and inverse kinematic solutions used in this paper.

A. Forward kinematics

The forward kinematics are used for calculating the position state of the manipulator. The state of each joint is gathered and translated from unit values to absolute angles. Then the angle of each joint is used in order to calculate the position of the end effector relative to the manipulator base. The relative position $p^{\{b\}}$ is calculated using

$$\begin{bmatrix} p^{\{b\}} \\ 1 \end{bmatrix} = \mathbf{T}_{\{s\}}^{\{b\}} \begin{bmatrix} p^{\{s\}} \\ 1 \end{bmatrix}, \quad (3)$$

where $p^{\{s\}} = (0, 0, 0)$.

B. Inverse kinematics

Inverse kinematics is used in order to calculate the joint angles from a target position, which will be determined by the object to be grasped. There are mainly two different ways of determining the inverse kinematic, closed-form mathematical solution and as an optimization problem. Since optimization does not guarantee the best solution, due to local minima in the minimization approach we will continue with the closed-form mathematical solution. A closed form solution requires more advanced mathematics and may include singularities. Singularities represents a point where a unique solution does not exist, meaning there exist either no solution or infinite solutions. How we cope with singularities are mentioned later in this section. The equation for the closed-form solution can be written as

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{X}), \quad (4)$$

where \mathbf{q} represents joint positions, \mathbf{X} is the Cartesian coordinates of the end effector and \mathbf{f} is a function that relates joint positions to Cartesian space. Differentiating the closed-form equation gives

$$\dot{\mathbf{X}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (5)$$

where \mathbf{J} is the Jacobian matrix of the manipulator. The Jacobian explains the velocity relationships of \mathbf{f} in (4). Using the Jacobian and (5) the inverse kinematic problem becomes a problem of solving linear equations, more specifically

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})^{-1} \dot{\mathbf{X}}. \quad (6)$$

To ensure that (6) can be solved even when \mathbf{J} is not square and the inverse can not be computed the pseudoinverse of the Jacobian \mathbf{J}^\dagger is used instead. This gives

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q})\dot{\mathbf{X}}, \quad (7)$$

where

$$\mathbf{J}^\dagger = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda\mathbf{I})^{-1}, \quad (8)$$

and λ is a damping term that makes the joints stop when the manipulator closes in on a singularity. However, this introduces small position errors for the end effector position when attempting to reach a desired position [14].

C. Kinematic control

All of the servos are controlled through angular velocities, where the inverse kinematics estimate the desired joint velocities, which are used as a reference for velocity control of the manipulator. The velocity controller takes the reference velocities as input to internal PID-controllers in the servos that converts the reference velocities to PWM values to operate the servo output. Moreover, the desired joint velocities are retrieved through (7), where $\dot{\mathbf{X}}$ is the desired end effector velocity calculated from

$$\dot{\mathbf{X}} = \gamma(\sigma_d - \sigma). \quad (9)$$

Here, γ is the gain, and σ_d and σ are the desired and measured position of the end effector in Cartesian coordinates. The desired position is retrieved as a waypoint from the mission planner, which is presented in Section V-B.

IV. COMPUTER VISION

The manipulator is equipped with a low-light HD USB camera, and is therefore a unique platform for combining underwater manipulation with computer vision. The video stream can be used for supervision to aid in manual control or in autonomous monitoring of the environment as a tool for autonomous intervention. In the work conducted in this paper computer vision is used to incorporate state-of-the-art object detection solutions to identify objects of interest in an autonomous gripping system. This section describes the object detection system, the development of the model through machine learning, and position estimation of the object.

A. Object detector

The object detector is based on YOLOv5 developed by [15]. This algorithm is an extension of the popular YOLO algorithms originally developed by Joseph Redmond and later with help of Ali Farhadi on YOLOv3 [16]. The detector was trained to detect one single object used in the autonomous gripping system, and is built as a neural network that is trained using supervised learning algorithms with labeled images of the object in order to learn to identify it. The object is a clear orange cylindrical object depicted in Fig. 4. The YOLOv5 algorithm offers 4 different network versions, a small, medium, large, and an extra large version. Since the detector should be a small part of a larger control system it should be as fast as possible, however there is a trade-off between computational power and precision and accuracy.

The image dataset consists of 6345 images in the training set, 1585 images in the validation set and 1114 images in the test set. Typically it is recommended to divide the



Fig. 4. The cylinder object to be grasped in the experiments. The cylinder is 120mm tall, has an upper diameter of 30mm and a lower diameter of 50mm.

image dataset into 70 % for training, 20 % for validation and 10 % for testing in order to get a good model with avoiding overfitting and to be able to accurately evaluate the model [17]. The reason our separation of the dataset is not exactly 70/20/10 is because the test set is acquired from a separate accusation of images. This was a deliberate tactic to make sure the test set was independent of the other image sets in order to avoid bias in the detector system. All images in the dataset were acquired by separating frames from video recordings of the object, both in pool environment and at surface level. The detector was trained for 50 epochs with batch size 16 on a Nvidia GeForce RTX-2080 Ti and achieved a good mean average precision(mAP) of mAP=0.993 with intersection of union (IoU) at 0.5 and average mAP=0.854 for IoU from 0.5-0.95. The precision and recall are also satisfactory with P=0.962 and R=0.989, respectively.

B. From 2D detection to 3D target position

In order to get a better understanding of the object's position in 3D space a scaling function was used. In combination with the object detector and the manipulator's inverse kinematics, the scaling function provides the control system with a relative target position of the object without the use of any external sensors. Thus, this software solutions allows for autonomous gripping using only sensors already integrated in the manipulator system.

The scaling method is inspired by [18] and uses a mathematical approach, where the intrinsic parameters of the camera are involved. Moreover, an object's position in the Cartesian space are calculated by

$$\begin{aligned} x &= (W/w_{bb} + H/h_{bb})f/2 \\ y &= x(px_y - px_{y_0})dy \\ z &= x(px_z - px_{z_0})dz \end{aligned} \quad (10)$$

where W and H are the known width and height of the object, w_{bb} and h_{bb} are the width and height of the bounding box identifying the detection and f is the focal length of the camera. In this setup we assume that W and H are always align with the bounding box w_{bb} and h_{bb} , meaning we assume that the object is in a standing position. (px_y, px_z) represents the pixel position in y- and z-direction of the object's center, while (px_{y_0}, px_{z_0}) represents the pixel position of the center position of the image frame. The parameters



Fig. 5. Experimental setup of manipulator and object out of sight in the pool.

dy and dz are representations of the physical dimensions of each pixel in the y - and z -direction in the image plane, respectively, and are calculated by

$$\begin{aligned} dy &= \frac{2x \tan(FOV_y/2)^{-1}}{PX_y} \\ dz &= \frac{2x \tan(FOV_z/2)^{-1}}{PX_z} \end{aligned} \quad (11)$$

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

V. TESTING PROCEDURE AND SETUP

The manipulator is mounted to a fixed base plate and placed in the pool at 1.5 meters in the MC-lab at NTNU mclab, and the setup can be seen in Fig. 5. In the individual trials, the object is placed either on a pedestal or on bottom of the pool, within reach of the arm, but not within sight. The YOLOv5 small version was employed for object detection, which managed to run in real-time using only the CPU on a Dell Latitude 7490 with Intel Core i7-8650 1.90Ghz processor.

A. Experimental procedure

The experimental procedure follows a "minimal" approach where obstacle avoidance is omitted and we assume there is no occlusion of objects. In addition we assume the object is always placed in a vertical standing position, meaning \hat{q}_4 will be known through readings of q_4 and the desired rotation of the gripper at 90° in Grip mode. This assumption was made to be able to solve the inverse kinematic in (7) with three unknowns $[\hat{q}_1, \hat{q}_2, \hat{q}_3]$ given three desired velocities $[\hat{x}, \hat{y}, \hat{z}]$. The experimental procedure follows the steps given below.

- 0) **Calibrate:** Match the origin of the respective coordinate frames with the servos configuration. Manual procedure where the manipulator is set in origin (see Fig. 3) and all angles are set to 0 in this position.
- 1) **Search mode:** Using the end effector camera to search for an object. If no object is detected the arm will



Fig. 6. Manipulator in Search mode.

follow a pre-determined path by move around its own base axis until it detects an object.

- 2) **Track mode:** The manipulator detects an object and will both place it in the center of the camera image and start tracking it by moving towards it.
- 3) **Grip mode:** If the object is within reach, the manipulator will start the gripping procedure. When the gripping procedure starts, the target position is saved as a static position in the reference frame.
- 4) **Extract mode:** The manipulator will move back to base position.

B. Online Planning and Waypoint navigation

To increase the robustness of the autonomous gripping system, a simple online planner and a waypoint navigation procedure were developed and implemented. The purpose of the planner is to generate waypoints such that the manipulator can reach the desired gripping position in a smooth and safe manner. Undesirable positions such as directly under/over the object or through/behind the object can then be avoided, and the velocities and aggressiveness of the manipulator can be adjusted and specified for different steps of the gripping procedure. Once a waypoint is reached, the planner updates the desired waypoint according to the mission. A waypoint is defined as

$$WP = \{\sigma_d, \theta_5, \gamma\}$$

where σ_d is the desired position of the end effector in Cartesian coordinates, θ_5 is the opening angle of the gripper and γ determines the gain in (9).

VI. RESULTS

This section presents the experimental testing results. Fig. 6 depicts one of the tests where the manipulator is in searching mode and Fig. 7 illustrates how the manipulator perceives the detected object.

In all experiments the object was placed at a position $[x_1, y_1, z_1]$ relative to the manipulator base. The manipulator starting position was set randomly for every trial and the object position was alternated between the pool floor and on top of a structure as illustrated in Fig. 7. This ensured randomized and distinctive relative starting positions between the end effector and the object. In total, 20 autonomous fixed-base gripping tests were conducted, where the manipulator

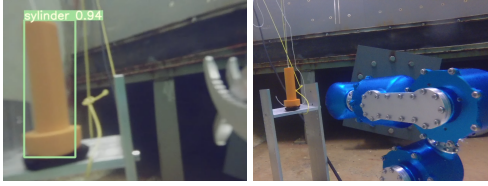


Fig. 7. Detected object during experiments from manipulator camera and external camera.

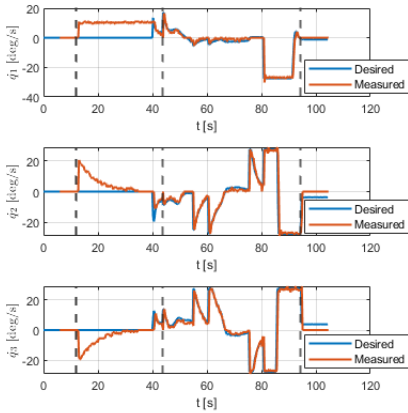


Fig. 8. Desired joint velocities from controller and real joint velocities for one of the tests.

was able to detect the object in all of the 20 tests and managed to grasp the object in 17 of the cases, yielding a grasping success rate of 85%. The desired joint velocities given by the inverse kinematics controller and the real joint velocities from one of the tests is presented in Fig. 8 and the target position and end effector position from the same test is presented in Fig. 9.

The first vertical dotted line in the plots represent when search mode was initiated and the second line represents when grip mode was initialised. The final dotted line represents the time instance when the experimental procedure was finished and the manipulator had successfully extracted the object to the desired position. It can be seen that joint 1 holds a constant velocity when in searching mode in order to search around its own base. Velocities in joints 2 and 3 in this mode represent the manipulator rising from base position to searching position. Note that the blue line describing the desired joint velocities from the controller represent the desired joint velocities for track mode and grip mode and are therefore zero until around 40 seconds when the object is first detected and track mode is initialized. Moreover, the desired velocities are zero until an object is detected because they are defined using the relative position between the object

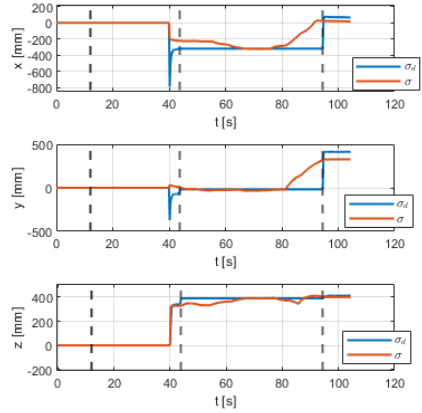


Fig. 9. Target position and relative end effector position for one of the tests.

TABLE III
RMSE FOR JOINT VELOCITIES [DEG/S] OVER ALL 20 TRIALS.

	\dot{q}_1	\dot{q}_2	\dot{q}_3	Total
RMSE	2.63	4.15	3.86	3.55

and the end-effector.

The root mean square error (RMSE) for all tests are listed in Table III. The RMSE values are retrieved from when an object is detected and the desired joint velocities are activated. Moreover this includes the (2) track, (3) grip and (4) extract steps from the experimental procedure.

VII. DISCUSSION & CONCLUSIONS

In this work autonomous grasping of an object was conducted using a manipulator with an integrated camera near the end effector and performing object detection with a detector trained with machine learning. A total of 20 trials were conducted and results from experimental testing showed that the manipulator was able to detect the object in all trials, and successfully grasp the object in 17/20 trials yielding a successful grasping rate of 85%.

In the first tests, some parameters were corrected during trials after an error was detected after test 3. Test 2 and 3 was unsuccessful and the reason was identified to be inaccurate calibration of the origin combined with small open angle of the gripper. The positional readings from the joints of the manipulator reset when the power is disconnected, and the manipulator needs to be manually put in origin with the help of a joystick controller and origin reset. During the first 3 tests it was noticed that this calibration was poorly conducted, which in combination with a small gripper opening angle lead to poor initial grasping results. The gripper angle was set to open with a distance of 5cm between each finger tip of the gripper, which corresponds to approximately 1.5 times the width of the object. With a poor calibration of the origin and hence flawed self-awareness,

the desired accuracy was hard to achieve. After the first 3 trials the origin was re-calibrated and the gripper opening angle was corrected to 8 cm, which corresponds to 2.5 times the width of the object. With these corrected parameters the manipulator managed 16/17 of the remaining trials. The initial 3 trials were still included in the overall success rate in order to highlight the system's sensitivity to calibration.

The object detection and position estimation was tested for different YOLOv5 detectors, more specifically the small, medium, large and extra large version. More complex algorithm leads to more accurate detection at the cost of more computational power. The gripping procedure ran on a laptop with CPU power, thus computational power was a limitation. A separate model for all different versions were trained and tested and it was found that the detection capabilities and the position estimation accuracy were nearly unaffected by the choice of model. Consequently, the small version was selected due to the limitations in computational power.

As stated in the experimental procedure, the target position was considered to be stationary when initializing *Step 3) Grip mode*. Moreover, when the manipulator moves closer towards the object to grasp it, the object becomes too close to the camera to perform a complete detection with acceptable position estimate. Thus, the gripping procedure rather remembers the position of the object from the last acceptable detection instead of relying on flawed or incomplete detections. An alternative solution could be to still use the detection, although perhaps a more unreliable detection, as a measurement in a Kalman filter. This could increase the robustness of the whole procedure and could correct changes in object position during gripping due to collision with other objects or the gripper itself. In order to implement this however, the detector should be retrained and strengthened to understand the difference of complete and incomplete detections.

The manipulator has limited working space and reach where objects can be gripped. However, the experiments conducted in this work is fundamental for further development where the manipulator can eventually be placed on an underwater vehicle. The searching can then be performed by the vehicle itself, and if the object is out of reach, the vehicle can move towards the object and place itself in a position where grasping of the object can be achieved. If the grasping is unsuccessful the vehicle can circle the object and move to another position of gripping before attempting again.

VIII. FUTURE WORK

Future work involves mounting the SeaArm-2 manipulator on an underwater vehicle and carry out missions that performs search, detection, tracking and grasping of objects in the water column autonomously. In addition we want to implement torque control and compare to our current velocity control approach.

IX. ACKNOWLEDGEMENT

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, T. B. Gjersvik, A. A. Transeth, and I. B. Utne, "Next generation subsea inspection, maintenance and repair operations," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 434–439, 2016. 10th IFAC Conference on Control Applications in Marine Systems/CAMS 2016.
- [2] E. Simetti, "Autonomous underwater intervention," *Current Robotics Reports*, vol. 1, pp. 117–122, 2020.
- [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] Z. Zhang, C. Wang, Q. Zhang, Y. Li, X. Feng, and Y. Wang, "Research on autonomous grasping control of underwater manipulator based on visual servo," in *2019 Chinese Automation Congress (CAC)*, pp. 2904–2910, 2019.
- [5] A. Peñalver, J. J. Fernández, and P. J. Sanz, "Autonomous underwater grasping using multi-view laser reconstruction," in *OCEANS 2017 - Aberdeen*, pp. 1–5, 2017.
- [6] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions auvs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009. Autonomous Underwater Vehicles.
- [7] E. Simetti, G. Casalino, S. Torelli, A. Sperindé, and A. Turetta, "Floating underwater manipulation: Developed control methodology and experimental validation within the trident project," *Journal of Field Robotics*, vol. 31, no. 3, pp. 364–385, 2014.
- [8] E. Simetti, F. Wanderlingh, S. Torelli, M. Bibuli, A. Odetti, G. Bruzzone, D. L. Rizzini, J. Aleotti, G. Palli, L. Moriello, and U. Scarcia, "Autonomous underwater intervention: Experimental results of the maris project," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 3, pp. 620–639, 2018.
- [9] 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, and P. Letier, "Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles," *IEEE Robotics Automation Magazine*, vol. 25, no. 4, pp. 24–33, 2018.
- [10] M. Cai, Y. Wang, S. Wang, R. Wang, Y. Ren, and M. Tan, "Grasping marine products with hybrid-driven underwater vehicle-manipulator system," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1443–1454, 2020.
- [11] D. L. Xi Q., Rauschenbach T., "Review of underwater machine vision technology and its applications," *Mar. Technol. Soc. J.*, vol. 51 (1), pp. 75–97, 2017.
- [12] O. A. N. Eidsvik, B. O. Arnesen, and I. Schjølberg, "Seaarm-a subsea multi-degree of freedom manipulator for small observation class remotely operated vehicles," in *2018 European Control Conference (ECC)*, pp. 983–990, 2018.
- [13] 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.
- [14] 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.
- [15] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, A. Hogan, lorenzomamma, tkianai, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Hatovix, J. Poznanski, L. Yu, changyu98, P. Rai, R. Ferriday, T. Sullivan, W. Xinyu, YuriRibeiro, E. R. Claramunt, hypesala, pritul dave, and zychen, "ultralytics/yolov5: v3.0," Aug. 2020.
- [16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 04 2018.
- [17] Jacob, "The train, validation, test split and why you need it," Sep 2020.
- [18] A. Makarov, V. Lukić, and O. Rahnama, "Distance and speed measurements from monocular images," in *Real-Time Image and Video Processing 2016* (N. Kehtarnavaz and M. F. Carlsson, eds.), vol. 9897, pp. 130–140, International Society for Optics and Photonics, SPIE, 2016.

Article 5

Unsupervised Domain Transfer for Task Automation in Unmanned Underwater Vehicle Intervention Operations

Albert Sans-Muntadas, **Martin B. Skaldebø**, Mikkel C. Nielsen,
Ingrid Schjøberg
IEEE Journal of Oceanic Engineering
[doi: 10.1109/JOE.2021.3126016](https://doi.org/10.1109/JOE.2021.3126016)

This article is not included due to copyright

Article 6

Autonomous underwater grasping using a novel vision-based distance estimator

Martin B. Skaldebø, Bent O. A. Haugaløkken, Ingrid Schjølberg
Accepted for publication in International Journal of Mechanical Engineering
and Robotics Research (December, 2022)

Autonomous underwater grasping using a novel vision-based distance estimator

Martin B. Skaldebø^{a,*}, Bent O. A. Haugaløkken^b and Ingrid Schjølberg^c

^aDepartment of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway

^bDepartment of Seafood Technology, SINTEF Ocean, Trondheim, Norway

^cFaculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Trondheim, Norway

ARTICLE INFO

Keywords:

object tracking, underwater manipulator, monocular vision, autonomous intervention

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.

1. 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].

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

*Corresponding author

martin.b.skaldebo@ntnu.no (M.B. Skaldebø);

bent.haugalokken@sintef.no (B.O.A. Haugaløkken);

ingrid.schjolberg@ntnu.no (I. Schjølberg)

ORCID(s): 0000-0001-8339-8066 (M.B. Skaldebø);

0000-0003-0271-531X (B.O.A. Haugaløkken); 0000-0002-8750-0301 (I. Schjølberg)

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.

2. 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. [12] developed a conditional generative adversarial network (GAN) for real-time underwater image enhancement. 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, [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 [20] conducted reinforcement learning for mobile autonomous grasping with a robot on land, and [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 ice cream. 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 of 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).

3. Object Detection 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 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.

3.1. 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 developed 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.

3.2. 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.

3.3. Computer Vision Framework

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

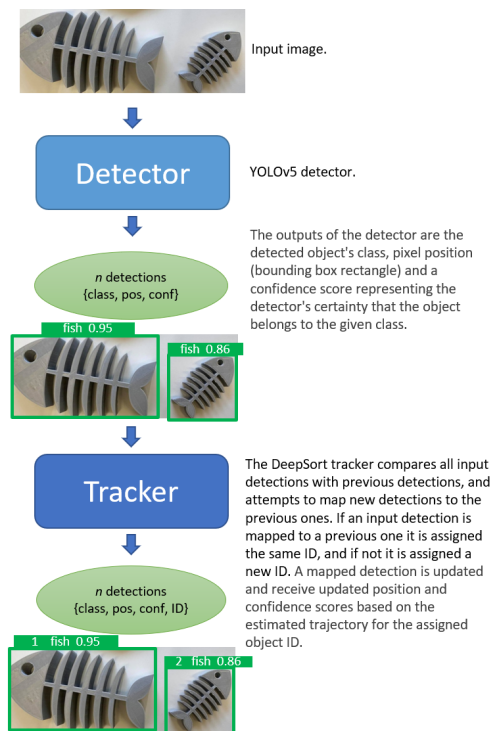


Figure 1: Outline of the computer vision framework with YOLOv5 detector and DeepSort tracker.

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, [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.

4. 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.

4.1. 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:

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

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). Equation (1) is derived from Figure 2, where the focal lengths $f = f_x = f_y$ represents the distance from the origin O_c to the principal point (px_{y_0}, px_{z_0}) 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 (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

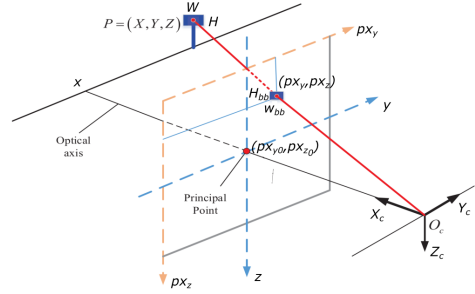


Figure 2: Illustration of domain transformation for an image and corresponding 3D scene, inspired by [37].

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:

$$x_j - x_i = \Delta x_{ij} = (W/w_{bb_j} - W/w_{bb_i})f \quad (2)$$

and the height:

$$x_j - x_i = \Delta x_{ij} = (H/h_{bb_j} - H/h_{bb_i})f, \quad (3)$$

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 x_i and x_j 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 (2) and (3), it is possible to estimate the width and height of the object as follows:

$$\hat{W} = \frac{w_{bb_j} w_{bb_i} \Delta x_{ij}}{f(w_{bb_i} - w_{bb_j})}, \quad (4)$$

$$\hat{H} = \frac{h_{bb_j} h_{bb_i} \Delta x_{ij}}{f(h_{bb_i} - h_{bb_j})}. \quad (5)$$

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

4.2. Multiple Measurements

The distance estimator exploits multiple measurements to increase the accuracy of the object size estimation. To calculate \hat{W} and \hat{H} from (4) and (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 \hat{W} and \hat{H} with every combination of two measurements, meaning an update of the \hat{W} and \hat{H} from (4) and (5) to \hat{W}_{ij} and \hat{H}_{ij} . The final estimates then become:

$$\hat{W} = \frac{1}{n_c} \sum_i \sum_{j \neq i} \hat{W}_{ij} \quad (6)$$

$$\hat{H} = \frac{1}{n_c} \sum_i \sum_{j \neq i} \hat{H}_{ij} \quad (7)$$

where \hat{W}_{ij} and \hat{H}_{ij} are calculated from (4) and (5) and n_c is the number of possible combinations between samples. n_c can be found as follows:

$$n_c = \frac{n_s^2 - n_s}{2}, \quad (8)$$

where n_s is the number of samples.

4.3. 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:

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

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:

$$y = x(px_y - px_{y_0})dy \quad (10)$$

$$z = x(px_z - px_{z_0})dz, \quad (11)$$

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:

$$dy = \frac{2x \tan(FOV_y/2)^{-1}}{PX_y} \quad (12)$$

$$dz = \frac{2x \tan(FOV_z/2)^{-1}}{PX_z}, \quad (13)$$

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.

4.4. 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

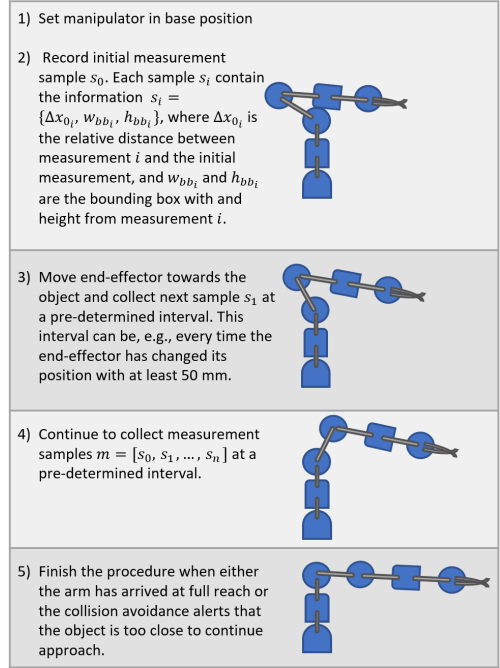


Figure 3: Outline of distance estimator procedure.

threshold. The lower-level procedure follows the steps outlined in Figure 3.

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 (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 (6)-(7). The higher-level control procedure also ensures that the manipulator stops if it moves too close to the object.

5. 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.

Table 1
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 HD USB camera

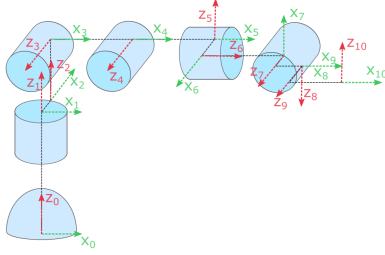


Figure 4: Coordinate axis system of manipulator.

5.1. 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 1.

5.1.1. 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]. Figure 4 illustrates the manipulator's coordinate axis system; the corresponding DH parameters are listed in Table 2. Note that these parameters represent the manipulator configuration as assembled in Figure 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° 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.

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

Table 2
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	$q_5/2$	40	$-\pi/2$

$$\mathbf{T}_{i-1}^i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & ac\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & as\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (14)$$

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

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

This results in a very comprehensive matrix, which is not shown here due to space constraints. The \mathbf{T}_0^8 and \mathbf{T}_0^{10} 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:

$$\begin{bmatrix} \sigma_{(b)} \\ 1 \end{bmatrix} = \mathbf{T}_{(b)}^{(ee)} \begin{bmatrix} \sigma_{(ee)} \\ 1 \end{bmatrix}, \quad (16)$$

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

5.1.2. 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:

$$\dot{\sigma}_\chi = \mathbf{J}_\chi(\mathbf{q})\dot{\mathbf{q}}, \quad (17)$$

where the value χ represents the task. The tasks considered in this paper are (1) manipulator control with camera and \mathbf{T}_0^8 as a reference system and (2) manipulator gripper control with the gripper and \mathbf{T}_0^{10} as a reference system. The task variables σ_χ and \mathbf{J}_χ are determined by the task-specific transformation matrix along with (16) in Section 5.1.1 and the DH parameters (Table 2), respectively. Moreover, to determine the reference joint velocities based on the reference Cartesian velocities, the pseudo-inverse of the Jacobian, \mathbf{J}^\dagger , is used:

$$\dot{\mathbf{q}}_r = \mathbf{J}_\chi^\dagger(\mathbf{q})\dot{\sigma}_{\chi,r}, \quad (18)$$

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

$$\mathbf{J}_\chi^\dagger = \mathbf{J}_\chi^T(\mathbf{J}_\chi\mathbf{J}_\chi^T + \lambda\mathbf{I})^{-1}, \quad (19)$$

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 $\dot{\sigma}_{\chi,r}$ are determined by:

$$\dot{\sigma}_{\chi,r} = \gamma_\chi(\dot{\sigma}_{\chi,d} - \dot{\sigma}_\chi). \quad (20)$$

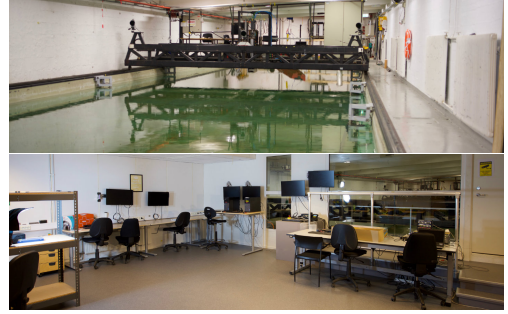
Here, γ_χ is the task-specific gain, and $\dot{\sigma}_{\chi,d}$ and $\dot{\sigma}_\chi$ are the task-specific desired and measured values for $\dot{\sigma}$, respectively.

5.2. Laboratory Setup

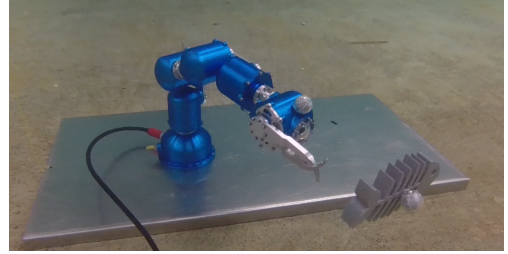
The experiments were conducted in the Marine Cybernetics laboratory (MC-lab) at NTNU [40]. The laboratory facility consists of a pool and control room, depicted in Figure 5a. In the experiments, the manipulator was attached to a steel plate that was lowered to the bottom of the pool. The laboratory setup is depicted in Figure 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.

6. 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



(a) MC-lab. Above: Laboratory pool with dimensions 40m × 6.45m × 1.5m. 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.

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.

6.1. 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 3, and their 3D positions in space were estimated using the distance estimator described in Section 4. 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 Figure 6a. In this experiment, the system was able to quite accurately estimate the relative

Table 3
Distance estimator RMSE values for case study 1

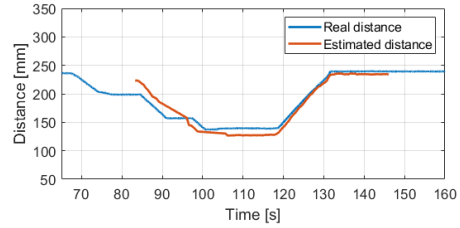
RMSE fish	RMSE skeleton	RMSE total
124.24 mm	26.81 mm	75.53 mm

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).

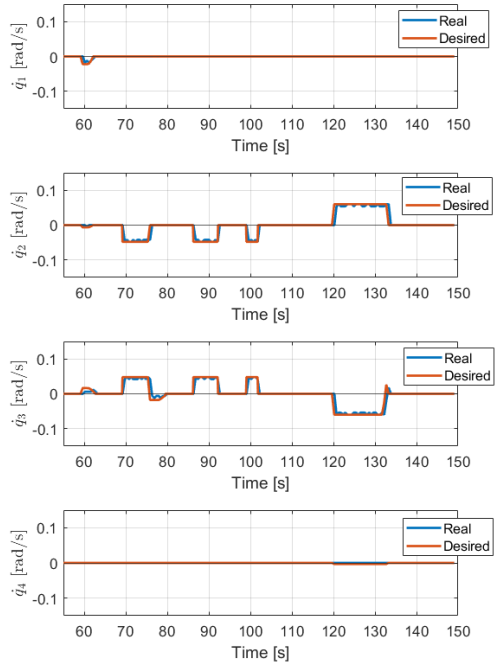
The joint velocities for the same experiment can be seen in Figure 6b. The manipulator's return to base position can also be seen in these velocity plots at approximately 120 seconds, where the joint velocities for joints 2 and 3 are high values with opposite signs. The velocity plots also illustrate 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 Figure 3.

The two objects used in the experiments are depicted in Figure 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.

The distance estimator's root mean square error (RMSE) was calculated and is listed in Table 3 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 Figure 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.



(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.

6.2. Case Study 2: Autonomous Grasping

In this case study, the manipulator attempted to autonomously grasp an object following the experimental procedure explained in Table 4. 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

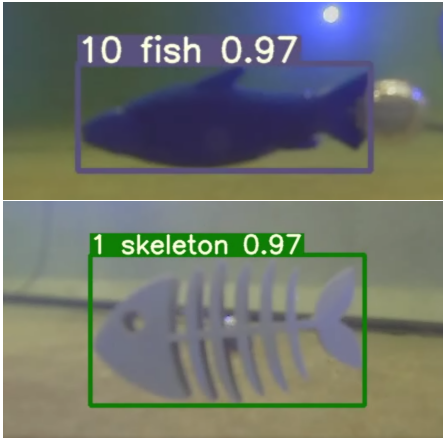


Figure 7: Detection of fish and skeleton objects used in experiments with attached Qualisys markers. The labels atop the bounding boxes represent tracker ID, object class, and detection confidence.

Table 4

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.

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.

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 5. Each trial started with the manipulator at a unique starting position, with the object in different positions, and with different relative

Table 5

Distance estimator RMSEs for case study 2

RMSE fish	RMSE skeleton	RMSE total
8.96 mm	10.47 mm	9.21 mm

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 Figure 8, demonstrating the grasping and initial retrieval of a fish object. Figure 9a plots the desired and real joint velocities for the same grasping experiment. The modes defined in the experimental procedure in Table 4 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 Figure 8, where the object is placed horizontally, resulting in an optimal rotation angle of $q_4 = 0$.

End-effector velocities in Cartesian coordinates are presented in Figure 9b. 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.

7. 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

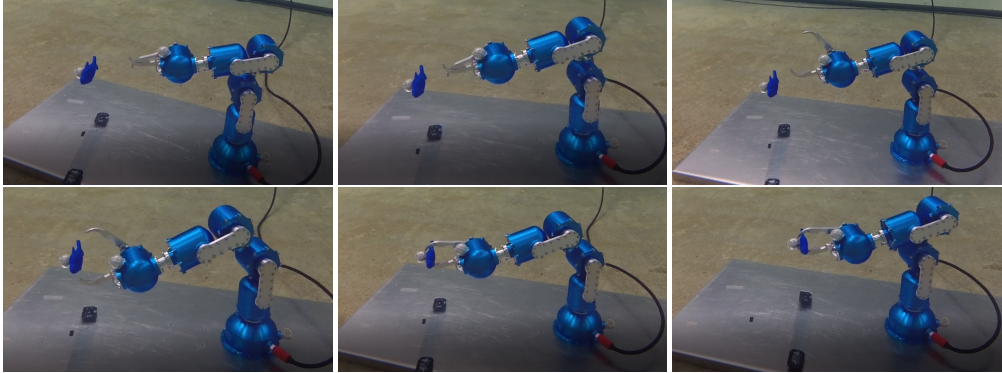


Figure 8: Grasping sequence.

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 Figures 10a and 10b. The plot in Figure 10a 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 overshoot in its estimates by up to 200 mm, which is insufficient accuracy for grasping. For the experiment in Figure 10b, 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 overshoot by approximately 150 mm.

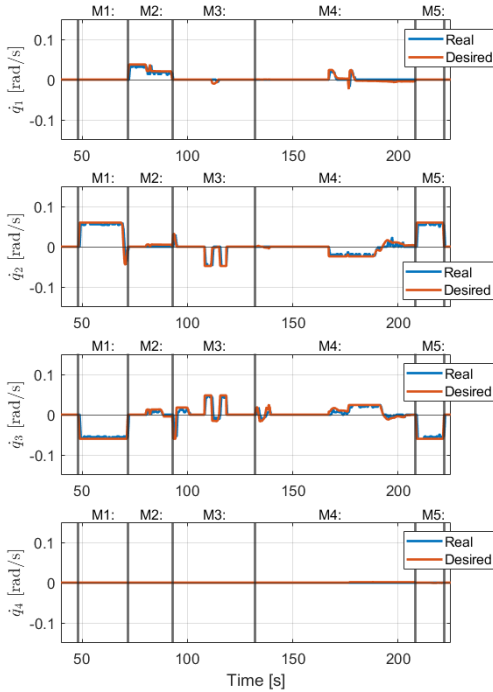
Another minor challenge discovered in the computer vision framework was the latency introduced by the tracker. In search mode, the system naturally only discovers parts of the object at first, before the entire object is within the field of view. The tracker constantly attempts to retain a stable detection, 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

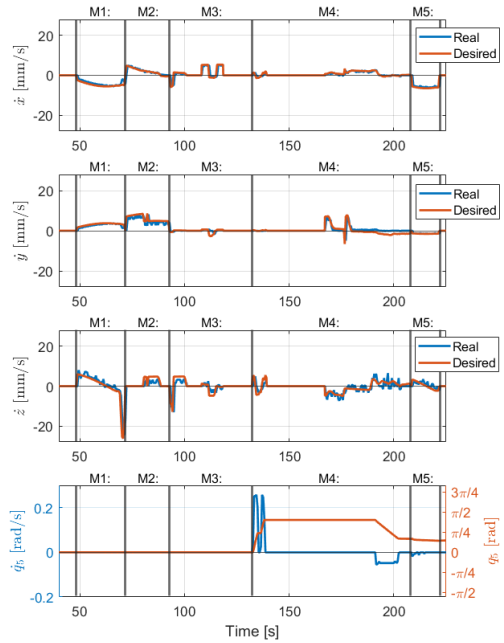
measure of something unknown when there are limited data with which to compare. However, the system designed in this paper gathers new measures whenever practical. Hence, there is a potential to filter out inaccurate measurements. One way to do this could be to only accept measurements that do not differ 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 inaccurate 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.

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.

Autonomous grasping using novel distance estimator



(a) Joint velocities for a grasping experiment 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.



(b) Plots 1-3 show Cartesian velocities for end-effector for a grasping experiment 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.

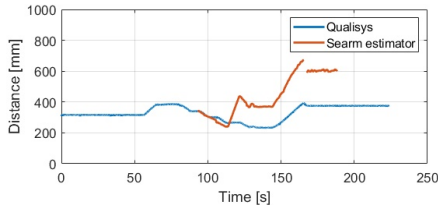
Figure 9: Joint velocities and Cartesian end-effector velocities for a the grasping experiment depicted in Figure 8.

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.

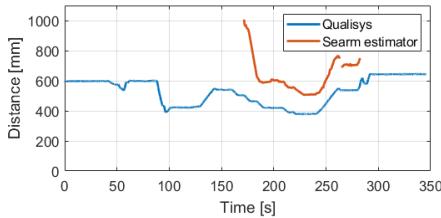
8. Conclusions

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.



(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 10: Estimated and measured relative distance between manipulator and object for two experimental trials that yielded high RMSE values.

9. Acknowledgments

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," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 183–188, 2015. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV 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 *OCEANS 2015-MTS/IEEE Washington*, pp. 1–10, IEEE, 2015.
- [5] E. Simetti, "Autonomous underwater intervention," *Current Robotics Reports*, vol. 1, pp. 117–122, 2020.
- [6] M. B. Skaldebø, B. O. A. Haugalokken, and I. Schjølberg, "Seaarm-2 - fully electric underwater manipulator with integrated end-effector camera," in *2021 European Control Conference (ECC)*, pp. 236–242, 2021.
- [7] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions auvs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009. Autonomous Underwater Vehicles.
- [8] F. Bonin-Font, G. Oliver, S. Wirth, M. Massot, P. Lluís 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 *OCEANS 2019 - Marseille*, pp. 1–8, 2019.
- [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 *2021 IEEE International Conference on Mechatronics (ICM)*, pp. 1–7, 2021.
- [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 *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2955–2962, 2012.
- [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," in *Journal of Ocean Engineering and Technology*, vol. 34, p. 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, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, A. Hogan, lorenzomamma, tkianai, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Hatovix, J. Poznanski, L. Yu, changyu98, P. Rai, R. Ferriday, T. Sullivan, W. Xinyu, YuriRibeiro, E. R. Claramunt, hopesala, pritul dave, and yzchen, "ultralytics/yolov5: v3.0," Aug. 2020.
- [31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.
- [32] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.
- [33] B. Sekachev, N. Manovich, M. Zhiltsov, A. Zhavoronkov, D. Kalinin, B. Hoff, TOSmanov, D. Kruchinin, A. Zankevich, DmitriySidnev, M. Markelov, Johannes222, M. Chenuet, a andre, telenachos, A. Melnikov, J. Kim, L. Ilouz, N. Glazov, Priya4607, R. Tehrani, S. Jeong, V. Skubriev, S. Yonekura, vugia truong, zliang7, lizhming, and T. Truong, "opencv/cvat: v1.1.0," Aug. 2020.
- [34] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017.
- [35] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016.
- [36] M. Broström, "Real-time multi-object tracker using yolov5 and deep sort." https://github.com/mikel-brostrom/Yolov5_DeepSort_Pytorch, 2020.
- [37] A. Makarov, V. Lukić, and O. Rahnama, "Distance and speed measurements from monocular images," in *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.
- [38] 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.
- [39] 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.
- [40] NTNU, "Marine cybernetics teaching laboratory." Online, 11 2020.

Article 7

Underwater vehicle manipulator system (UVMS) with BlueROV2 and SeaArm-2 manipulator

Martin B. Skaldebø, Ingrid Schjøberg, Bent O. A. Haugaløkken
41st International Conference on Ocean, Offshore & Arctic Engineering
(OMAE 2022), Hamburg
[doi: 10.1115/OMAE2022-79913](https://doi.org/10.1115/OMAE2022-79913)

OMAE2022-79913

**UNDERWATER VEHICLE MANIPULATOR SYSTEM (UVMS) WITH BLUEROV2 AND
SEAARM-2 MANIPULATOR**

Martin Skaldebø*

Department of Marine Technology
Norwegian University of Science and
Technology (NTNU)
7052 Trondheim, Norway
Email: martin.b.skaldebo@ntnu.no

Ingrid Schjølberg

Department of Marine Technology
Norwegian University of Science and
Technology (NTNU)
7052 Trondheim, Norway
Email: ingrid.schjolberg@ntnu.no

Bent O. A. Haugaløkken

Sintef Ocean
Trondheim, Norway
Email: bent.haugalokken@sintef.no

ABSTRACT

This work presents a novel digital twin for an underwater vehicle manipulator system (UVMS). The twin represents a framework for low cost UVMS and builds on the BlueROV2 and the SeaArm-2 manipulator. UVMS's are currently applied in ocean interventions and the presented platform enables testing and verification of intervention methods in subsea inspection, maintenance and repair operations. The UVMS twin represents an excellent research platform for such systems as it is low cost with highly accessible and customizable software. The paper further presents a platform for the UVMS where the robotic system is modeled in the Gazebo simulator. The Marine Cybernetics laboratory at NTNU is modeled and the UVMS can be deployed into this environment for testing before deployed in real world experiments. The twin represents a simulator which offers a base where overall system, control system, communication, etc. can be tested before progressing to real world experiments. The low cost of the equipment is essential to perform experiments and tests of subsea intervention efficiently and at a sustainable cost.

*Address all correspondence to this author.

INTRODUCTION

Unmanned underwater vehicles (UUVs) have replaced human divers in most underwater intervention and operations. A spectrum of versatile UUVs are employed to a variety of inspection, maintenance, and repair (IMR) operations [1], e.g. on oil cables, fish cages, underwater control sites, ocean wind sites [2, 3]. IMR operations are mostly performed by remotely controlled vehicles (ROVs), where the operator has manual control over the vehicle. Other underwater operations such as ocean mapping, collecting ocean samples, under ice exploration are often performed by autonomous underwater vehicles (AUVs) where the vehicle operates independently without interference by an operator. AUVs are untethered and communication and information between vehicle and operator are not regular, meaning the vehicle must be able to perform autonomously while operating underwater.

Autonomy can be interpreted in different levels of autonomy, and defined in four levels from automatic control to highly autonomous control [4]. The motivation for segregating into different levels of autonomy derives from defining functionalities and capabilities of a system and to determine the overall risk aspects of the systems. Remote control from onshore is an in-

creasingly popular alternative to deploying support vessel and crew due to significant reduction in cost and the independence of surface weather conditions. Remote operations of underwater vehicles demands increased requirements for the overall system in terms of autonomous functionality and safety and fault management systems. Moreover, such systems are important in order to cause minimal or no harm of equipment when communication is lost or the control is in any way lost. For instance, if communication with the underwater vehicle is temporarily lost the system should be able to, autonomously, dock safely, return to surface or in any other ways continue mission or avoid being lost or damaged. Moreover, this relates to the levels of autonomy where the system should be able to automatically execute mission-related tasks whenever operator command is unavailable. Thus, the system does not have to be a highly autonomous system to operate sufficiently in such scenarios, however it should possess some autonomous functionalities. As mentioned, autonomous functionalities are important in safety and fault management systems. However, it could also bring huge benefits in increasing safety, accuracy and efficiency in other aspects of operations. Examples are proficient autonomous docking that can outperform manual docking in difficult environments, autonomous panel intervention to replace manually controlling manipulators to grasp and turn valves, autonomously grasping to locate, pick up and retrieve objects [5–8]. Increasing the level of autonomy naturally shifts the operator towards a more supervisory role. In order to retain the situational awareness during a supervisory role the operator should receive visual and sensory feedback from the vehicle during autonomous operations. Camera systems are important sensors in this aspect given the reliable and information rich data output. Continuous development of software for computer vision systems also increases the importance of camera systems as fundamental sensors in an autonomous system. Acoustic sonars are also substantially used sensors for underwater operations [9–11], however visual data poses spatial and temporal resolution far superior to acoustic signals [12].

UUVs are more frequently equipped with manipulators to improve intervention capabilities for the vehicle, referred to as underwater vehicle manipulator systems (UVMSs) [13]. The diversity in operating tasks and conditions are met with an equal diversified assembly of UVMSs. The systems range from simple small systems with 1 degree of freedom (DoF) grippers with limited depth ratings to large systems with manipulators capable of lifting several hundred kilos at depth of several thousand meters [3]. Given the large spectrum of sizes, shapes, manipulator DoF, etc. of UVMSs, they are very versatile tools and, thus, are deployed in a wide range of situations, such as the oil and gas industry, aquaculture, ocean mapping, sea mining, environmental monitoring and surveillance [1, 14–16].

The main contribution of the presented work is the novel digital twin for an underwater vehicle manipulator system (UVMS). The twin is a model-based simulator including a visu-

alization module that offers a base where overall system, control system, communication, etc. can be tested before progressing to real world experiments and implementations. UVMSs have complex dynamics and there is a need for twin models for validation and testing of intervention operations. The twin builds on low cost equipment such as the BlueROV2 and the SeaArm-2 manipulator. The platform for the UVMS is the Gazebo simulator. Furthermore, an environment model is included and the UVMS can be integrated into this environment for testing before deployed in real world experiments.

SPECIFICATIONS

In this section the specifications for the twin of the UVMS applied in this work will be presented. The UVMS consists of the small commercially available ROV BlueROV2 and the underwater manipulator SeaArm-2 [17]. Interaction between the two robots and between the UVMS and operator are communicated through the Robotic Operating System (ROS) [18].

BlueROV2

The UVMS is given by the equations of motion and consist of a ROV with the attached underwater manipulator. The ROV utilized in this work is the BlueROV2 depicted in Figure 1. The ROV is setup in a heavy configuration meaning it has eight thrusters providing six DoFs control. BlueROV2 is developed by BlueRobotics [19] and is a high-performance low cost underwater ROV used by professionals as well as for leisure activities. The ROV comes with open source software and can be controlled using the open source ArduSub subsea vehicle control firmware. For the work conducted in this article the open source software is replaced by in-house software solutions. This to ensure the integration of all different components in the UVMS functions optimally. The utilized software is based on ROS, enabling safe and fast communication between the involved robots and operators [20]. The main specifications of the BlueROV2 utilized in this work is listed in Table 1.



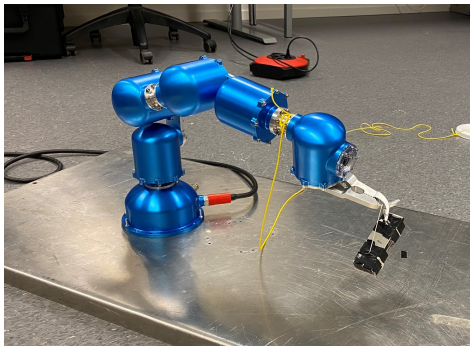
FIGURE 1. The BlueROV2 underwater vehicle

TABLE 1. BlueROV2 specifications

Parameter	Value
Weight in Air	11.5 [kg]
L × H × W	457 × 254 × 575 [mm]
Thrusters	T-200
Battery	14.8 [V], 18 [Ah]
Depth rating	100 [m]
Camera	Low-Light HD USB camera
On-board Computer	Raspberry Pi 3B and Navio2

SeaArm-2 manipulator

The SeaArm-2 is a fully electric small modular manipulator and the DoFs can be adjusted by modifying the number and configuration of modules. With the standard assembly as depicted in Figure 2 the manipulator has four DoFs, maximum reach of 693.75 mm and lifting capacity of 5 kg at full reach. A digital twin is built for the SeaArm-2 depicted in Figure 2 and the main specifications of the manipulator are listed in Table 2.

**FIGURE 2.** The SeaArm-2 manipulator.

The SeaArm-2 manipulator has an integrated monocular camera in the end-module and an on-board computer, and supports connection through both USB and Ethernet connection. When controlling the manipulator independently of other systems, power and Ethernet connections are connected through a wet-mate connector. This connection enables transfer of power from a topside power supply and data to a topside computer. The topside computer and the SeaArm-2 manipulator operate

TABLE 2. SeaArm-2 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
Camera	Low-light HD USB camera
On-board computer	Raspberry Pi 3B
Communication	RS485 and Ethernet

as a server-client communication pair and transmit User Datagram Protocol (UDP) messages for communication. The on-board computer receives and analyses messages and can distribute forces and velocities to joints, turn on/off camera feed, log and read joint positions and velocities and more. Tasks that requires more computational power such as object detection and tracking on the camera feed are performed topside at the operator computer.

As mentioned, the SeaArm-2 manipulator also supports USB connection. This connection is used when the manipulator is connected to an ROV. Both power and data are then transmitted directly from the ROV and data are transmitted between the on-board computers through the USB connection. Commands are transmitted from the surface operator computer to the on-board computer on the ROV before the relevant messages are sent to the manipulator.

The SeaArm-2 manipulator is controlled through kinematic control where the geometric relations of the manipulator are explained through the Denavit-Hartenberg (DH) convention. The DH-convention describes the geometric relations between the joints with lengths and angles and is a mathematical description of the manipulator. In order to establish the manipulator kinematics the joints are represented with a set of coordinate frames as presented in Figure 3. The corresponding DH-parameters are listed in Table 3.

The joint angles are represented by $\mathbf{q} = [q_1, q_2, q_3, q_4, q_5]$, where q_1 and q_4 have 360° continuous rotation, q_2 and q_3 have limited rotation angle due to self collision and q_5 is the gripper opening angle, as illustrated in Figure 4.

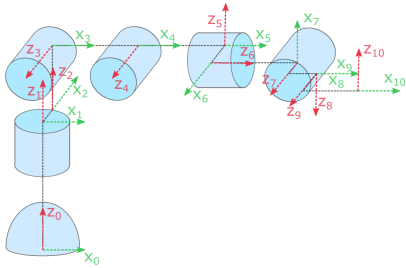


FIGURE 3. Coordinate axis system of manipulator [21].

TABLE 3. 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	$q_5/2$	40	$-\pi/2$

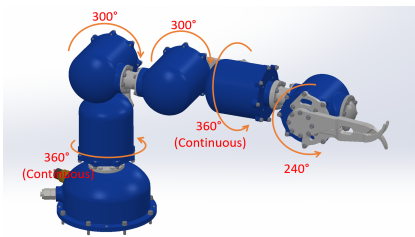


FIGURE 4. SeaArm-2 rotational limitations.

Underwater Vehicle Manipulator System

The combined UVMS will have the manipulator attached on the bottom of the ROV. The manipulator is connected to the ROV through a USB cable and controlled through the on-board computer of the manipulator. The ROV is tethered, and commu-

nication between the operator and UVMS takes place through the vehicle's tether.

DIGITAL TWIN

This section describes the digital twin consisting of a physical-based model of the UVMS represented in the simulator, the simulated environment and the real-time connection to UVMS through ROS. The simulator represents a safe perimeter for testing of control systems and other software solutions before they are applied to the real system. The simulator is a mathematical and physical-based model of the UVMS, designed from the real system in order to minimize the complexity of transferring knowledge and software solutions from simulation to real world experiments. For the simulator the open source simulator Gazebo is used where the environment and the robots included in the UVMS are modeled from 3D graphics files. The environment and UVMS are modeled in a way to represent a realistic version of the real system, and to enable systems such as control algorithms, communication protocols etc., directly transferable to real world experiments.

Gazebo simulator

The simulated environment is modeled in the Gazebo simulator, which is an open-source simulator that offers 3D robot simulation and visualization [22]. The Gazebo simulator focuses on the ability to simulate dynamics and incorporating the accuracy in robot sensors and actuators. Other systems may provide an equally good simulated environment, such as, e.g., the commercially available COSMIR system [23] developed by Festo that is designed for industrial simulation, but also applicable for robotic research. Webots [24] is another commercially available tool developed by Cyberbotics for research and development purposes. Moreover, the free of charge and open-source availability of Gazebo and directly transferable system from simulator to real world deemed it the preferred simulator for this work.

The simulator software is integrated with ROS in a way that the software running on the operator computer listens to data transmitted from the simulator instead of the hardware. Such software on the operator computer can be control systems, computer vision frameworks, motion planners, and so on. In this way the simulator replaces the hardware and subscribes to topics such as thruster forces and torques, manipulator joint velocities, joystick commands, and camera images.

Environmental model

The Marine Cybernetics Laboratory (MC-Lab) at NTNU [25] consist of a control room and a laboratory pool with dimensions $40\text{m} \times 6.45\text{m} \times 1.5\text{m}$. The pool of MC-Lab is used for experimental testing of both surface and underwater systems at NTNU. MC-Lab is therefore used as the base environment in

the simulator to represent a real world experimental platform. The laboratory pool is modeled in Gazebo as seen from Figure 5. Gravity, density of water, hard walls/floor and dimensions equal to the real world laboratory are modeled to provide a realistic representation of the real world environment. The simulator provides an excellent environment where modeled robots can be deployed for testing and verification of developed systems, such as advanced control concepts, before deployed in a real environment.

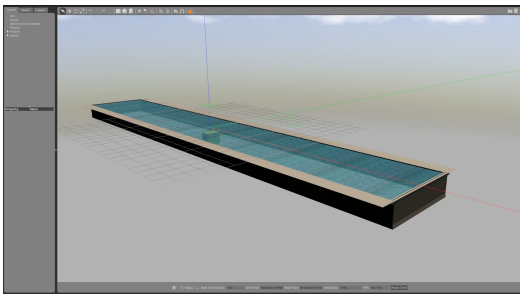


FIGURE 5. Marine cybernetics laboratory in the Gazebo simulator

BlueROV2

The modeled BlueROV2 in the Gazebo simulator is depicted in Figure 6. The BlueROV2 is put together with 3D files representing the BlueROV2 body and the thrusters. Forces and torques are applied through the thruster positions using a joystick and thruster allocation matrix. This means that an input from the joystick to move in the x-direction goes through the thruster allocation matrix, which distributes forces and torques to the relevant thrusters.

SeaArm-2 Manipulator

The modeled SeaArm-2 is depicted in Figure 7, where it is attached to a underwater panel template in the MC-lab environment. The manipulator is placed below the water surface in the figure, however the blurry visual aspects of water is neglected in order to generate a cleaner perception of the scene. The water surface, however, is constructed as was illustrated in Figure 5 with the entire pool visible. The manipulator is assembled based on 3D graphics files for the different modules, links and the gripper. Correct rotations and joint limits are part of the assembly to represent the real world most accurately.

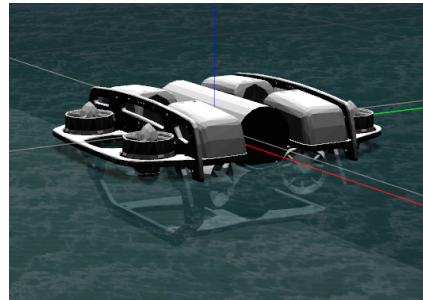


FIGURE 6. The BlueROV2 in the MC-lab environment in the Gazebo simulator.

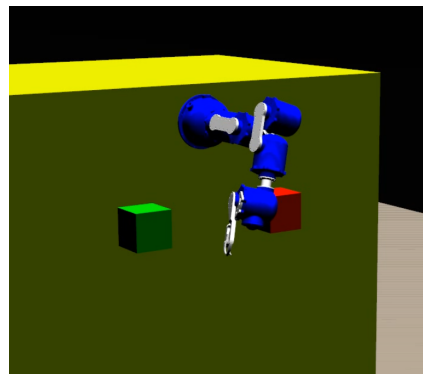


FIGURE 7. The SeaArm-2 manipulator modeled in the MC-lab environment in the Gazebo simulator

Underwater Vehicle Manipulator System

The SeaArm-2 and the BlueROV2 are modeled in Gazebo as separate robots, in order to reflect the real world where the combined UVMS are built from two individual robots. The SeaArm-2 manipulator is attached underneath the BlueROV2. The two robots can be controlled individually or as a joint system, depending only on the applied control system. The combined UVMS is depicted in Figure 8, where it is placed in the modeled MC-lab environment. From the figure it can be observed how the UVMS floats in the water surface with the top of the BlueROV2 visible above the waterline, while the rest of the UVMS is submerged.

Some properties of the two robots are difficult to identify accurately in a simulated environment, and are therefore either simplified or not considered in the current simulator model. Communication delay between operator computer and UVMS and de-

creased forces and torques effects from low battery level are not included in the simulator. Moreover, acquiring accurate inertia properties of the two robots and the combined UVMS is a demanding task, hence they are simply estimated in the simulator.

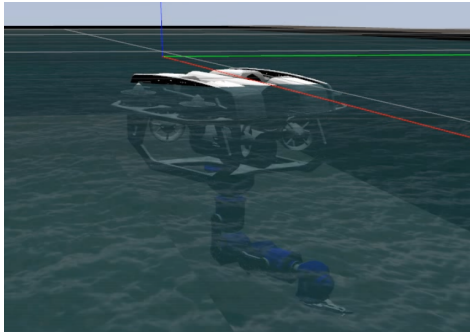


FIGURE 8. The combined UVMS with the BlueROV2 and the SeaArm-2 in the MC-lab environment in the Gazebo simulator

UVMS AND DIGITAL TWIN PERFORMANCE

The presented system is tested for real time applications and time delays. This is to ensure that the system is proficient in relevant operations. If the overall system has a significant time delay, either in the digital twin or within the control system or communication protocols of the UVMS, the digital twin would not be capable of following the real world UVMS. Moreover, this will create a discrepancy between the two resulting in a digital twin with constantly delay compared to the UVMS that diminishes the whole system.

Test procedure

The system is tested by measuring and comparing time slots for when actions are performed at different sections of the system. The time delays between the measures reveals the time delays present in the system. The system is tested by sending velocity commands from the operator computer to the UVMS. The UVMS receives the commands, then demands action to the thrusters, logs current velocities and transmits the vehicle status back to the operator computer. When the operator computer receives the log on current status of the UVMS, it is further transmitted to the digital twin where the behaviour of the UVMS is simulated.

TABLE 4. Time delay between different sections in the system.

Section	Time delay
Velocity command input to operator computer	0.000s
UVMS receive velocity command	0.007s
UVMS demands action to thrusters	0.007s
UVMS transmits status log to operator computer	0.096s
Operator computer logs status	0.107s
Digital twin receives status log	0.109s
Digital twin demands action to thrusters	0.110s

Test results

The time delays between the sections of the system is listed in Table 4. The largest time delay occurs between the UVMS demands action to the thrusters and when the UVMS transmits the log to the operator computer. Moreover, this time delay is caused by the UVMS onboard computer collecting and logging all status updates of the vehicle. However, the overall time delays in the system is negligible when it comes to real time capabilities. From the time a velocity command is initialized on the operator computer to the digital twin registers the movement it takes an average of 0.110s. A time series of the same test is depicted in Figure 9 with two plots. The first plot shows the PWM values to the gripper over time. Moreover the plot includes 3 lines representing when the PWM signal is registered as input on the operator computer, when the physical UVMS demands the action to the thrusters and when the digital twin demands the action to the simulator. Unlike the other joints of the manipulator, the gripper is controlled with PWM signal instead of desired velocity. This enables the gripper to grasp objects with a desired force instead of forcing a velocity when the gripper is closed. The second plot in Figure 9 shows the physical UVMS gripper angle and the gripper angle in the simulator over time. The digital twin follows the physical system with a time delay of approximately 0.105s equal to the results from Figure 4. The overall time delay in the system does not affect the real time applications of the system, and the digital twin is capable of representing the UVMS in a sufficient manner.

DISCUSSION

The presented UVMS is a low cost system designed for research and development. The system utilizes the BlueROV2 underwater vehicle and the SeaArm-2 manipulator, which both can be categorized as small, low cost and versatile within their respective segments. The combined robots result in a UVMS tailored for exploring new and innovative solutions for a wide range of industries involving the underwater scene, such as the

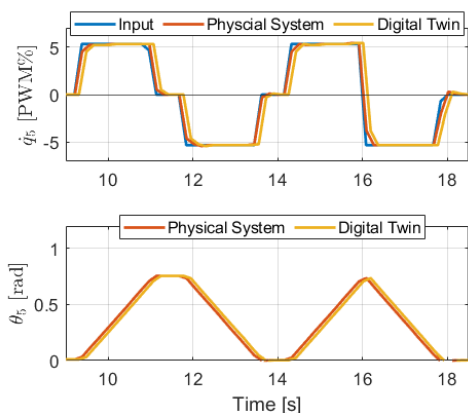


FIGURE 9. Top: Manipulator gripper velocity over time. Bottom: Manipulator gripper angle over time

oil and gas industry, aquaculture, ocean mapping, sea mining, environmental monitoring and surveillance. Due to the low cost of the system there are physical restrictions of the systems depths, lifting capabilities, and overall robustness. However, state-of-the-art software solutions tested and verified on this system can be transferred to more expensive systems once they are demonstrated to work. Such system may include work class ROVs or industrial arms that would require a significant investment. The flexible system also allows for optimization of hardware on DoFs, thruster power, manipulator capabilities, change of sensors, and so on, at low additional costs. This ensures that the end-product can be an optimized product, and that the way to the desired end-product can be cheaper than doing the same iterations with equipment ten times the cost.

The digital twin provides an excellent platform for initial testing of the UVMS where validating control algorithms, visual presentation and system capabilities can be performed before deployed in a real environment. Moreover, the analysis and results from the simulator should be considered with some restraints compared to real world experiments. The hydrodynamic coefficients for the SeaArm-2 are not accurately calculated and for the digital twin these are only approximate estimations [13]. However, the estimates will give a realistic behavior, which should be investigated more closely in experimental testing. This will further strengthen and verify the performance of the digital twin.

Further analysis of the UVMS system and identification of hydrodynamic coefficients and reacting forces between the manipulator and ROV can be implemented in the digital twin and result in a improved twin. Moreover, at this stage the digital twin is a great tool for initial testing and verification, and can pro-

vide results close to reality when it comes to dynamic behaviour. The Gazebo simulator provides functionalities for such analysis and by identifying hydrodynamic coefficients and implementing them in the simulator, the digital twin will be an even better tool. Nonetheless, it is important to notice that however proficient the simulator becomes it can not replace physical experiments, where unidentified responses and hardware communication can lead to problems not occurring in the simulator.

CONCLUSIONS AND FURTHER WORK

This paper has presented a novel digital twin for an underwater vehicle manipulator system (UVMS). The twin represents a low cost UVMS and builds on mathematical models and communication infrastructure for BlueROV2 and the SeaArm-2 manipulator. The presented UVMS digital twin represents an excellent research platform for such systems as it is low cost with highly accessible and customizable software. The Marine Cybernetics laboratory at NTNU is modeled and the UVMS can be deployed into this environment for testing before deployed in real world experiments. This helps lower the cost of experiments by minimizing time spent on software bug fixes and tuning and allocation troubles before entering the laboratory. In addition it may increase safety for both personnel and equipment by ensuring the system works before running it on real hardware. In future work the UVMS will be deployed in the Marine Cybernetics Laboratory for experimental testing in autonomous object search, grasp and retrieval missions.

ACKNOWLEDGMENT

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]. The authors would like to thank Mikkel Cornelius Nielsen and Albert Sans Muntadas for support in designing the simulator environment.

REFERENCES

- [1] Schjøberg, I., and Utne, I. B., 2015. "Towards autonomy in roV operations". *IFAC-PapersOnLine*, **48**(2), pp. 183–188. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV 2015.
- [2] Christ, R. D., and Wernli, R. K., 2014. *The ROV Manual*. Elsevier Ltd.
- [3] Sivčev, S., Coleman, J., Omerdić, E., Dooly, G., and Toal, D., 2018. "Underwater manipulators: A review". *Ocean Engineering*, **163**, pp. 431 – 450.
- [4] Utne, I., Sørensen, A., and Schjøberg, I., 2017. "Risk management of autonomous marine systems and operations". p. V03BT02A020.

- [5] Marani, G., Choi, S. K., and Yuh, J., 2009. "Underwater autonomous manipulation for intervention missions auvs". *Ocean Engineering*, **36**(1), pp. 15 – 23. Autonomous Underwater Vehicles.
- [6] Simetti, E., Casalino, G., Torelli, S., Sperindé, A., and Turetta, A., 2014. "Floating underwater manipulation: Developed control methodology and experimental validation within the trident project". *Journal of Field Robotics*, **31**(3), pp. 364–385.
- [7] Simetti, E., Wanderlingh, F., Torelli, S., Bibuli, M., Odetti, A., Bruzzone, G., Rizzini, D. L., Aleotti, J., Palli, G., Moriello, L., and Scarcia, U., 2018. "Autonomous underwater intervention: Experimental results of the maris project". *IEEE Journal of Oceanic Engineering*, **43**(3), pp. 620–639.
- [8] Cai, M., Wang, Y., Wang, S., Wang, R., Ren, Y., and Tan, M., 2020. "Grasping marine products with hybrid-driven underwater vehicle-manipulator system". *IEEE Transactions on Automation Science and Engineering*, **17**(3), pp. 1443–1454.
- [9] Chen, Z., Zhang, Z., Dai, F., Bu, Y., and Wang, H., 2017. "Monocular vision-based underwater object detection". *Sensors*, **17**(8).
- [10] Chen, Z., Zhang, Z., Bu, Y., Dai, F., Fan, T., and Wang, H., 2018. "Underwater object segmentation based on optical features". *Sensors*, **18**(1).
- [11] Cho, H., Gu, J., Joe, H., Asada, A., and Yu, S.-C., 2015. "Acoustic beam profile-based rapid underwater object detection for an imaging sonar". *Journal of Marine Science and Technology*, **20**(1), Mar, pp. 180–197.
- [12] Bonin-Font, F., Oliver, G., Wirth, S., Massot, M., Negre, P. L., and Beltran, J.-P., 2015. "Visual sensing for autonomous underwater exploration and intervention tasks". *Ocean Engineering*, **93**, pp. 25 – 44.
- [13] Antonelli, G., 2014. *Underwater Robotics*. Springer, Cham.
- [14] Bjelland, H. V., Føre, M., Lader, P., Kristiansen, D., Holmen, I. M., Fredheim, A., Grøtli, E. I., Fathi, D. E., Oppedal, F., Utne, I. B., et al., 2015. "Exposed aquaculture in Norway". In *OCEANS 2015-MTS/IEEE Washington*, IEEE, pp. 1–10.
- [15] Simetti, E., Campos, R., Vito, D. D., Quintana, J., Antonelli, G., Garcia, R., and Turetta, A., 2021. "Sea mining exploration with an uvms: Experimental validation of the control and perception framework". *IEEE/ASME Transactions on Mechatronics*, **26**(3), pp. 1635–1645.
- [16] Simetti, E., 2020. "Autonomous underwater intervention". *Current Robotics Reports*, **1**, pp. 117–122.
- [17] Skaldebø, M. B., Haugaløkken, B. O. A., and Schjøberg, I., 2021. "Seaarm-2 - fully electric underwater manipulator with integrated end-effector camera". In *2021 European Control Conference (ECC)*, pp. 236–242.
- [18] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.
- [19] BlueRobotics, 2021. Bluerov2. "URL <https://bluerobotics.com/store/rov/bluerov2/>".
- [20] Haugaløkken, B. O. A., Skaldebø, M. B., and Schjøberg, I., 2020. "Monocular vision-based gripping of objects". *Robotics and Autonomous Systems*, **131**, p. 103589.
- [21] Skaldebø, M., Haugaløkken, B. O. A., and Schjøberg, I., 2021. "Autonomous underwater grasping using a novel vision-based distance estimator". *Journal of Robotics and Automation*. unpublished paper.
- [22] Koenig, N., and Howard, A., 2004. "Design and use paradigms for gazebo, an open-source multi-robot simulator". In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE Cat. No.04CH37566), Vol. 3, pp. 2149–2154 vol.3.
- [23] FESTO DIDACTIC GMBH Co, 2003 [Online]. *COSIMIR® – PC-based 3D robot simulation software*. Rechbergstraße 3, 73770 Esslingen, Germany. "URL <https://www.festo-didactic.com/ov3/media/customers/1100/00030775001075223611.pdf>".
- [24] Michel, O., 2004. "Webots: Professional mobile robot simulation". *Journal of Advanced Robotics Systems*, **1**(1), pp. 39–42.
- [25] NTNU, 2020. Marine cybernetics teaching laboratory. Online, 11.

Article 8

Dynamic Bayesian Networks for Reduced Uncertainty in Underwater Operations

Martin B. Skaldebø, Ingrid Schjøberg
14th IFAC Conference on Control Applications in Marine Systems, Robotics,
and Vehicles, (CAMS 2022), Hamburg
[doi: 10.1016/j.ifacol.2022.10.462](https://doi.org/10.1016/j.ifacol.2022.10.462)

Dynamic Bayesian Networks for Reduced Uncertainty in Underwater Operations

Martin Skaldebo* Ingrid Schjølberg**

* *Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: martin.b.skaldebo@ntnu.no).*

** *Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: ingrid.schjolberg@ntnu.no).*

Abstract: This paper presents a novel framework for modelling dynamic Bayesian belief networks (BBNs) for online risk assessment in underwater operations. Existing frameworks spans from commercial software with restricted code access to non-profit open source frameworks. Frameworks with restricted code access often provides general user interfaces and visualization tools, while open source frameworks provides access to code for developers. The model presented in this paper pursues a best of both worlds scenario, where the model implementation should be uncomplicated while also providing visualization and verification to provide the user with a clear perception of the implemented model. The presented method is an expansion of the Bayesian model of the pomegranate python library, and simplifies the procedure of building, verifying and utilizing BBN models. The method is applied to a conceptual design of an underwater scenario case study with a model for an underwater vehicle manipulator system.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Bayesian belief network, online risk assessment, dynamic risk models, underwater intervention

1. INTRODUCTION

Humans conduct risk assessment in their decision making process, both intentionally and unconsciously. In underwater operations as in other maritime, aerial and land-based operations, risk assessment is a vital part of the human operator's decision making process. If such systems should move towards more autonomy, it is imperative that the risk assessment follows. However, it has proven to be difficult to map the human capability for risk assessment as a computational ability. This is due to the challenges in identifying the relevant knowledge in an ocean of available information, and difficulties in computerizing the background knowledge of the process. "We are drowning in information and starving for knowledge" - *Rutherford D. Roger*. Increased autonomy in various types of operations is not a goal in itself, rather a results of reducing costs and risk and increasing safety. Technological advances in autonomous systems can arguably improve the safety, efficiency and performance by aiding in decision-making [1]. In this paper, an approach of using dynamic Bayesian belief networks (BBNs) is proposed, presenting *Another Bayesian Belief Analyser* (ABBA) which is a novel BBN framework designed for dynamic and online risk assessment purposes. A clear advantage of using BBNs when assessing risk is enabling a network to assess risk by analysing data while also exploiting the human knowledge of the process. Using historic data to train the models can allow for robust models specialized for a specific task or scenario. BBNs can also use expert judgement to decide causal dependencies and probabilities in the networks. Consulting experts when

designing the internal behaviour of BBNs can however lead to flawed models and conclusions, due to limited and subjective knowledge. In both approaches, deciding arcs and nodes in the networks allows for human knowledge to be transferred to the risk assessment process. When doing so, identifying relevant nodes becomes imperative. If important causal dependencies or nodes are missing in the BBN model, it would significantly affect the results. ABBA attempts to circumvent the use of expert judgement by analysing historic data of the process, however leaving it to the user to decide the causal dependencies to exploit the users overall knowledge of the process.

The main contributions of this paper is summarized with:

- Discussing the importance of risk assessment in underwater operations and presenting how the increase in autonomy brings higher requirements for risk assessment.
- An overview of existing BBN frameworks will be provided along with discussion about the novelties of ABBA compared to existing solutions.
- The development of ABBA, a dynamic approach of designing BBNs with graphical visualization and adaptable source code for implementation with other systems.
- A conceptual design of a case study presenting how ABBA is utilized to reduce uncertainties in an underwater vehicle manipulator system (UVMS).

The paper is structured as follows. Section 2 presents information about risk assessment and the use of risk in

Table 1. Processes with dynamic and online risk assessment

	Process	Dynamic	Online
1	Risk assessment performed by the captain on a fishing vessel regarding the risk for going out at sea for a mission	YES	NO
2	Advanced Guidance and navigation control with risk assessment for collision for Autonomous Surface Vessels (ASV) and Autonomous underwater vehicles (AUV)	YES	YES
3	Updating evidences in a Bayesian belief network (BBN) to update risk values	NO	YES
4	Updating evidences in a BBN to update risk values while also using gathered information to update and retrain the model	YES	YES

underwater operations. Section 3 discuss Bayesian belief networks and existing solutions for developing BBNs. Section 4 presents the novel ABBA framework with information about configuration, data formats and verification method, and a model of ABBA for a UVMS is presented in Section 5. Section 6 provides results which are discussed in Section 7 and conclusions and suggestions for further work are provided in Section 8.

2. RISK ASSESSMENT

2.1 Dynamic and online risk

There are contradicting beliefs and opinions concerning some terminology in the risk community, especially dynamic risk and online risk are two terms that are interchangeably used. Time dependency, varying models, parameter updates and connectivity are aspects that are emphasized when attempting to define these terms. In this paper, the elemental definitions of the adjectives are used to formalize definitions of the risk terms. Thus, in this paper, dynamic risk is defined as a process exposed to constant change, activity or progress, while online risk is defined as a process that is controlled by, or connected to, a computer. The definitions are taken from the online dictionary *Lexico* provided by Dictionary.com and Oxford University Press [2]. Table 1 illustrates some examples of processes and how they would be characterized with the definitions used in this paper. Examples 3 and 4 from Table 1 are the most relevant for the work presented in this paper.

The reason example 3 is considered not to be dynamic while the other processes are considered dynamic is the fixed model used for the assessment. Advanced risk models applied in e.g. ASVs and AUVs (ex. 2) can include changing systems considering different variables for different situations, and more intelligent solutions may even include models able to evolve during operations with the use of internal learning. Example 3 represents a risk process that involves performing risk assessment with a fixed model. Moreover, the input to the model will be updated with sensor data and the calculated risk will be included in the decision-making process of the system, hence it is online,

yet not dynamic. Example 4 however can be considered to be dynamic. Utilizing ABBA in such a process allows for the user to alternate the model how ever desired and train a new model that considers new sensors and new data. The framework is also capable of being incorporated as part of an intelligent system where it can decide to train a new model with new data if the system finds this beneficial.

2.2 Risk assessment in increasingly autonomous operations

Risk is often defined as the product of likelihood of an event and consequence if the event occurs,

$$\text{Risk} = \text{Likelihood} \cdot \text{Consequence}. \quad (1)$$

Considering this definition and further analysing the risk often attempts to answer three main questions [3].

- (1) What can go wrong?
- (2) What is the likelihood of something going wrong?
- (3) What are the consequences if something goes wrong?

These three question demonstrates why risk assessment are challenging for increasingly complex systems. Identifying everything that can go wrong have proven to be problematic, especially since a person's information and knowledge about a complex system can be inadequate. Lack of historic data for new systems can decrease the knowledge about the likelihood of something going wrong as well as the severity of the consequences. It is therefore essential that risk analysis becomes a vital part of the design, building and operation process, especially for increasingly autonomous and intelligent systems [1].

Introducing more autonomy in operations will increase the necessity of risk assessment performed by the system. Take the underwater segment, that is an area highly focused on increased autonomy. The underwater scene provides a harsh environment in constant change, which introduce the importance of dynamic and online solutions. With increased autonomy follows increased significance of decisions, which again increase the importance of making correct decisions. Autonomous underwater vehicles (AUVs) may possess collision avoidance systems, path planning systems and other systems analysing when the risk of continuing operation is too high and counteractions should be taken. In such operations the humans usually do not have the ability nor the accessibility to the system in order to oversee the system, and the system must perform the decision-making independently. Loss of AUV during operations is not an uncommon phenomena which proves that existing risk assessment solutions have room for improvements.

In the underwater scene, increasing autonomy in remotely operated vehicles (ROVs) is also a growing field of interest. Since humans are still an important participant in ROV operations, it may not require the system to act and decide independently in critical situations, but the systems should however possess strong risk awareness applications. In a human supervisory system, humans should not intervene unless necessary, thus the system should be able to perform decision-making independently up to a certain significance of potential consequences. Such systems should also be able to provide the human supervisor with a simple situational awareness. If the human supervisor should be

able to assess the overall risk of operation, it could be problematic to inspect all available data and sensory feedback. The semi-autonomous systems should therefore be able to collect available data, analyse it and provide a simpler understanding of the potential risk of operation for the human supervisor to understand. Consequently, risk assessment is important in all levels of autonomy, whereas requirements for robustness and assurance of correct and sufficient decisions increases with the increased levels of autonomy.

3. BAYESIAN BELIEF NETWORK SOFTWARE

There are several existing software and frameworks for modeling BBNs, both licensed products and free to use open source codes. Table 2 lists the most used available software and frameworks. The graphical user interface (GUI) based software provide visualization tools that the command line based software does not. However, such frameworks often have free access to the source code, which makes it highly adaptable for different applications because the user can modify the source code to fit different purposes.

3.1 Commercial software

Some of the most known commercial software used when computing BBNs are GeNie, HUGIN and Netica. The software all have a graphical user interface (GUI) and restricted source code. GeNie is a GUI to the SMILE Engine which is a reasoning and causal discovery engine for graphical models i.e. BBNs and others. Smile and GeNie are provided by BayesFusion which specializes in using Bayesian networks in artificial intelligence modelling and machine learning software. SMILE is the software library which can be embedded into existing user software and GeNie is BayesFusion's GUI for the library where interactive modelling is supported. HUGIN EXPERT A/S provides decision support software and uses graphical models based on Bayesian network technology. Their software mainly provide analytic solutions for detecting fraud, credit card fraud and assisting corporations in analyzing their customers in suspicion of money laundering. Netica is a program for BBNs and influence diagrams developed by Norsys Software Corp., which is used for drawing networks and calculation of causal relationships either by individual probabilities from expert judgement or learned from data. BayesFusion provides free academic versions of their software GeNie, while HUGIN and Netica are only provided without cost through lite test-versions. [6, 7, 8].

3.2 Free to use software

Existing non-commercial software provides free use of software, but are often less developed than the commercial software. However, with access to source code, the software can be highly adaptable and easily incorporated with other systems and code. This often requires a good comprehension of the code and can provide problems for users not that familiar with either the programming language, architecture of the code, or the utilized models. Some available software, e.g. the pomegranate package, are powerful

tools that provide a range of models and applications. For advanced users these tools are exceptional, however the large libraries can be hard to comprehend for less advanced users. See [9, 10, 11, 12, 13, 14, 15] for more information on non-commercial software listed in Table 2.

4. ANOTHER BAYESIAN BELIEF ANALYZER (ABBA)

ABBA is an open source python library based on the Bayesian module of the pomegranate library. ABBA extends pomegranate in two ways by (1) enabling easy calculations and implementations of conditional probability tables (CPTs), and (2) visualizing the Bayesian network similar to other graphical user interface (GUI) based software. Thereby incorporating the best of both worlds. The library takes a .txt file as input where all the relevant nodes are described. The .txt file has the format `<<name1;name2;scale;limits;parent nodes>>` where the attributes are described in further details in Table 3. Note that the input file differentiate between attributes with semicolon.

4.1 Historic Data Format

ABBA uses historic data to calculate the conditional probability tables (CPTs). This circumvents the need for expert judgements. This data should be in a specific format for the library to be able to correctly read it. The data should be collected in a .csv file, where each node has its own column of historic data and where each row represents a measurement. The name of the node in the .csv file should match the attribute `name2` from the input .txt file. If the historic data is collected from a larger database it can contain insignificant or irrelevant datapoints. This is of no concern because the relevant datapoints that should be included in the BBN are specified in the input .txt file. Thereby, extracting only information about the relevant nodes from the .csv file.

4.2 Verification

In order to use the model for predicting new scenarios, we have to be confident in the model's legitimacy. In order to determine if the model is able to predict unseen scenarios it should be properly verified. The verification of the model will provide insight in how well it predicts unseen situations and a good verification contribute with a sense of belief in the model. The verification is used in order to validate if the software performs as intended.

ABBA uses root mean square error (RMSE) with K-fold cross validation to validate the model. The training data is divided into K partitions and the training is performed K times. For each iteration the model trains on $K - 1$ partitions and is validated on the last partition, which is repeated until all partitions have been used for validation. The RMSE represents the difference in the observed or true values and the estimated/predicted values. Considering all K samples the overall RMSE used for verification in ABBA is computed with

$$RMSE = \sqrt{\frac{\sum_{i=1}^K \sum_{j=1}^{n_i} (y_i - \hat{y}_i)^2}{K n_i}}, \quad (2)$$

Table 2. Review of Bayesian network modeling software. Updated and extended from an earlier review from [4] and [5]

Name	Programming language	Type of interface	Licensing type	Access to source code
GeNie	C++, Java, R, Python, .NET	GUI	Commercial	Restricted
HUGIN	Undisclosed	GUI	Commercial	Restricted
Netica	Undisclosed	GUI	Commercial	Restricted
VIBES	Java	GUI	Free	Restricted
SamIam	Java	GUI	Free	Restricted
UnBBayes	Java	GUI	Free	Restricted
BNT	Matlab, C	GUI, Command line	Free	Open
Pomegranate	Python	Command line	Free	Open
bnlearn	R	Command line	Free	Open
gRain	R	Command line	Free	Open

Table 3. Description of the attributes in the input file to ABBA

Attribute	Description
name1	Name of the node (should not contain spaces)
name2	Name of the attribute in the .cvs file
scale	Scale defining different states in the CPTs
limits	Limits for classification within the relevant scales
parent nodes	name1 of all parent nodes. If none, leave blank

where y_i and \hat{y}_i are model variables and predicted values respectively, n_i is the number of samples in iteration i , and K is the number of partitions in the cross validation.

5. CASE STUDY

The case study is an attempt to further improve the methods and results presented in [16] and [17]. [16] presented a relative dynamic positioning system of a low cost underwater vehicle relative to a object of interest using monocular camera for detection, and [17] continued this work by introducing a manipulator to simultaneously perform intervention of the relevant object.

In the previous work, the solution encounter problems when the manipulator occludes the object of interest. The dynamic positioning system is dependent on the distance calculated by a scaling function, and occlusion of the object results in a faulty detection and estimated distance. The goal of this case study is thus to investigate if ABBA can be used to formulate a value of belief in the estimated distance. In this way, a low belief in the estimated distance should result in the system not trusting this estimation and hence find other measurements to use for the estimation, and a high belief should result in the system being confident in the estimation.

5.1 BBN modeling

The methodology used for modeling the architecture of the BBN in this case study is adopted from [18]. The authors of the paper states three requirements for the nodes that have to be met in order to develop a BBN.

- (1) The nodes can be defined
- (2) The state of the nodes can be represented by measurable variables
- (3) The target node and any other node in the network have known traceable direct/indirect relationships

The final structure of the BBN is depicted in Fig. 1.

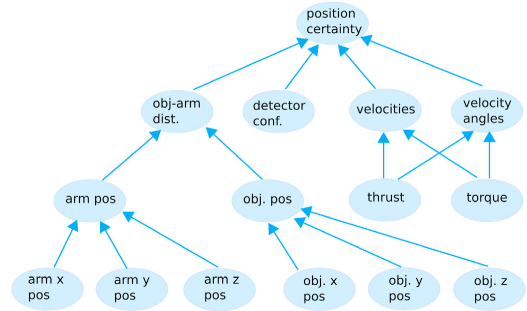


Fig. 1. BBN structure with causal dependencies between nodes.

5.2 ABBA in the control loop

ABBA should be incorporated in the control loop in order to influence the estimation from the detector. Fig. 2 presents how ABBA influences the motion control system (MCS), where the Position estimator block and ABBA block is new relative to the control loop from [17] and the dotted lines represents dependencies introduced in the new control loop. In this model, ABBA represents a trained BBN with fixed structure as presented in Fig. 1. The nodes *arm x pos*, *arm y pos*, *arm z pos*, *obj. x pos*, *obj. y pos*, *obj. z pos*, *thrust* and *torque* are updated online in the control loop and influence the other nodes as well as the target node, *position certainty*. In order to preserve as much of the original MCS as possible, the guidance system is kept original. The relative distance fed to the MCS however is influenced by the indication from ABBA. Previously the MCS received relative distances from the detector which was used for calculating reference velocities in the guidance system. In the new control loop the MCS receives relative distances from the position estimator, which are already adjusted by the influence of the BBN. The position estimator will use the position certainty from ABBA to adjust the distances from the detector. If the system no longer trust the measures from the distance estimator, it will put higher value to other sensor measures in the system and less on the object detection.

5.3 Collection and augmentation of data

Sufficient amount of data have to be collected in order to train the BBN. The data consist of measurements with a constant time step. A satisfactory dataset should include

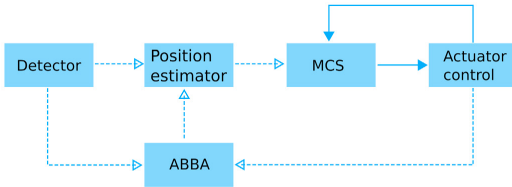


Fig. 2. Control loop.

Table 4. Data collection for each node in the BBN.

Node	Measure procedure
Arm x position	Through forward kinematics in kinematic control
Arm y position	Through forward kinematics in kinematic control
Arm z position	Through forward kinematics in kinematic control
Object x position	Extracted from the detector
Object y position	Extracted from the detector
Object z position	Extracted from the detector
Arm position	Calculated based on parent nodes
Object position	Calculated based on parent nodes
Thrust	Equal the thrust given to the propellers
Torque	Equal to the torque given to the propellers
Distance between object and arm	Calculated based on parent nodes
Detector confidence	Extracted from the detector. The confidence is a measure of how confident the detector is in the detection of the object and is a result of the training of the detector.
Velocities	Estimated with a Kalman filter based on position data
Velocity angles	Estimated with a Kalman filter based on position data
Position certainty	Calculated by considering difference in believed position and true position from Qualisys

values for every node involved in the BBN. Some of the nodes in the BBN architecture were included after performing parent divorcing and does not represent a direct measurement or a direct causal relationship with the parent nodes. Take the node *arm pos*, which has definitional relationships with the parent nodes, where the position can be defined from the nodes *arm x pos*, *arm y pos* and *arm z pos*. To ensure a sufficient dataset, the calculated values will be evaluated equivalent to direct measurements and recorded equally in a .csv file. The position certainty as the target node also needs to be measured for collected data. If this value could be measured at all times, the need for the BBN to estimate it would be unnecessary. Moreover, for the collection of the dataset, this value will be measured using the motion capture system Qualisys. Qualisys is installed and calibrated in the marine cybernetics laboratory (MC-Lab) at NTNU and uses optical tracking technology to measure position of objects using cameras and known markers attached to the objects. An overview of how the specific nodes in the BBN are measured is listed in Table 4.

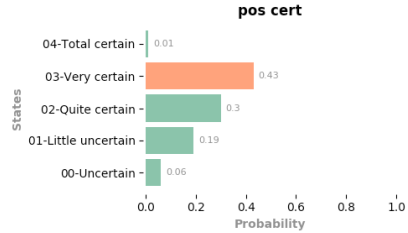


Fig. 3. CPT for the target node.

6. RESULTS

ABBA has been utilized to build a BBN model with architecture and nodes as presented in Fig. 1. Each node in the BBN includes CPTs as presented in Fig. 3 for the target node. Note that no experiments have yet been conducted and thus the data presented in the figure are only from a conceptual design.

ABBA is able to build the BBN with corresponding CPTs from the input .csv file and the descriptive .txt file. A built in generic classifier categorize measurements from the .csv file based on the limits and scales described in the .txt file and builds causal dependencies as described in the same file.

A conceptual design of the BBN in the control loop is presented, where the BBN will affect the MCS by invoking evidence on child nodes based on real time sensor measurements. As explained in Section 5.2, the nodes *arm x pos*, *arm y pos*, *arm z pos*, *obj. x pos*, *obj. y pos*, *obj. z pos*, *thrust* and *torque* will be updated real time. These measurements can be extracted at real time with the same procedure used when logging the data. These procedures are discussed in Table 4.

7. DISCUSSION

A BBN for the presented case was built using ABBA. Once the data is collected in a .cpt file and the parameters are correctly described in a .txt file ABBA does the rest. The tool is simple to use and provides good visualisation of the network. A built-in generic classifier enables the tool to be directly incorporated in dynamic systems where evidence derives from measurements. The classifier characterizes measured values within the scales and limits as described in the input .txt file and updates the network based on these measurements. The BBN model is also capable of being updated real time if measurements, time dependencies or other factors in the overall system requires it. Such scenarios could appear if the system operates in a dynamic environment where updates of the models are required in order to detain a compatible system.

As discussed in Section 1, an advantage in the BBN approach is exploiting human knowledge of a process while still using historic data to train a model. The human knowledge is exploited in the modelling of the network. When modelling the BBN presented in the conceptual design of the case study, a generic 8-step approach was used. Challenges that emerged when modeling the BBN

are partially discussed in Section 5.1, however other challenges occurred as well. Parent divorcing and introduction of intermediary nodes solved challenges related to complex CPT calculations and computational load. Another investigated solution here was the alteration of scales and limits corresponding to the relevant nodes. The number of feasible states for a node determines the accuracy of the measurements and a should not be too low nor too high. The scales and limits determines in which state the measurements are categorized and should be carefully considered. An analyse of the available data was also performed here, although not explicitly explained in the modeling description. The available data should resemble a normal distribution which can be arranged by alternating the scales and limits to fit the data. If the scales for a node categorize all available data to one single state, there is no existing information for the other nodes. Information and knowledge about what happens if another state is achieved are then missing from the model. During the modeling of the BBN, the data should therefore be thoroughly analysed.

An analyse tool for analysing the data before training the CPTs could easily be implemented in the ABBA tool. Such a tool could analyse the available data before determine, or at least suggest, scales and limits for nodes in the model. The analyse tool could then provide scales and limits that would ensure normal distributed distributions. A limitation with such a tool is the decrease of human knowledge in the process. Automatically chosen scales and limits could also provide worse overall achievement of the model if the data is not good. If the original data is skewed or flawed in any other way, the analyse tool would adjust the scales and limits according to bad information. However, it could still be beneficial to create this analysing tool to at least suggest scales and limits. Manually analysing the data and numerous iterations in the modeling steps for determining the scales and limits were time consuming, and a suggestive tool could help streamline this process.

A conceptual design of the model incorporated in the control loop have been presented. Furthermore, this will provide more robust estimates of the relative position of the object. However, so far this is only a conceptual design and no experiments have been conducted to verify the suggested solution. Experiments of the preliminary system without the use of ABBA was conducted in MC-Lab at the department of marine technology at NTNU. Preferably, new experiments should be conducted at the same facility using the same equipment. In this way the experiments are comparable with previous experiments. This is essential in order to draw any conclusions regarding the improvements of the system. A proper verification of the model has not been conducted in the lack of available data. The quality of the model depends both on the quality of the data as well as the architecture of the BBN model. In order to ensure that the presented model is valid it should be verified using the presented verification method with collected data from MC-Lab.

8. CONCLUSIONS

A framework for modeling dynamic BBNs have been presented in this paper. The presented framework, ABBA,

enables simple calculations of CPTs and provides strong verification and visualization tools. A conceptual design is presented, where ABBA should help increase estimations in a UVMS performing DP and autonomous intervention. In this case, a static BBN model is built using ABBA where relevant nodes in the network can be updated in real time in order to perform online real time risk assessment. The natural next steps are to expand the presented work with collection of historic data and experimental testing of the system.

REFERENCES

- [1] *Risk Management of Autonomous Marine Systems and Operations*, volume Volume 3B: Structures, Safety and Reliability of *International Conference on Offshore Mechanics and Arctic Engineering*, 06 2017. V03BT02A020.
- [2] *Lexico, Powered by Oxford*. Provided by Dictionary.com and Oxford University Press (OUP), November 2019.
- [3] Stanley Kaplan and B. John Garrick. On the quantitative definition of risk. *Risk Analysis*, 1(1):11–27, 1981.
- [4] Mohamed Ali Mahjoub and Karim Kalti. Software comparison dealing with bayesian networks. In Derong Liu, Huaguang Zhang, Marios Polycarpou, Cesare Alippi, and Haibo He, editors, *Advances in Neural Networks – ISNN 2011*, pages 168–177, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [5] Kevin Murphy. Software for graphical models: A review. *Int Soc Bayesian Anal Bull*, 14, 01 2007.
- [6] *GeNie Modeler: Complete Modeling Freedom*. BayesFusion LCC, 2019.
- [7] *Explore Bayesian Network Technology*. Hugin Expert A/S, 2019.
- [8] *Netica Application*. Norsys Software Corp., 2019.
- [9] *Variational Inference for Bayesian Networks*. Sourceforge, 2019.
- [10] *Sensitivity Analysis, Modeling, Interfercens And More*. Automated Reasoning Group, UCLA, 2019.
- [11] *UnBBayes Overview*. University of Brasilia, 2019.
- [12] *Bayes Net Toolbox for Matlab*. Kevin Murphy - Github, 2019.
- [13] Jacob Schreiber. Pomegranate: fast and flexible probabilistic modeling in python. *CoRR*, abs/1711.00137, 2017.
- [14] *bnlearn - an R package for Bayesian network learning and inference*. Marco Scutari, 2019.
- [15] *gRain: Graphical Independence Networks*. Søren Højsgaard, 2019.
- [16] M. Skaldebo, B. O. Arnesen, and I. Schjøberg. Dynamic positioning of an underwater vehicle using monocular vision-based object detection with machine learning. *OCEANS 2019 Seattle Conference Exposition, Seattle, USA*, 2019.
- [17] B. O. Arnesen, M. Skaldebo, and I. Schjøberg. Monocular vision-based gripping of objects. *Robotics and Autonomous Systems*, 2019.
- [18] Jeevith Hegde, Ingrid Bouwer Utne, Ingrid Schjøberg, and Brede Thorkildsen. A bayesian approach to risk modeling of autonomous subsea intervention operations. *Reliability Engineering System Safety*, 175:142 – 159, 2018.

Article 9

Autonomous subsea intervention (SEAVENTION)

Aksel A. Transeth, Ingrid Schjøberg, Anastasios M. Lekkas, Petter Risholm, Ahmed Mohammed, **Martin B. Skaldebø**, Bent O. A. Haugaløkken, Magnus Bjerkeng, Maria Tsiourva, Frederic Py
14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, (CAMS 2022), Lyngby
[doi: 10.1115/OMAE2022-79913](https://doi.org/10.1115/OMAE2022-79913)

Autonomous subsea intervention (SEAVENTION)[★]

Aksel A. Transeth^{*} Ingrid Schjølberg^{**}
Anastasios M. Lekkas^{***} Petter Risholm^{****}
Ahmed Mohammed^{****} Martin Skaldebø^{**}
Bent O. A. Haugaløkken[†] Magnus Bjerkgeng^{*}
Maria Tsiourva^{*} Frederic Py^{*}

^{*} SINTEF Digital, Mathematics and Cybernetics, Trondheim, Norway
(e-mail: aksel.a.transeth@sintef.no)

^{**} NTNU, Department of Marine Technology, Trondheim, Norway
(e-mail: ingrid.schjolberg@ntnu.no)

^{***} NTNU, Department of Engineering Cybernetics, Trondheim,
Norway (e-mail: anastasios.lekkas@ntnu.no)

^{****} SINTEF Digital, Smart Sensor Systems, Oslo, Norway (e-mail:
ahmed.mohammed@sintef.no)

[†] SINTEF Ocean, Aquaculture operations and robotics, Trondheim,
Norway (e-mail: bent.haugalokken@sintef.no)

Abstract: This paper presents the main results and latest developments in a 4-year project called autonomous subsea intervention (SEAVENTION). In the project we have developed new methods for autonomous inspection, maintenance and repair (IMR) in subsea oil and gas operations with Unmanned Underwater Vehicles (UUVs). The results are also relevant for offshore wind, aquaculture and other industries. We discuss the trends and status for UUV-based IMR in the oil and gas industry and provide an overview of the state of the art in intervention with UUVs. We also present a 3-level taxonomy for UUV autonomy: mission-level, task-level and vehicle-level. To achieve robust 6D underwater pose estimation of objects for UUV intervention, we have developed marker-less approaches with input from 2D and 3D cameras, as well as marker-based approaches with associated uncertainty. We have carried out experiments with varying turbidity to evaluate full 6D pose estimates in challenging conditions. We have also devised a sensor autocalibration method for UUV localization. For intervention, we have developed methods for autonomous underwater grasping and a novel vision-based distance estimator. For high-level task planning, we have evaluated two frameworks for automated planning and acting (AI planning). We have implemented AI planning for subsea inspection scenarios which have been analyzed and formulated in collaboration with the industry partners. One of the frameworks, called T-REX demonstrates a reactive behavior to the dynamic and potentially uncertain nature of subsea operations. We have also presented an architecture for comparing and choosing between mission plans when new mission goals are introduced.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Autonomous and remotely operated marine vessels; Intelligence and autonomy in marine systems and operations

1. INTRODUCTION

Underwater infrastructure plays a key role in today's society and includes, e.g., oil and gas installations, aquaculture facilities, underwater cables, etc. Moreover, both, e.g., aquaculture and oil and gas infrastructure are expected to be installed in even more exposed areas offshore. Also, renewable energy installations such as offshore wind is alone projected to reach \$56.8 billion by 2026 (Nhede, 2021). To optimize asset uptime and ensure proper HSE (Health,

Safety and Environment) Unmanned Underwater Vehicles (UUVs) are used for inspection, maintenance and repair (IMR) of offshore infrastructure, and the global underwater robotics market in general is expected to reach \$4914 million at a CAGR of 12.5 % from 2018 to 2025 (MarketResearchFuture.com, 2020). Still, most UUVs in IMR operations are piloted by humans or otherwise operating with a limited degree of autonomy. Increased autonomy in UUV operations can improve HSE and efficacy, lower emissions and reduce cost in UUV-based IMR operations (Schjølberg et al., 2016). Methods to achieve such autonomy are the topic of our 4-year project called Autonomous Subsea Intervention (SEAVENTION) and in this paper we summarize the main results and latest developments from the project. The subsea oil and gas industry has been the

^{*} This research was funded by the Norwegian Research Council, grant number 280934 and its industry partners Equinor, Oceaneering, IKM and TechnipFMC. The work was carried out in the SEAVENTION project (www.sintef.no/SEAVENTION) led by SINTEF with NTNU as research partner.

main focus for the SEAVENTION project, but the results are also applicable to IMR operations in other sectors such as offshore wind and aquaculture.

The subsea oil and gas industry has been a long-term user of UUVs for IMR. In particular, Remotely Operated Vehicles (ROVs) have been the standard for IMR. ROVs are tethered and typically have two robot arms for intervention operations such as turning valves, cleaning, etc. Over the recent years, UUVs have been endowed with autonomous capabilities, but these are mainly limited to non-intervention type tasks such as inspection, hovering, follow-pipeline, move from A to B, etc. Intervention operations (e.g., cleaning, turn valves) are mostly remotely piloted by one or more human operators. Thus, the efficiency and success-rate of such operations rely on the operator skills. Hence, factors such as limited visibility and ocean currents can make operations very challenging to perform in a safe and efficient manner. Hence, there is a need for methods for robust perception and intervention to meet these industry challenges.

In this paper we present main results and recent developments on UUV autonomy from the SEAVENTION project ranging from perception and control to automated high-level mission/task planning. We discuss the status and trends in IMR with UUVs in the oil and gas industry and provide an overview of the state of the art on UUV intervention. Moreover, we list IMR use-cases, present a 3-level taxonomy for UUV autonomy, and we summarize our results on marker-based (with associated uncertainty) and marker-less approaches for object detection and localization for UUV intervention. We also present an approach to UUV localization sensor autocalibration. Lastly, we provide an architecture for comparing plans for UUV missions in a planning framework called ROSPlan and compare ROSPlan with another framework called T-REX.

2. BACKGROUND

2.1 State of the industry

In this section, we provide an overview of how subsea IMR operations with UUVs are typically being performed in the oil and gas industry today and what are the subsea operational concept trends.

The main mode of operation for IMR is that ROVs and ROV pilots are brought by large topside support vessel to the area of subsea installations in need of IMR. The ROVs are tethered and operated by ROV pilots often onboard the surface vessel (typically large ships, see Fig.1). In recent years, remote support of operations with ROV pilots and/or subject matter experts onshore have become more commonplace. E.g., SEAVENTION project partner IKM has demonstrated remote controlled ROV operations over a distance of 11 000 km (IKM Subsea, 2019). Moreover, SEAVENTION partner Oceaneering reports that *ROV remote piloting from shore can increase safety and reduce the environmental footprint of operations, potentially achieving up to 25% reduction in offshore personnel on board (POB) and provide a significant reduction in emissions associated with the work* (Oceaneering, 2021).

A next step beyond today's need for large topside support vessel is to have UUVs that perform IMR on subsea



Fig. 1. Example of ROV support vessel (Ken Doerr, CC BY 2.0 <https://creativecommons.org/licenses/by/2.0>, via Wikimedia Commons)



Fig. 2. A Merlin Resident ROV with its “garage” from IKM. Image courtesy of IKM.

templates without topside vessels present. In June 2017, the world's first commercial ROV operation with a “work-class” ROV was carried out from IKM Subsea's operation center in Stavanger, Norway (See Fig.2. Even though such “resident” UUVs are still not commonplace in subsea IMR, oil and gas companies such as Equinor are pushing toward such operational capabilities and thus we may see more of it in the future. As an example, Equinor has engaged a company called Saipem to deliver resident UUVs (Saipem, 2019). In addition to resident UUVs designed to operate within limited areas, e.g., close to a subsea template-type of infrastructure, other UUVs are starting to appear, such as the Freedom Autonomous Vehicle from Oceaneering (see Fig. 3) with a survey range of 120 km and offshore trials reported to be in 2021 (Oceaneering, 2022). To support UUVs traveling between subsea assets, companies such as Equinor in cooperation with Blue Logic are testing UUV docking stations with standardized interfaces for UUV charging (BlueLogic, 2019).

In the past, UUVs were divided into mainly two categories; 1) ROVs (tethered, limited operational area, hovering capabilities, typically equipped with two robot arms and a range of inspection and intervention tools) and 2) Autonomous Underwater Vehicles (AUVs). AUVs were often “torpedo shaped”, designed for long-range surveys and monitoring missions (e.g., seabed mapping, pipeline inspection) without hovering and intervention capabilities. However, we now see more “hybrid” types of UUVs capable of hovering, intervention, and more long-range survey operations. Such UUVs include, e.g., the Freedom Autonomous Vehicle, Aquanaut by Nauticus Robotics (nauti-

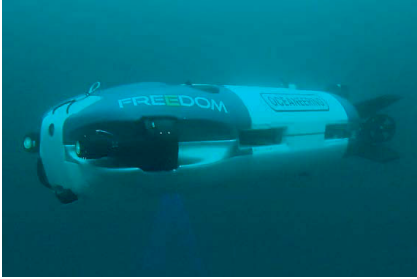


Fig. 3. The Freedom Autonomous Vehicle from Oceaneering. Image courtesy of Oceaneering.

cusrobotics.com) and a snake-like UUV called the Eelume Vehicle (Fig. 4). Some of these can switch between being tethered and un-tethered. Equinor has coined the term “Underwater Intervention Drones” (UIDs) for UUVs that can operate without a tether and perform intervention tasks.



Fig. 4. The Eelume vehicle. Image courtesy of Eelume.

2.2 The state of the art on autonomous intervention

UUVs are to an increasing extent equipped with manipulators for intervention purposes. Such vehicles are referred to as underwater vehicle manipulator systems (UVMS) (Antonelli, 2014). This provides a moving base for the manipulator that strengthens manipulation capabilities and enables more autonomy in intervention operations. Autonomous underwater intervention is relevant in a vast amount of diversified scenarios, from pipelines and operational panels in offshore industry to collecting organisms such as plants and fish. The latter case requires gentle and agile grasp in order to not damage or injure the object of interest (Huang et al., 2020). Such scenarios would require a system with high accuracy and delicate movements which again set the requirement for both hardware and software. A manipulator is a versatile tool with its 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 (Schjølberg and Utne, 2015), aquaculture (Bjelland et al., 2015), ocean mapping, environmental monitoring, surveillance, etc. (Simetti, 2020). The high variation in scenarios where manipulators and UVMSs are deployed is met with an equally high variation in available systems. Underwater manipulators vary in the range from small electric manipulators with limited lifting capacity and depth rating to large hydraulic manipulators

capable of lifting hundreds of kilos at depths of up to several thousand meters. Manipulators in all sizes vary from simple setups with few to none joints and simple open/close gripper functionalities to more advanced manipulator that may inherit a variety of integrated capabilities, e.g., force feedback, joint position readings, multiple internal controllers and so on (Sivčev et al., 2018).

Autonomous underwater intervention has long been a prominent research topic, with considerable variations of innovative solutions in the research community. One of the first autonomous underwater intervention operations in the oceanic environment was conducted through the SAUVIM project (Marani et al., 2009). Since then, among other projects that targeted autonomous intervention operation we find, e.g., TRIDENT (Simetti et al., 2014), MARIS (Simetti et al., 2018) and DexROV (Gancet et al., 2015). One of the leading projects in autonomous intervention today is the SUONO project (Topini et al., 2021), which aims to develop autonomous systems capable of performing underwater intervention operations such as free-floating manipulation tasks on a subsea panel.

3. USE CASES AND LEVELS OF AUTONOMY

Typical IMR operations subsea which involves the use of UUVs (mostly ROVs) include cleaning, visual inspection, valve operations, hot stab operations, installation/retrieval/replacement of modules and components (e.g., flying leads, jumpers, cables, sensors, meters), electrical faultfinding and hydraulic lead detection, operating hatches, cutting, surveying (e.g., with cameras) and Cathodic Potential (CP) measurements of structures and pipes (Schjølberg et al., 2016).

The above operations are complex and can be broken down to subtasks such as change tool; detect, locate and move to valve; record camera data; operate tool; docking; charging; etc. where the latter is specifically for non-tethered UUVs. IMR operations require UUV capabilities ranging from low-level vehicle control to high-level mission planning. To this end, we propose a 3-level division of autonomous UUV capabilities:

Mission-level: UUV capabilities in terms of planning its own missions fully or partly (in collaborating with a human operator). Such mission planning can include sequencing and coordinating a variety of task-level capabilities to achieve mission goals while taking into account vehicle and environment constraints.

Task-level: Capabilities to carry out single tasks autonomously or through high-level cooperation with a human pilot. Such tasks could include turn valve, inspect gauge, follow-up pipeline, etc.

Vehicle-level: These capabilities include autonomous hovering, collision avoidance, and object detection and localization.

With the above taxonomy, we see from Sec. 1 and 2.1 that today’s UUV operations in oil and gas are mostly limited to vehicle-level autonomous capabilities, while for inspection operations (e.g., pipeline following), UUVs show, to some extent, task-level autonomy.

4. ROBUST 6D UNDERWATER POSE ESTIMATION

To achieve reliable autonomous UUV intervention we need robust 6D (6-Degrees of Freedom) pose estimation (vehicle-level autonomous) capabilities. To achieve such pose estimation, we have developed and tested marker-less deep learning (DL) approaches with input from 2D and 3D underwater cameras, as well as marker-based approaches with associated uncertainty with 2D cameras (e.g., with Aruco markers). We have carried out controlled experiments with varying turbidity to evaluate if the proposed systems provide robust 6D pose estimates in challenging conditions. To facilitate the training of machine learning-based perception systems, we have also implemented an approach to collect and automatically annotate underwater 6D pose estimation datasets. In the following, we summarize our efforts on these topics.

Collection and automatic annotation of dataset for 6D pose estimation The performance of 6D localization has significantly improved with the advent of deep learning – especially in terrestrial application. Our research has focused on how 6D DL methods can be adapted to images acquired in the underwater environment while retaining their superior performance. One of the main challenges is that the visual appearance of the same object will vary with the turbidity. The higher turbidity, the more noise and less contrast we will observe in the images. To reliably train such DL networks requires large annotated datasets which can be costly to generate.

One approach which we have developed (and published in Mohammed et al. (2021)), is a way of generating a dataset for 6D localization with automated 6D labeling even under turbid conditions. We created a mockup subsea panel which contained objects such as valves, gauges and fish-tails. A number of Aruco markers were placed around the panel (both above and below the water). Two cameras were rigidly attached to each other, where one camera was located underwater while the other (an underwater 3D camera) was located above water. Since the positions of the Aruco markers in relation to the objects of interest were well calibrated, we could use the Aruco detections above water to annotate the 6D localization of the objects in the underwater camera even under very turbid conditions. The level of water turbidity is varied by adding clay. We measured the water turbidity, by way of attenuation lengths, to be in the range of 8.3m (clear) to 2.2m (turbid).

Deep learning model for 6D pose estimation We also developed and trained a DL pipeline to predict the 6D pose of the annotated objects. Fig. 5 shows the developed DL pipeline. The DL network includes 4 sub-tasks that combined solves the task of object 6D pose estimation. Class and box prediction sub-networks handle detecting objects with 3D data while handling multiple object categories and instances. The processing time for a single frame is 62.5msec or 16 frames per second using a single GPU (GeForce RTX 2080 Ti, 11GB). In Fig. 6 (a), we show the annotated detections (ground truth) overlaid on the intensity image across turbidities. Fig. 6 (b) shows the estimated pose projected on the intensity image.

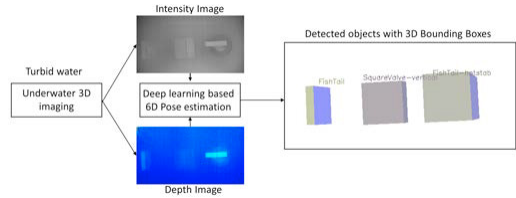


Fig. 5. 6D pose estimation pipeline: The deep learning model takes both intensity and depth image as an input. The network is trained in a single stage to detect and regress the 6D pose under different level of water turbidity.

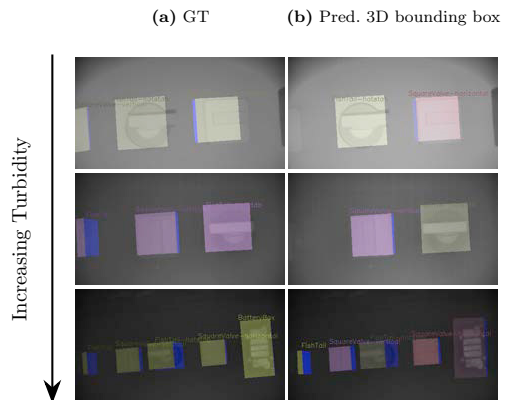


Fig. 6. Qualitative results for detection and pose estimation: Column (a) and (b) shows ground truth and estimated pose projected on the intensity image respectively. The top, middle and last row shows ground truth and predicted pose result for increasing turbidity.

Uncertainty of 6D pose estimation Basing critical autonomous decisions on highly uncertain localization information can potentially lead to catastrophic outcomes that not only risk the success of the autonomous vehicle’s mission, but also endanger human lives. As current DL-approaches to pose estimation do not associate an uncertainty with the 6D pose prediction, we have developed methodology for associating 6D predictions with the inherent model and data uncertainty.

Aruco markers are often used as an effective way for autonomous systems to be able to locate themselves in relation to rigid objects. Here we summarize an approach published in Risholm et al. (2021) where we developed a system for 6D estimation of Aruco markers with associated uncertainties in the challenging underwater environment. A state-of-the-art object detection framework (Efficient-Det) was adapted to predict the corner locations of Aruco markers, while dropout sampling at inference time is used to estimate the predictive 6-DoF pose uncertainty.

We captured a dataset of Aruco markers in a wide variety of turbidities, with ground truth position of the corner

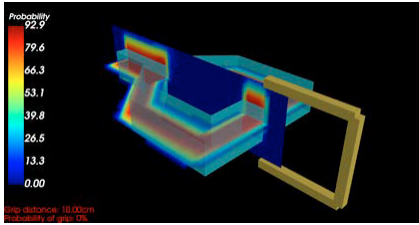


Fig. 7. Gripper in relation to the predicted distribution of the location of the fish-tail handle. As the gripper is moving closer to the handle and closing its grip distance, the probability of having a solid grip will increase according to the uncertainty field.

locations, to train the network to robustly predict the 6D pose. We report translational errors of 2.6cm at low turbidity (8.5m attenuation length) and up to 10.5cm at high turbidities (0.3m attenuation length) while the associated uncertainty (inter-quartile range) ranges from 3.2cm up to 27.9cm. The rotational errors varied from 5.6° to 10.7° with uncertainty of 6.4° to 26.2°. Compared to OpenCV Aruco library with standard detection parameters, we observe that the detection rate falls off rapidly with higher turbidities, while the proposed method provides a detection rate of 100% Risholm et al. (2021).

One direct application of the proposed approach to 6D pose estimation is for autonomous interventions subsea. If an Aruco marker is rigidly placed in relation to a fish-tail handle which an UUV should intervene with, the UUV can automatically position itself and the gripper in relation to the fish-tail. In Fig. 7 we show an example where we have used the pose distribution given by the proposed algorithm to create a probability volume of the location of the fish-tail in relation to the gripper. When the gripper (the yellow model) is closing its grip, we can report the probability of whether it is now gripping the fish-tail. The gripping procedure can be adjusted according to the uncertainty of the pose estimate of the fish-tail. With high uncertainty, the movements can be slower, and the gripper can open up more before closing up the gripper. This will help reduce the risk of damaging the gripper and the fish-tail.

5. AUTOCALIBRATION

This section presents an autocalibration method for localising a vehicle with two or more sensors that we published in Bjerkgeng et al. (2019). In this context, the calibration is the problem of finding the relative positions/orientations and time delays between the different sensors used on an UUV. If this calibration is not accurate, then a sensor fusion for e.g. localization will have poor performance. The method, which we have validated with real data from experiments, was first proposed by Furgale et al. (2013), and uses B-splines to represent vehicle trajectories. The calibration is performed by capturing time series data from all the sensors during vehicle motions, then an optimization problem is solved off-line. Our implementation splits the dataset into 1) identification and 2) validation, as in cross-validation. To avoid over-parametrization, one is used during optimization and the other is used to check the result after the optimization has terminated.

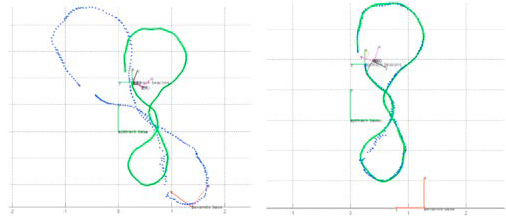


Fig. 8. Vehicle trajectory before and after calibration. Note that on-board sensors and base stations/transponder poses are calibrated at the same time.

The method does not need extra equipment or external references, other than a data series which sufficiently excites all the relevant degrees of freedom. It will calibrate relative poses between on-board and external sensor references, as well as scaling and sensor time-delays. Our main contribution is the integration of cross-validation in the solver. Cross validation enables the user to see if the calibration result is correct, and can detect over-fitting.

The method requires that a user models the bias characteristics of each sensor. Steps building on the cross-validation were taken to automate this step, but no simple solution was found. It was seen that the quality of the calibration result was quite sensitive to bias modeling. If the user has little information about the bias characteristics of e.g. a gyroscope, then this modeling error would be a significant source of error.

6. AUTONOMOUS UNDERWATER GRASPING AND MANIPULATION

Autonomous intervention relies heavily on both software and hardware. An autonomous system requires software that provides intelligent solutions regarding navigation, guidance, perception, pose estimation, grasp position, decision making, and more. In order for this to be possible, its software demands the hardware to provide the necessary sensor information (e.g., joint feedback, sonars, cameras, etc.). However, sensors give additional cost and payload of the system. In the work presented in this section, we have therefore focused on intelligent software solutions with minimal additional sensors to provide intelligent systems capable of autonomous functionalities at low cost. In (Haugaløkken et al., 2020) we developed a grasping procedure to grasp known objects using monocular vision with a small UVMS. This work followed the work presented in Skaldebø et al. (2019), where we presented a large image dataset of the object of interest, an automatic labeling procedure of the image dataset, training of the detection model and the object detection procedure. One of the main goals was to provide an effective solution for object retrieval mission for a small, low cost UVMS. Moreover, in these works we also designed a navigation, guidance, and control system for the vehicle to maintain a desired position relative to an object detected through monocular vision and object detection using the vehicle’s camera. The system proceeded to grasp the object while maintaining the desired position relative to the object and thus provided a task-level autonomous capability. The system was validated in experimental test-

ing with a UVMS consisting of a BlueROV2 vehicle and a SeaArm manipulator. Moreover, in the experimental validation of the system, the gripper was closed manually, making it a semi-autonomous operation. The experiments were conducted in the Marine Cybernetics Laboratory (MC-lab) pool at NTNU, where two out of a total of seven experimental trials were successful. A test was considered successful if the object was grasped. The five unsuccessful experiments failed when the manipulator stopped, either due to singularities or hitting the vehicle because of occlusion. Thus, occlusion avoidance presented itself as the greatest challenge.

Presented with the challenges from (Haugaløkken et al., 2020) we developed a new system in (Skaldebø et al., 2022) with a new manipulator, a SeaArm-2. This work presented fixed-base autonomous underwater grasping of objects using a monocular camera integrated in the manipulator end-effector as the main sensor. In addition, the work presented a novel distance estimator enabling relative distance estimation between object and manipulator without prior knowledge of the object size or shape using only a monocular camera. The distance estimator combines state-of-the-art object detector and tracker systems for 2D footage with the inherent information of end-effector translation through joint manipulation. In this way the system estimated the size and shape of objects of unknown size, e.g. underwater infrastructure, fish, plastic waste, etc., and further calculated the relative distance based on the estimated sizes. The distance estimator was validated in experimental testing using an experimental procedure consisting of the modes 1) Search, 2) Estimate distance, 3) Grasp and 4) Retrieve. In the grasp mode, the system also estimates preferred gripping angle in order to best grasp the object. This is especially relevant for elongated objects, where it might be impossible to grasp it lengthwise. The experiments were performed in the MC-lab pool with the SeaArm-2 fixed at the bottom of the pool. The manipulator was able to find, estimate relative distance, grasp and retrieve the relevant object in 12 out of 12 trials. In the experimental validation the distance estimator found the relative distance between the object and manipulator with a root mean square error of 9.21 mm.

7. AUTOMATED TASK PLANNING AND ACTING

While typically, UUVs operation rely either on teleoperation or pre-scripted missions, neither of those approaches are tractable when mission objectives or environments are continuously and unpredictably evolving. Instead the UUV needs to autonomously determine what needs to be done by balancing its (human) user-specified objectives, and the current context the UUV is in. Goal-directed high level control (Ghallab et al., 2016) provide such capability: it aims at making the *acting* control loop of the robot being informed by *automated planning*.

Acting is the process to identify and execute, at any point, the best course of actions to execute a command, given its “contract” (e.g. maximum duration, allowed battery budget, successful completion, ...). This process is tightly embedded within the UUV control loop and therefore focuses primarily on the current actions at hand rather than how they contribute to the long term objective of

the operation. Those long term objectives are handled by *planning*: it is an inference process that given 1) a model of possible actions, 2) the current state of the actor – in our case the UUV – within the world, and 3) a set of desired objectives; identifies a sequence of actions that should allow the actor to fulfil its objective. The idea is then to give this plan to the *acting* part of the UUV.

Integrating *acting* in *planning* is far from trivial as both processes affect each other but yet both have conflicting temporal constraints and scope. Many different designs have been introduced to address this tension but our work, in following sections, focused on two specific frameworks:

ROSPlan (Cashmore et al., 2015) is a popular framework for integrating AI Planning tools with a Robot Operating System (ROS)-enabled system (Quigley et al., 2009). It supports planners that use the Planning Domain Definition Language (PDDL) standard (Fox and Long, 2003). The *planner* is seen as a service which, when called upon, will search for a complete plan for the mission. This *plan* is then given to the *action* loop that will call upon the *planner* only if the observed world state has deviated from what was planned. The clear functional separation of *planning* and *acting* simplifies the architectural design which greatly contributed to its popularity and the impressively wide range of planning frameworks it supports. Still seeing the planner as a service also means that whenever the system *plans*, the *acting* loop can only wait.

T-REX (Py et al., 2010) on the other hand aims for tighter integration of *planning* and *acting*. It allow multiple *planning* and *acting* decision loops – called *reactors* – to be composed by having a well-defined model of ownership of *state variable* (each state variable is maintained by one and only one reactor, that declare it *internal*, while others can only receive *observation* updates and request future *goal* values to this variable) along with requiring every *reactor* planning to be suspended at a specified “tick” rate in order for each to identify their *internal* state for other reactors to consume. This blurs the line between *planning* and *acting* as each reactor do both concurrently, but at the cost of a much more complex integration of any planning framework. Therefore, as far as we know, the only planner it fully supports is the EUROPA planning framework (Frank and Jónsson, 2003).

7.1 Automated task replanning with ROSPlan

In this section, we contribute a previously unpublished strategy to address the problem of automated replanning for UUV missions in dynamically changing underwater environments. We explicitly explored enhancements in the replanning approach during the execution of inspection, maintenance and repair (IMR) tasks as new goals can emerge during IMR mission. We present a method that analyzes the trade-off between continuing with the current plan and deferring new goals until later, versus performing a full replanning that incorporates the new goal. Our strategy also considers the urgency of the goals to be achieved, alongside the resources available to the mission. Implementation and testing of this approach was done within the ROSplan framework. Simulated action components are

set up with code which includes the name and duration of each action. Thus, we can test planning for a UUV without having to engage a physics-based simulator in the testing process.

The proposed architecture of our planning system is shown in Fig. 9. An operator provides a description of the world model the UUV is operating in and of a set of mission goals to be achieved. These are stored in a Knowledge Base. An initial problem is generated by a Problem Generator and fed to a planner which produces an initial UUV mission plan. The plan is then executed and a module called Dispatcher updates the Knowledge Base when the actions of the UUV are achieved or failed. If the operator decides to add new goals while the aforementioned thread of the initial plan is executed, a new parallel thread handles the updated problem and generates a new plan. We have developed and inserted a Replanning Node that evaluates the revised plan, and if it meets the criteria of the human operator, it then triggers a new dispatch. Hence, our approach allows the parallel evaluation of candidate new plans that adapt to goals that emerge during the mission execution. This is without interrupting the current mission and while respecting the overall mission resource constraints, alongside possible priorities of certain actions in the original plan.

The features of the Replanning Node sum up to:

- It prompts the operator to add possible new goals or other facts if necessary.
- It allows the operator to select which action or clusters of actions need to have been completed before launching a new plan dispatch.
- It allows to update the knowledge base with the new goals and accordingly facilitates new plans that also consider the operator-defined priorities.
- It offers functionality that cancels the current plan and updates the dispatcher with the new plan based on the updated knowledge base and metrics of urgency of the mission actions.

It is noted that while replanning iterations take place, the ongoing mission as commanded by ROSPlan may continue.

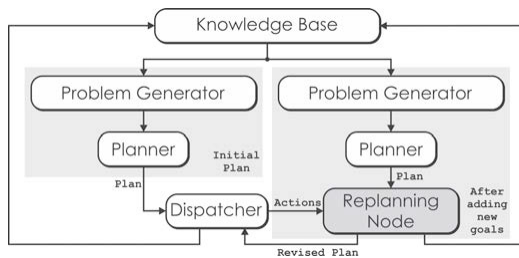


Fig. 9. The Replanning Node in the proposed architecture of the planning system.

7.2 Comparison of AI Planning frameworks

In Xue and Lekkas (2020) we conducted a comparison between T-REX and ROSPlan (without the extensions in Sec. 7.1), in the context of a subsea scenario formulated

in collaboration with Oceaneering, considering the design and capabilities of their Freedom Autonomous Vehicle. Compared to past efforts, where mission planning pertained mostly to path planning tasks, such as dynamic waypoint (re)planning, we considered a more detailed scenario in terms of possible states, tasks and actions, including, for instance, inspection of certain components (valves, pipes), doing self-diagnosis before leaving the docking station, installing tools depending on the tasks, and so on.

The subsea scenario in Xue and Lekkas (2020) took into account several of the novel features of the Freedom vehicle, such as the existence of an underwater warehouse, where the vehicle can switch tools, and a docking station, where it can charge and communicate with a shore control centre. The mission itself included the following steps: First perform a pipeline inspection, then an inspection of Valves No. 1 and 3, located on Panel 1, and finally intervention of Valves No. 3 and 4, located on Panel 2. To accomplish the mission, numerous actions had to be planned, including the vehicle performing self-diagnosis, undocking, moving from one location to another, approaching a component, performing inspection and/or intervention, and others. In addition, we introduced an unplanned event in order to test the replanning capability of each framework; When the vehicle inspects Valve No.2 it finds it is not in the right setting and therefore intervention is required. It should be noted that only the action planning aspect of the problem was dealt with, without simulating the complete guidance, navigation and control system of the vehicle.

Our results showed that ROSPlan planned for 12 unnecessary actions compared to T-REX (56 vs. 44 actions) to complete the task. The 12 additional actions were not implemented (since ROSPlan updated the plan in the next working cycle), but T-REX avoided planning them altogether, hence demonstrating a more reactive and computationally efficient behavior. To summarize further, it can also be advantageous to use ROSPlan for planning and execution UUV tasks with less stringent requirements to reactive behaviors, as, e.g., ROSPlan has a rather large user group, supports different types of planners and is well documented.

8. CONCLUSION AND FUTURE WORK

We have presented results ranging from vehicle-level autonomy (e.g., perception, autocalibration) to mission-level autonomy (task planning). We have also focused on robustness as, e.g., perception methods tested in varying turbidity. These aspects will be important in enabling fully autonomous UUV missions in the cases where close (human) operator involvement is not possible or desirable. Further work will focus on further increasing robustness of methods and integrating capabilities into autonomous missions with a suitable level of operator involvement.

ACKNOWLEDGEMENTS

The authors acknowledge the valuable input from the industry partners of the SEAVENTION project: Equinor, TechnipFMC, IKM and Oceaneering.

REFERENCES

Antonelli, G. (2014). *Underwater Robotics*. Springer.

- Bjelland, H.V., Føre, M., Lader, P., Kristiansen, D., Holmen, I.M., Fredheim, A., Grøtli, E.I., Fathi, D.E., Oppedal, F., Utne, I.B., et al. (2015). Exposed aquaculture in Norway. In *OCEANS 2015-MTS/IEEE Washington*, 1–10. IEEE.
- Bjerkeng, M., Transeth, A., and Myhre, T. (2019). Spatial and temporal autocalibration for underwater vehicles. *IFAC-PapersOnLine*, 52(21), 260–264.
- BlueLogic (2019). Subsea Docking Station (SDS). URL <https://www.bluelogic.no/news-and-media/subsea-docking-station-sds->.
- Cashmore, M., Fox, M., Long, D., Magazzeni, D., Ridder, B., Carrera, A., Palomeras, N., Hurtos, N., and Carreras, M. (2015). Rosplan: Planning in the robot operating system. In *Proc. Int. Conf. on Automated Planning and Scheduling*, volume 25.
- Fox, M. and Long, D. (2003). PDDL2.1: an extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20, 61–124.
- Frank, J. and Jónsson, A. (2003). Constraint-based attribute and interval planning. *Constraints*, 8(4).
- Furgale, P., Rehder, J., and Siegwart, R. (2013). Unified temporal and spatial calibration for multi-sensor systems. In *Proc. 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1280–1286. IEEE.
- Gancet, J., Urbina, D., Letier, P., Ilzokvitz, M., Weiss, P., Gauch, F., Antonelli, G., Indiveri, G., Casalino, G., Birk, A., Pflingsthor, M.F., Calinon, S., Tanwani, A., Turetta, A., Walen, C., and Guilpain, L. (2015). Dexrov: Dexterous undersea inspection and maintenance in presence of communication latencies. *IFAC-PapersOnLine*, 48(2), 218–223. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles.
- Ghallab, M., Nau, D., and Traverso, P. (2016). *Automated planning and acting*. Cambridge University Press.
- Haugaløkken, B.O.A., Skaldebø, M.B., and Schjølberg, I. (2020). Monocular vision-based gripping of objects. *Robotics and Autonomous Systems*, 131, 103–589.
- Huang, H., Tang, Q., Li, J., Zhang, W., Bao, X., Zhu, H., and Wang, G. (2020). A review on underwater autonomous environmental perception and target grasp, the challenge of robotic organism capture. *Ocean Engineering*, 195, 106–644.
- IKM Subsea (2019). Onshore Control Center. URL <https://www.ikm.com/ikm-subsea/products/services/onshore-control-center>.
- Marani, G., Choi, S.K., and Yuh, J. (2009). Underwater autonomous manipulation for intervention missions auvs. *Ocean Engineering*, 36(1), 15–23.
- MarketResearchFuture.com (2020). Underwater Robotics Market Research Report. Technical report. URL <https://www.marketresearchfuture.com/reports/underwater-robotics-market-7605>.
- Mohammed, A., Kvam, J., T. Thielemann, J., Haugholt, K.H., and Risholm, P. (2021). 6D pose estimation for subsea intervention in turbid waters. *Electronics*, 10(19).
- Nhede, N. (2021). Global offshore wind market to hit \$56.8bn by 2026. *Power Eng. Int.* URL <https://www.powerengineeringint.com/renewables/wind/global-offshore-wind-market-to-hit-56-8bn-by-2026/>.
- Oceaneering (2021). Oceaneering and BP Conduct First ROV Remote Piloting Operation Offshore West of Shetland. URL <https://www.oceaneering.com/oceaneering-and-bp-conduct-first-rov-remote-piloting-operation-offshore-west-of-shetland/>.
- Oceaneering (2022). Freedom™ Autonomous Vehicle. URL <https://www.oceaneering.com/rov-services/next-generation-subsea-vehicles/freedom/>.
- Py, F., Rajan, K., and McGann, C. (2010). A systematic agent framework for situated autonomous systems.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al. (2009). ROS: An open-source robot operating system. In *Proc. ICRA workshop on open source software*, volume 3, 5. Japan.
- Risholm, P., Ivarsen, P.O., Haugholt, K.H., and Mohammed, A. (2021). Underwater marker-based pose-estimation with associated uncertainty. In *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV) Workshops*, 3713–3721.
- Saipem (2019). Saipem signs pioneering contract to deliver Hydrone-based technology to Equinor. URL <https://www.saipem.com/en/media/news/2019-10-02/saipem-signs-pioneering-contract-deliver-hydrone-based-technology-equinor>.
- Schjølberg, I., Gjersvik, T.B., Transeth, A.A., and Utne, I.B. (2016). Next generation subsea inspection, maintenance and repair operations. *IFAC-PapersOnLine*, 49(23), 434–439. Proc. 10th IFAC Conf. on Control Applications in Marine Systems (CAMS) 2016.
- Schjølberg, I. and Utne, I.B. (2015). Towards autonomy in ROV operations. *IFAC-PapersOnLine*, 48(2), 183–188. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV) 2015.
- Simetti, E. (2020). Autonomous underwater intervention. *Current Robotics Reports*, 1, 117–122.
- Simetti, E., Wanderling, F., Torelli, S., Bibuli, M., Odetti, A., Bruzzone, G., Rizzini, D.L., Aleotti, J., Palli, G., Moriello, L., and Scarica, U. (2018). Autonomous underwater intervention: Experimental results of the maris project. *IEEE J Oceanic Eng.*, 43(3), 620–639.
- Simetti, E., Casalino, G., Torelli, S., Sperindé, A., and Turetta, A. (2014). Floating underwater manipulation: Developed control methodology and experimental validation within the trident project. *Journal of Field Robotics*, 31(3), 364–385.
- Sivčev, S., Coleman, J., Omerdić, E., Dooly, G., and Toal, D. (2018). Underwater manipulators: A review. *Ocean Engineering*, 163, 431 – 450.
- Skaldebø, M., Haugaløkken, B.O.A., and Schjølberg, I. (2022). Autonomous underwater grasping using a novel vision-based distance estimator. Submitted to *Journal of robotics and autonomous systems*.
- Skaldebø, M., Haugaløkken, B.O.A., and Schjølberg, I. (2019). Dynamic positioning of an underwater vehicle using monocular vision-based object detection with machine learning. In *OCEANS 2019*, 1–9.
- Topini, E., Bucci, A., Gelli, J., Topini, A., Ridolfi, A., and Allotta, B. (2021). Advanced underwater manipulation systems: an overview of the suono project. In *OCEANS 2021: San Diego – Porto*, 1–7.
- Xue, L. and Lekkas, A.M. (2020). Comparison of ai planning frameworks for underwater intervention drones. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, 1–9.

Article 10

System integration of underwater vehicle manipulator system (UVMS) for autonomous grasping

Martin B. Skaldebø, Ingrid Schjølberg, Bent O. A. Haugaløkken
Accepted at Journal of Marine Science and Technology (January 2023)

This paper is awaiting publication and is not included

Part III

Previous PhD Theses Published at the Department of Marine Technology

**Previous PhD theses published at the Department of Marine Technology
(earlier: Faculty of Marine Technology)
NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY**

Report No.	Author	Title
	Kavlie, Dag	Optimization of Plane Elastic Grillage, 1967
	Hansen, Hans R.	Man-Machine Communication and Data-Storage Methods in Ship Structural Design, 1971
	Gisvold, Kaare M.	A Method for non-linear mixed -integer programming and its Application to Design Problems, 1971
	Lund, Sverre	Tanker Frame Optimization by means of SUMT-Transformation and Behaviour Models, 1971
	Vinje, Tor	On Vibration of Spherical Shells Interacting with Fluid, 1972
	Lorentz, Jan D.	Tank Arrangement for Crude Oil Carriers in Accordance with the new Anti-Pollution Regulations, 1975
	Carlsen, Carl A.	Computer-Aided Design of Tanker Structures, 1975
	Larsen, Carl M.	Static and Dynamic Analysis of Offshore Pipelines during Installation, 1976
UR-79-01	Brigt Hatlestad, MK	The finite element method used in a fatigue evaluation of fixed offshore platforms. (Dr.Ing. Thesis)
UR-79-02	Erik Pettersen, MK	Analysis and design of cellular structures. (Dr.Ing. Thesis)
UR-79-03	Sverre Valsgård, MK	Finite difference and finite element methods applied to nonlinear analysis of plated structures. (Dr.Ing. Thesis)
UR-79-04	Nils T. Nordsve, MK	Finite element collapse analysis of structural members considering imperfections and stresses due to fabrication. (Dr.Ing. Thesis)
UR-79-05	Ivar J. Fylling, MK	Analysis of towline forces in ocean towing systems. (Dr.Ing. Thesis)
UR-79- x	Finn Gunnar Nielsen, MH	Hydrodynamic problems related to oil barriers for offshore application
UR-80-06	Nils Sandsmark, MM	Analysis of Stationary and Transient Heat Conduction by the Use of the Finite Element Method. (Dr.Ing. Thesis)
UR-80-09	Sverre Haver, MK	Analysis of uncertainties related to the stochastic modeling of ocean waves. (Dr.Ing. Thesis)

UR-81-15	Odland, Jonas	On the Strength of welded Ring stiffened cylindrical Shells primarily subjected to axial Compression
UR-82-17	Engesvik, Knut	Analysis of Uncertainties in the fatigue Capacity of Welded Joints
UR-82-18	Rye, Henrik	Ocean wave groups
UR-83-30	Eide, Oddvar Inge	On Cumulative Fatigue Damage in Steel Welded Joints
UR-83-33	Mo, Olav	Stochastic Time Domain Analysis of Slender Offshore Structures
UR-83-34	Amdahl, Jørgen	Energy absorption in Ship-platform impacts
UR-84-37	Mørch, Morten	Motions and mooring forces of semi submersibles as determined by full-scale measurements and theoretical analysis
UR-84-38	Soares, C. Guedes	Probabilistic models for load effects in ship structures
UR-84-39	Aarsnes, Jan V.	Current forces on ships
UR-84-40	Czujko, Jerzy	Collapse Analysis of Plates subjected to Biaxial Compression and Lateral Load
UR-85-46	Alf G. Engseth, MK	Finite element collapse analysis of tubular steel offshore structures. (Dr.Ing. Thesis)
UR-86-47	Dengody Sheshappa, MP	A Computer Design Model for Optimizing Fishing Vessel Designs Based on Techno-Economic Analysis. (Dr.Ing. Thesis)
UR-86-48	Vidar Aanesland, MH	A Theoretical and Numerical Study of Ship Wave Resistance. (Dr.Ing. Thesis)
UR-86-49	Heinz-Joachim Wessel, MK	Fracture Mechanics Analysis of Crack Growth in Plate Girders. (Dr.Ing. Thesis)
UR-86-50	Jon Taby, MK	Ultimate and Post-ultimate Strength of Dented Tubular Members. (Dr.Ing. Thesis)
UR-86-51	Walter Lian, MH	A Numerical Study of Two-Dimensional Separated Flow Past Bluff Bodies at Moderate KC-Numbers. (Dr.Ing. Thesis)
UR-86-52	Bjørn Sortland, MH	Force Measurements in Oscillating Flow on Ship Sections and Circular Cylinders in a U-Tube Water Tank. (Dr.Ing. Thesis)
UR-86-53	Kurt Strand, MM	A System Dynamic Approach to One-dimensional Fluid Flow. (Dr.Ing. Thesis)
UR-86-54	Arne Edvin Løken, MH	Three Dimensional Second Order Hydrodynamic Effects on Ocean Structures in Waves. (Dr.Ing. Thesis)
UR-86-55	Sigurd Falch, MH	A Numerical Study of Slamming of Two-

		Dimensional Bodies. (Dr.Ing. Thesis)
UR-87-56	Arne Braathen, MH	Application of a Vortex Tracking Method to the Prediction of Roll Damping of a Two-Dimension Floating Body. (Dr.Ing. Thesis)
UR-87-57	Bernt Leira, MK	Gaussian Vector Processes for Reliability Analysis involving Wave-Induced Load Effects. (Dr.Ing. Thesis)
UR-87-58	Magnus Småvik, MM	Thermal Load and Process Characteristics in a Two-Stroke Diesel Engine with Thermal Barriers (in Norwegian). (Dr.Ing. Thesis)
MTA-88-59	Bernt Arild Bremdal, MP	An Investigation of Marine Installation Processes – A Knowledge - Based Planning Approach. (Dr.Ing. Thesis)
MTA-88-60	Xu Jun, MK	Non-linear Dynamic Analysis of Space-framed Offshore Structures. (Dr.Ing. Thesis)
MTA-89-61	Gang Miao, MH	Hydrodynamic Forces and Dynamic Responses of Circular Cylinders in Wave Zones. (Dr.Ing. Thesis)
MTA-89-62	Martin Greenhow, MH	Linear and Non-Linear Studies of Waves and Floating Bodies. Part I and Part II. (Dr.Tech. Thesis)
MTA-89-63	Chang Li, MH	Force Coefficients of Spheres and Cubes in Oscillatory Flow with and without Current. (Dr.Ing. Thesis)
MTA-89-64	Hu Ying, MP	A Study of Marketing and Design in Development of Marine Transport Systems. (Dr.Ing. Thesis)
MTA-89-65	Arild Jæger, MH	Seakeeping, Dynamic Stability and Performance of a Wedge Shaped Planing Hull. (Dr.Ing. Thesis)
MTA-89-66	Chan Siu Hung, MM	The dynamic characteristics of tilting-pad bearings
MTA-89-67	Kim Wikstrøm, MP	Analysis av projekteringen for ett offshore projekt. (Licenciat-avhandling)
MTA-89-68	Jiao Guoyang, MK	Reliability Analysis of Crack Growth under Random Loading, considering Model Updating. (Dr.Ing. Thesis)
MTA-89-69	Arnt Olufsen, MK	Uncertainty and Reliability Analysis of Fixed Offshore Structures. (Dr.Ing. Thesis)
MTA-89-70	Wu Yu-Lin, MR	System Reliability Analyses of Offshore Structures using improved Truss and Beam Models. (Dr.Ing. Thesis)
MTA-90-71	Jan Roger Hoff, MH	Three-dimensional Green function of a vessel with forward speed in waves. (Dr.Ing. Thesis)
MTA-90-72	Rong Zhao, MH	Slow-Drift Motions of a Moored Two-Dimensional Body in Irregular Waves. (Dr.Ing. Thesis)
MTA-90-73	Atle Minsaas, MP	Economical Risk Analysis. (Dr.Ing. Thesis)

MTA-90-74	Knut-Aril Farnes, MK	Long-term Statistics of Response in Non-linear Marine Structures. (Dr.Ing. Thesis)
MTA-90-75	Torbjørn Sotberg, MK	Application of Reliability Methods for Safety Assessment of Submarine Pipelines. (Dr.Ing. Thesis)
MTA-90-76	Zeuthen, Steffen, MP	SEAMAID. A computational model of the design process in a constraint-based logic programming environment. An example from the offshore domain. (Dr.Ing. Thesis)
MTA-91-77	Haagensen, Sven, MM	Fuel Dependant Cyclic Variability in a Spark Ignition Engine - An Optical Approach. (Dr.Ing. Thesis)
MTA-91-78	Løland, Geir, MH	Current forces on and flow through fish farms. (Dr.Ing. Thesis)
MTA-91-79	Hoen, Christopher, MK	System Identification of Structures Excited by Stochastic Load Processes. (Dr.Ing. Thesis)
MTA-91-80	Haugen, Stein, MK	Probabilistic Evaluation of Frequency of Collision between Ships and Offshore Platforms. (Dr.Ing. Thesis)
MTA-91-81	Sødahl, Nils, MK	Methods for Design and Analysis of Flexible Risers. (Dr.Ing. Thesis)
MTA-91-82	Ornberg, Harald, MK	Non-linear Response Analysis of Floating Fish Farm Systems. (Dr.Ing. Thesis)
MTA-91-83	Marley, Mark J., MK	Time Variant Reliability under Fatigue Degradation. (Dr.Ing. Thesis)
MTA-91-84	Krokstad, Jørgen R., MH	Second-order Loads in Multidirectional Seas. (Dr.Ing. Thesis)
MTA-91-85	Molteberg, Gunnar A., MM	The Application of System Identification Techniques to Performance Monitoring of Four Stroke Turbocharged Diesel Engines. (Dr.Ing. Thesis)
MTA-92-86	Mørch, Hans Jørgen Bjelke, MH	Aspects of Hydrofoil Design: with Emphasis on Hydrofoil Interaction in Calm Water. (Dr.Ing. Thesis)
MTA-92-87	Chan Siu Hung, MM	Nonlinear Analysis of Rotordynamic Instabilities in Highspeed Turbomachinery. (Dr.Ing. Thesis)
MTA-92-88	Bessason, Bjarni, MK	Assessment of Earthquake Loading and Response of Seismically Isolated Bridges. (Dr.Ing. Thesis)
MTA-92-89	Langli, Geir, MP	Improving Operational Safety through exploitation of Design Knowledge - an investigation of offshore platform safety. (Dr.Ing. Thesis)
MTA-92-90	Sævik, Svein, MK	On Stresses and Fatigue in Flexible Pipes. (Dr.Ing. Thesis)
MTA-92-91	Ask, Tor Ø., MM	Ignition and Flame Growth in Lean Gas-Air Mixtures. An Experimental Study with a Schlieren

		System. (Dr.Ing. Thesis)
MTA-86-92	Hessen, Gunnar, MK	Fracture Mechanics Analysis of Stiffened Tubular Members. (Dr.Ing. Thesis)
MTA-93-93	Steinebach, Christian, MM	Knowledge Based Systems for Diagnosis of Rotating Machinery. (Dr.Ing. Thesis)
MTA-93-94	Dalane, Jan Inge, MK	System Reliability in Design and Maintenance of Fixed Offshore Structures. (Dr.Ing. Thesis)
MTA-93-95	Steen, Sverre, MH	Cobblestone Effect on SES. (Dr.Ing. Thesis)
MTA-93-96	Karunakaran, Daniel, MK	Nonlinear Dynamic Response and Reliability Analysis of Drag-dominated Offshore Platforms. (Dr.Ing. Thesis)
MTA-93-97	Hagen, Arnulf, MP	The Framework of a Design Process Language. (Dr.Ing. Thesis)
MTA-93-98	Nordrik, Rune, MM	Investigation of Spark Ignition and Autoignition in Methane and Air Using Computational Fluid Dynamics and Chemical Reaction Kinetics. A Numerical Study of Ignition Processes in Internal Combustion Engines. (Dr.Ing. Thesis)
MTA-94-99	Passano, Elizabeth, MK	Efficient Analysis of Nonlinear Slender Marine Structures. (Dr.Ing. Thesis)
MTA-94-100	Kvålsvold, Jan, MH	Hydroelastic Modelling of Wetdeck Slamming on Multihull Vessels. (Dr.Ing. Thesis)
MTA-94-102	Bech, Sidsel M., MK	Experimental and Numerical Determination of Stiffness and Strength of GRP/PVC Sandwich Structures. (Dr.Ing. Thesis)
MTA-95-103	Paulsen, Hallvard, MM	A Study of Transient Jet and Spray using a Schlieren Method and Digital Image Processing. (Dr.Ing. Thesis)
MTA-95-104	Hovde, Geir Olav, MK	Fatigue and Overload Reliability of Offshore Structural Systems, Considering the Effect of Inspection and Repair. (Dr.Ing. Thesis)
MTA-95-105	Wang, Xiaozhi, MK	Reliability Analysis of Production Ships with Emphasis on Load Combination and Ultimate Strength. (Dr.Ing. Thesis)
MTA-95-106	Ulstein, Tore, MH	Nonlinear Effects of a Flexible Stem Seal Bag on Cobblestone Oscillations of an SES. (Dr.Ing. Thesis)
MTA-95-107	Solaas, Frøydis, MH	Analytical and Numerical Studies of Sloshing in Tanks. (Dr.Ing. Thesis)
MTA-95-108	Hellan, Øyvind, MK	Nonlinear Pushover and Cyclic Analyses in Ultimate Limit State Design and Reassessment of Tubular Steel Offshore Structures. (Dr.Ing. Thesis)
MTA-95-109	Hermundstad, Ole A., MK	Theoretical and Experimental Hydroelastic Analysis of High Speed Vessels. (Dr.Ing. Thesis)

MTA-96-110	Bratland, Anne K., MH	Wave-Current Interaction Effects on Large-Volume Bodies in Water of Finite Depth. (Dr.Ing. Thesis)
MTA-96-111	Herfjord, Kjell, MH	A Study of Two-dimensional Separated Flow by a Combination of the Finite Element Method and Navier-Stokes Equations. (Dr.Ing. Thesis)
MTA-96-112	Æsøy, Vilmar, MM	Hot Surface Assisted Compression Ignition in a Direct Injection Natural Gas Engine. (Dr.Ing. Thesis)
MTA-96-113	Eknes, Monika L., MK	Escalation Scenarios Initiated by Gas Explosions on Offshore Installations. (Dr.Ing. Thesis)
MTA-96-114	Erikstad, Stein O., MP	A Decision Support Model for Preliminary Ship Design. (Dr.Ing. Thesis)
MTA-96-115	Pedersen, Egil, MH	A Nautical Study of Towed Marine Seismic Streamer Cable Configurations. (Dr.Ing. Thesis)
MTA-97-116	Moksnes, Paul O., MM	Modelling Two-Phase Thermo-Fluid Systems Using Bond Graphs. (Dr.Ing. Thesis)
MTA-97-117	Halse, Karl H., MK	On Vortex Shedding and Prediction of Vortex-Induced Vibrations of Circular Cylinders. (Dr.Ing. Thesis)
MTA-97-118	Igland, Ragnar T., MK	Reliability Analysis of Pipelines during Laying, considering Ultimate Strength under Combined Loads. (Dr.Ing. Thesis)
MTA-97-119	Pedersen, Hans-P., MP	Levendefiskteknologi for fiskefartøy. (Dr.Ing. Thesis)
MTA-98-120	Vikestad, Kyrre, MK	Multi-Frequency Response of a Cylinder Subjected to Vortex Shedding and Support Motions. (Dr.Ing. Thesis)
MTA-98-121	Azadi, Mohammad R. E., MK	Analysis of Static and Dynamic Pile-Soil-Jacket Behaviour. (Dr.Ing. Thesis)
MTA-98-122	Ulltang, Terje, MP	A Communication Model for Product Information. (Dr.Ing. Thesis)
MTA-98-123	Torbergsen, Erik, MM	Impeller/Diffuser Interaction Forces in Centrifugal Pumps. (Dr.Ing. Thesis)
MTA-98-124	Hansen, Edmond, MH	A Discrete Element Model to Study Marginal Ice Zone Dynamics and the Behaviour of Vessels Moored in Broken Ice. (Dr.Ing. Thesis)
MTA-98-125	Videiro, Paulo M., MK	Reliability Based Design of Marine Structures. (Dr.Ing. Thesis)
MTA-99-126	Mainçon, Philippe, MK	Fatigue Reliability of Long Welds Application to Titanium Risers. (Dr.Ing. Thesis)
MTA-99-127	Haugen, Elin M., MH	Hydroelastic Analysis of Slamming on Stiffened Plates with Application to Catamaran Wetdecks. (Dr.Ing. Thesis)
MTA-99-	Langhelle, Nina K., MK	Experimental Validation and Calibration of

128		Nonlinear Finite Element Models for Use in Design of Aluminium Structures Exposed to Fire. (Dr.Ing. Thesis)
MTA-99-129	Berstad, Are J., MK	Calculation of Fatigue Damage in Ship Structures. (Dr.Ing. Thesis)
MTA-99-130	Andersen, Trond M., MM	Short Term Maintenance Planning. (Dr.Ing. Thesis)
MTA-99-131	Tveiten, Bård Wathne, MK	Fatigue Assessment of Welded Aluminium Ship Details. (Dr.Ing. Thesis)
MTA-99-132	Søreide, Fredrik, MP	Applications of underwater technology in deep water archaeology. Principles and practice. (Dr.Ing. Thesis)
MTA-99-133	Tønnessen, Rune, MH	A Finite Element Method Applied to Unsteady Viscous Flow Around 2D Blunt Bodies With Sharp Corners. (Dr.Ing. Thesis)
MTA-99-134	Elvekrok, Dag R., MP	Engineering Integration in Field Development Projects in the Norwegian Oil and Gas Industry. The Supplier Management of Norne. (Dr.Ing. Thesis)
MTA-99-135	Fagerholt, Kjetil, MP	Optimeringsbaserte Metoder for Ruteplanlegging innen skipsfart. (Dr.Ing. Thesis)
MTA-99-136	Bysveen, Marie, MM	Visualization in Two Directions on a Dynamic Combustion Rig for Studies of Fuel Quality. (Dr.Ing. Thesis)
MTA-2000-137	Storteig, Eskild, MM	Dynamic characteristics and leakage performance of liquid annular seals in centrifugal pumps. (Dr.Ing. Thesis)
MTA-2000-138	Sagli, Gro, MK	Model uncertainty and simplified estimates of long term extremes of hull girder loads in ships. (Dr.Ing. Thesis)
MTA-2000-139	Tronstad, Harald, MK	Nonlinear analysis and design of cable net structures like fishing gear based on the finite element method. (Dr.Ing. Thesis)
MTA-2000-140	Kroneberg, André, MP	Innovation in shipping by using scenarios. (Dr.Ing. Thesis)
MTA-2000-141	Haslum, Herbjørn Alf, MH	Simplified methods applied to nonlinear motion of spar platforms. (Dr.Ing. Thesis)
MTA-2001-142	Samdal, Ole Johan, MM	Modelling of Degradation Mechanisms and Stressor Interaction on Static Mechanical Equipment Residual Lifetime. (Dr.Ing. Thesis)
MTA-2001-143	Baarholm, Rolf Jarle, MH	Theoretical and experimental studies of wave impact underneath decks of offshore platforms. (Dr.Ing. Thesis)
MTA-2001-144	Wang, Lihua, MK	Probabilistic Analysis of Nonlinear Wave-induced Loads on Ships. (Dr.Ing. Thesis)
MTA-2001-145	Kristensen, Odd H. Holt, MK	Ultimate Capacity of Aluminium Plates under Multiple Loads, Considering HAZ Properties.

(Dr.Ing. Thesis)

MTA-2001-146	Greco, Marilena, MH	A Two-Dimensional Study of Green-Water Loading. (Dr.Ing. Thesis)
MTA-2001-147	Heggelund, Svein E., MK	Calculation of Global Design Loads and Load Effects in Large High Speed Catamarans. (Dr.Ing. Thesis)
MTA-2001-148	Babalola, Olusegun T., MK	Fatigue Strength of Titanium Risers – Defect Sensitivity. (Dr.Ing. Thesis)
MTA-2001-149	Mohammed, Abuu K., MK	Nonlinear Shell Finite Elements for Ultimate Strength and Collapse Analysis of Ship Structures. (Dr.Ing. Thesis)
MTA-2002-150	Holmedal, Lars E., MH	Wave-current interactions in the vicinity of the sea bed. (Dr.Ing. Thesis)
MTA-2002-151	Rognebakke, Olav F., MH	Sloshing in rectangular tanks and interaction with ship motions. (Dr.Ing. Thesis)
MTA-2002-152	Lader, Pål Furset, MH	Geometry and Kinematics of Breaking Waves. (Dr.Ing. Thesis)
MTA-2002-153	Yang, Qinzhen, MH	Wash and wave resistance of ships in finite water depth. (Dr.Ing. Thesis)
MTA-2002-154	Melhus, Øyvinn, MM	Utilization of VOC in Diesel Engines. Ignition and combustion of VOC released by crude oil tankers. (Dr.Ing. Thesis)
MTA-2002-155	Ronæss, Marit, MH	Wave Induced Motions of Two Ships Advancing on Parallel Course. (Dr.Ing. Thesis)
MTA-2002-156	Økland, Ole D., MK	Numerical and experimental investigation of whipping in twin hull vessels exposed to severe wet deck slamming. (Dr.Ing. Thesis)
MTA-2002-157	Ge, Chunhua, MK	Global Hydroelastic Response of Catamarans due to Wet Deck Slamming. (Dr.Ing. Thesis)
MTA-2002-158	Byklum, Eirik, MK	Nonlinear Shell Finite Elements for Ultimate Strength and Collapse Analysis of Ship Structures. (Dr.Ing. Thesis)
IMT-2003-1	Chen, Haibo, MK	Probabilistic Evaluation of FPSO-Tanker Collision in Tandem Offloading Operation. (Dr.Ing. Thesis)
IMT-2003-2	Skaugset, Kjetil Bjørn, MK	On the Suppression of Vortex Induced Vibrations of Circular Cylinders by Radial Water Jets. (Dr.Ing. Thesis)
IMT-2003-3	Chezian, Muthu	Three-Dimensional Analysis of Slamming. (Dr.Ing. Thesis)
IMT-2003-4	Buhaug, Øyvind	Deposit Formation on Cylinder Liner Surfaces in Medium Speed Engines. (Dr.Ing. Thesis)
IMT-2003-5	Tregde, Vidar	Aspects of Ship Design: Optimization of Aft Hull with Inverse Geometry Design. (Dr.Ing. Thesis)

IMT-2003-6	Wist, Hanne Therese	Statistical Properties of Successive Ocean Wave Parameters. (Dr.Ing. Thesis)
IMT-2004-7	Ransau, Samuel	Numerical Methods for Flows with Evolving Interfaces. (Dr.Ing. Thesis)
IMT-2004-8	Soma, Torkel	Blue-Chip or Sub-Standard. A data interrogation approach of identity safety characteristics of shipping organization. (Dr.Ing. Thesis)
IMT-2004-9	Ersdal, Svein	An experimental study of hydrodynamic forces on cylinders and cables in near axial flow. (Dr.Ing. Thesis)
IMT-2005-10	Brodtkorb, Per Andreas	The Probability of Occurrence of Dangerous Wave Situations at Sea. (Dr.Ing. Thesis)
IMT-2005-11	Yttervik, Rune	Ocean current variability in relation to offshore engineering. (Dr.Ing. Thesis)
IMT-2005-12	Fredheim, Arne	Current Forces on Net-Structures. (Dr.Ing. Thesis)
IMT-2005-13	Heggernes, Kjetil	Flow around marine structures. (Dr.Ing. Thesis)
IMT-2005-14	Fouques, Sebastien	Lagrangian Modelling of Ocean Surface Waves and Synthetic Aperture Radar Wave Measurements. (Dr.Ing. Thesis)
IMT-2006-15	Holm, Håvard	Numerical calculation of viscous free surface flow around marine structures. (Dr.Ing. Thesis)
IMT-2006-16	Bjørheim, Lars G.	Failure Assessment of Long Through Thickness Fatigue Cracks in Ship Hulls. (Dr.Ing. Thesis)
IMT-2006-17	Hansson, Lisbeth	Safety Management for Prevention of Occupational Accidents. (Dr.Ing. Thesis)
IMT-2006-18	Zhu, Xinying	Application of the CIP Method to Strongly Nonlinear Wave-Body Interaction Problems. (Dr.Ing. Thesis)
IMT-2006-19	Reite, Karl Johan	Modelling and Control of Trawl Systems. (Dr.Ing. Thesis)
IMT-2006-20	Smogeli, Øyvind Notland	Control of Marine Propellers. From Normal to Extreme Conditions. (Dr.Ing. Thesis)
IMT-2007-21	Storhaug, Gaute	Experimental Investigation of Wave Induced Vibrations and Their Effect on the Fatigue Loading of Ships. (Dr.Ing. Thesis)
IMT-2007-22	Sun, Hui	A Boundary Element Method Applied to Strongly Nonlinear Wave-Body Interaction Problems. (PhD Thesis, CeSOS)
IMT-2007-23	Rustad, Anne Marthine	Modelling and Control of Top Tensioned Risers. (PhD Thesis, CeSOS)
IMT-2007-24	Johansen, Vegar	Modelling flexible slender system for real-time

simulations and control applications

IMT-2007-25	Wroldsen, Anders Sunde	Modelling and control of tensegrity structures. (PhD Thesis, CeSOS)
IMT-2007-26	Aronsen, Kristoffer Høy	An experimental investigation of in-line and combined inline and cross flow vortex induced vibrations. (Dr. avhandling, IMT)
IMT-2007-27	Gao, Zhen	Stochastic Response Analysis of Mooring Systems with Emphasis on Frequency-domain Analysis of Fatigue due to Wide-band Response Processes (PhD Thesis, CeSOS)
IMT-2007-28	Thorstensen, Tom Anders	Lifetime Profit Modelling of Ageing Systems Utilizing Information about Technical Condition. (Dr.ing. thesis, IMT)
IMT-2008-29	Refsnes, Jon Erling Gorset	Nonlinear Model-Based Control of Slender Body AUVs (PhD Thesis, IMT)
IMT-2008-30	Berntsen, Per Ivar B.	Structural Reliability Based Position Mooring. (PhD-Thesis, IMT)
IMT-2008-31	Ye, Naiquan	Fatigue Assessment of Aluminium Welded Box-stiffener Joints in Ships (Dr.ing. thesis, IMT)
IMT-2008-32	Radan, Damir	Integrated Control of Marine Electrical Power Systems. (PhD-Thesis, IMT)
IMT-2008-33	Thomassen, Paul	Methods for Dynamic Response Analysis and Fatigue Life Estimation of Floating Fish Cages. (Dr.ing. thesis, IMT)
IMT-2008-34	Pákozdi, Csaba	A Smoothed Particle Hydrodynamics Study of Two-dimensional Nonlinear Sloshing in Rectangular Tanks. (Dr.ing.thesis, IMT/ CeSOS)
IMT-2007-35	Grytøy, Guttorm	A Higher-Order Boundary Element Method and Applications to Marine Hydrodynamics. (Dr.ing.thesis, IMT)
IMT-2008-36	Drummen, Ingo	Experimental and Numerical Investigation of Nonlinear Wave-Induced Load Effects in Containerships considering Hydroelasticity. (PhD thesis, CeSOS)
IMT-2008-37	Skejic, Renato	Maneuvering and Seakeeping of a Singel Ship and of Two Ships in Interaction. (PhD-Thesis, CeSOS)
IMT-2008-38	Harlem, Alf	An Age-Based Replacement Model for Repairable Systems with Attention to High-Speed Marine Diesel Engines. (PhD-Thesis, IMT)
IMT-2008-39	Alsos, Hagbart S.	Ship Grounding. Analysis of Ductile Fracture, Bottom Damage and Hull Girder Response. (PhD-thesis, IMT)
IMT-2008-40	Graczyk, Mateusz	Experimental Investigation of Sloshing Loading and Load Effects in Membrane LNG Tanks Subjected to Random Excitation. (PhD-thesis, CeSOS)

IMT-2008-41	Taghipour, Reza	Efficient Prediction of Dynamic Response for Flexible and Multi-body Marine Structures. (PhD-thesis, CeSOS)
IMT-2008-42	Ruth, Eivind	Propulsion control and thrust allocation on marine vessels. (PhD thesis, CeSOS)
IMT-2008-43	Nystad, Bent Helge	Technical Condition Indexes and Remaining Useful Life of Aggregated Systems. PhD thesis, IMT
IMT-2008-44	Soni, Prashant Kumar	Hydrodynamic Coefficients for Vortex Induced Vibrations of Flexible Beams, PhD thesis, CeSOS
IMT-2009-45	Amlashi, Hadi K.K.	Ultimate Strength and Reliability-based Design of Ship Hulls with Emphasis on Combined Global and Local Loads. PhD Thesis, IMT
IMT-2009-46	Pedersen, Tom Arne	Bond Graph Modelling of Marine Power Systems. PhD Thesis, IMT
IMT-2009-47	Kristiansen, Trygve	Two-Dimensional Numerical and Experimental Studies of Piston-Mode Resonance. PhD-Thesis, CeSOS
IMT-2009-48	Ong, Muk Chen	Applications of a Standard High Reynolds Number Model and a Stochastic Scour Prediction Model for Marine Structures. PhD-thesis, IMT
IMT-2009-49	Hong, Lin	Simplified Analysis and Design of Ships subjected to Collision and Grounding. PhD-thesis, IMT
IMT-2009-50	Koushan, Kamran	Vortex Induced Vibrations of Free Span Pipelines, PhD thesis, IMT
IMT-2009-51	Korsvik, Jarl Eirik	Heuristic Methods for Ship Routing and Scheduling. PhD-thesis, IMT
IMT-2009-52	Lee, Jihoon	Experimental Investigation and Numerical in Analyzing the Ocean Current Displacement of Longlines. Ph.d.-Thesis, IMT.
IMT-2009-53	Vestbøstad, Tone Gran	A Numerical Study of Wave-in-Deck Impact using a Two-Dimensional Constrained Interpolation Profile Method, Ph.d.thesis, CeSOS.
IMT-2009-54	Bruun, Kristine	Bond Graph Modelling of Fuel Cells for Marine Power Plants. Ph.d.-thesis, IMT
IMT-2009-55	Holstad, Anders	Numerical Investigation of Turbulence in a Skewed Three-Dimensional Channel Flow, Ph.d.-thesis, IMT.
IMT-2009-56	Ayala-Uraga, Efren	Reliability-Based Assessment of Deteriorating Ship-shaped Offshore Structures, Ph.d.-thesis, IMT
IMT-2009-57	Kong, Xiangjun	A Numerical Study of a Damaged Ship in Beam Sea Waves. Ph.d.-thesis, IMT/CeSOS.
IMT-2010-58	Kristiansen, David	Wave Induced Effects on Floaters of Aquaculture Plants, Ph.d.-thesis, CeSOS.

IMT 2010-59	Ludvigsen, Martin	An ROV-Toolbox for Optical and Acoustic Scientific Seabed Investigation. Ph.d.-thesis IMT.
IMT 2010-60	Hals, Jørgen	Modelling and Phase Control of Wave-Energy Converters. Ph.d.thesis, CeSOS.
IMT 2010- 61	Shu, Zhi	Uncertainty Assessment of Wave Loads and Ultimate Strength of Tankers and Bulk Carriers in a Reliability Framework. Ph.d. Thesis, IMT/ CeSOS
IMT 2010-62	Shao, Yanlin	Numerical Potential-Flow Studies on Weakly-Nonlinear Wave-Body Interactions with/without Small Forward Speed, Ph.d.thesis,CeSOS.
IMT 2010-63	Califano, Andrea	Dynamic Loads on Marine Propellers due to Intermittent Ventilation. Ph.d.thesis, IMT.
IMT 2010-64	El Khoury, George	Numerical Simulations of Massively Separated Turbulent Flows, Ph.d.-thesis, IMT
IMT 2010-65	Seim, Knut Sponheim	Mixing Process in Dense Overflows with Emphasis on the Faroec Bank Channel Overflow. Ph.d.thesis, IMT
IMT 2010-66	Jia, Huirong	Structural Analysis of Intect and Damaged Ships in a Collision Risk Analysis Perspective. Ph.d.thesis CeSoS.
IMT 2010-67	Jiao, Linlin	Wave-Induced Effects on a Pontoon-type Very Large Floating Structures (VLFS). Ph.D.-thesis, CeSOS.
IMT 2010-68	Abrahamsen, Bjørn Christian	Sloshing Induced Tank Roof with Entrapped Air Pocket. Ph.d.thesis, CeSOS.
IMT 2011-69	Karimirad, Madjid	Stochastic Dynamic Response Analysis of Spar-Type Wind Turbines with Catenary or Taut Mooring Systems. Ph.d.-thesis, CeSOS.
IMT - 2011-70	Erlend Meland	Condition Monitoring of Safety Critical Valves. Ph.d.-thesis, IMT.
IMT – 2011-71	Yang, Limin	Stochastic Dynamic System Analysis of Wave Energy Converter with Hydraulic Power Take-Off, with Particular Reference to Wear Damage Analysis, Ph.d. Thesis, CeSOS.
IMT – 2011-72	Visscher, Jan	Application of Particla Image Velocimetry on Turbulent Marine Flows, Ph.d.Thesis, IMT.
IMT – 2011-73	Su, Biao	Numerical Predictions of Global and Local Ice Loads on Ships. Ph.d.Thesis, CeSOS.
IMT – 2011-74	Liu, Zhenhui	Analytical and Numerical Analysis of Iceberg Collision with Ship Structures. Ph.d.Thesis, IMT.
IMT – 2011-75	Aarsæther, Karl Gunnar	Modeling and Analysis of Ship Traffic by Observation and Numerical Simulation. Ph.d.Thesis, IMT.

Imt – 2011-76	Wu, Jie	Hydrodynamic Force Identification from Stochastic Vortex Induced Vibration Experiments with Slender Beams. Ph.d.Thesis, IMT.
Imt – 2011-77	Amini, Hamid	Azimuth Propulsors in Off-design Conditions. Ph.d.Thesis, IMT.
IMT – 2011-78	Nguyen, Tan-Hoi	Toward a System of Real-Time Prediction and Monitoring of Bottom Damage Conditions During Ship Grounding. Ph.d.thesis, IMT.
IMT- 2011-79	Tavakoli, Mohammad T.	Assessment of Oil Spill in Ship Collision and Grounding, Ph.d.thesis, IMT.
IMT- 2011-80	Guo, Bingjie	Numerical and Experimental Investigation of Added Resistance in Waves. Ph.d.Thesis, IMT.
IMT- 2011-81	Chen, Qiaofeng	Ultimate Strength of Aluminium Panels, considering HAZ Effects, IMT
IMT- 2012-82	Kota, Ravikiran S.	Wave Loads on Decks of Offshore Structures in Random Seas, CeSOS.
IMT- 2012-83	Sten, Ronny	Dynamic Simulation of Deep Water Drilling Risers with Heave Compensating System, IMT.
IMT- 2012-84	Berle, Øyvind	Risk and resilience in global maritime supply chains, IMT.
IMT- 2012-85	Fang, Shaoji	Fault Tolerant Position Mooring Control Based on Structural Reliability, CeSOS.
IMT- 2012-86	You, Jikun	Numerical studies on wave forces and moored ship motions in intermediate and shallow water, CeSOS.
IMT- 2012-87	Xiang ,Xu	Maneuvering of two interacting ships in waves, CeSOS
IMT- 2012-88	Dong, Wenbin	Time-domain fatigue response and reliability analysis of offshore wind turbines with emphasis on welded tubular joints and gear components, CeSOS
IMT- 2012-89	Zhu, Suji	Investigation of Wave-Induced Nonlinear Load Effects in Open Ships considering Hull Girder Vibrations in Bending and Torsion, CeSOS
IMT- 2012-90	Zhou, Li	Numerical and Experimental Investigation of Station-keeping in Level Ice, CeSOS
IMT- 2012-91	Ushakov, Sergey	Particulate matter emission characteristics from diesel engines operating on conventional and alternative marine fuels, IMT
IMT- 2013-1	Yin, Decao	Experimental and Numerical Analysis of Combined In-line and Cross-flow Vortex Induced Vibrations, CeSOS

IMT-2013-2	Kurniawan, Adi	Modelling and geometry optimisation of wave energy converters, CeSOS
IMT-2013-3	Al Ryati, Nabil	Technical condition indexes doe auxiliary marine diesel engines, IMT
IMT-2013-4	Firoozkoohi, Reza	Experimental, numerical and analytical investigation of the effect of screens on sloshing, CeSOS
IMT-2013-5	Ommani, Babak	Potential-Flow Predictions of a Semi-Displacement Vessel Including Applications to Calm Water Broaching, CeSOS
IMT-2013-6	Xing, Yihan	Modelling and analysis of the gearbox in a floating spar-type wind turbine, CeSOS
IMT-7-2013	Balland, Océane	Optimization models for reducing air emissions from ships, IMT
IMT-8-2013	Yang, Dan	Transitional wake flow behind an inclined flat plate-----Computation and analysis, IMT
IMT-9-2013	Abdillah, Suyuthi	Prediction of Extreme Loads and Fatigue Damage for a Ship Hull due to Ice Action, IMT
IMT-10-2013	Ramirez, Pedro Agustin Pérez	Ageing management and life extension of technical systems- Concepts and methods applied to oil and gas facilities, IMT
IMT-11-2013	Chuang, Zhenju	Experimental and Numerical Investigation of Speed Loss due to Seakeeping and Maneuvering. IMT
IMT-12-2013	Etemaddar, Mahmoud	Load and Response Analysis of Wind Turbines under Atmospheric Icing and Controller System Faults with Emphasis on Spar Type Floating Wind Turbines, IMT
IMT-13-2013	Lindstad, Haakon	Strategies and measures for reducing maritime CO2 emissons, IMT
IMT-14-2013	Haris, Sabril	Damage interaction analysis of ship collisions, IMT
IMT-15-2013	Shainee, Mohamed	Conceptual Design, Numerical and Experimental Investigation of a SPM Cage Concept for Offshore Mariculture, IMT
IMT-16-2013	Gansel, Lars	Flow past porous cylinders and effects of biofouling and fish behavior on the flow in and around Atlantic salmon net cages, IMT
IMT-17-2013	Gaspar, Henrique	Handling Aspects of Complexity in Conceptual Ship Design, IMT
IMT-18-2013	Thys, Maxime	Theoretical and Experimental Investigation of a Free Running Fishing Vessel at Small Frequency of Encounter, CeSOS
IMT-19-2013	Aglen, Ida	VIV in Free Spanning Pipelines, CeSOS

IMT-1-2014	Song, An	Theoretical and experimental studies of wave diffraction and radiation loads on a horizontally submerged perforated plate, CeSOS
IMT-2-2014	Rogne, Øyvind Ygre	Numerical and Experimental Investigation of a Hinged 5-body Wave Energy Converter, CeSOS
IMT-3-2014	Dai, Lijuan	Safe and efficient operation and maintenance of offshore wind farms ,IMT
IMT-4-2014	Bachynski, Erin Elizabeth	Design and Dynamic Analysis of Tension Leg Platform Wind Turbines, CeSOS
IMT-5-2014	Wang, Jingbo	Water Entry of Freefall Wedged – Wedge motions and Cavity Dynamics, CeSOS
IMT-6-2014	Kim, Ekaterina	Experimental and numerical studies related to the coupled behavior of ice mass and steel structures during accidental collisions, IMT
IMT-7-2014	Tan, Xiang	Numerical investigation of ship's continuous- mode icebreaking in level ice, CeSOS
IMT-8-2014	Muliawan, Made Jaya	Design and Analysis of Combined Floating Wave and Wind Power Facilities, with Emphasis on Extreme Load Effects of the Mooring System, CeSOS
IMT-9-2014	Jiang, Zhiyu	Long-term response analysis of wind turbines with an emphasis on fault and shutdown conditions, IMT
IMT-10-2014	Dukan, Fredrik	ROV Motion Control Systems, IMT
IMT-11-2014	Grimsmo, Nils I.	Dynamic simulations of hydraulic cylinder for heave compensation of deep water drilling risers, IMT
IMT-12-2014	Kvittem, Marit I.	Modelling and response analysis for fatigue design of a semisubmersible wind turbine, CeSOS
IMT-13-2014	Akhtar, Juned	The Effects of Human Fatigue on Risk at Sea, IMT
IMT-14-2014	Syahroni, Nur	Fatigue Assessment of Welded Joints Taking into Account Effects of Residual Stress, IMT
IMT-1-2015	Böckmann, Eirik	Wave Propulsion of ships, IMT
IMT-2-2015	Wang, Kai	Modelling and dynamic analysis of a semi-submersible floating vertical axis wind turbine, CeSOS
IMT-3-2015	Fredriksen, Arnt Gunvald	A numerical and experimental study of a two-dimensional body with moonpool in waves and current, CeSOS
IMT-4-2015	Jose Patricio Gallardo Canabes	Numerical studies of viscous flow around bluff bodies, IMT

IMT-5-2015	Vegard Longva	Formulation and application of finite element techniques for slender marine structures subjected to contact interactions, IMT
IMT-6-2015	Jacobus De Vaal	Aerodynamic modelling of floating wind turbines, CeSOS
IMT-7-2015	Fachri Nasution	Fatigue Performance of Copper Power Conductors, IMT
IMT-8-2015	Oleh I Karpa	Development of bivariate extreme value distributions for applications in marine technology, CeSOS
IMT-9-2015	Daniel de Almeida Fernandes	An output feedback motion control system for ROVs, AMOS
IMT-10-2015	Bo Zhao	Particle Filter for Fault Diagnosis: Application to Dynamic Positioning Vessel and Underwater Robotics, CeSOS
IMT-11-2015	Wenting Zhu	Impact of emission allocation in maritime transportation, IMT
IMT-12-2015	Amir Rasekhi Nejad	Dynamic Analysis and Design of Gearboxes in Offshore Wind Turbines in a Structural Reliability Perspective, CeSOS
IMT-13-2015	Arturo Jesús Ortega Malca	Dynamic Response of Flexibles Risers due to Unsteady Slug Flow, CeSOS
IMT-14-2015	Dagfinn Husjord	Guidance and decision-support system for safe navigation of ships operating in close proximity, IMT
IMT-15-2015	Anirban Bhattacharyya	Ducted Propellers: Behaviour in Waves and Scale Effects, IMT
IMT-16-2015	Qin Zhang	Image Processing for Ice Parameter Identification in Ice Management, IMT
IMT-1-2016	Vincentius Rumawas	Human Factors in Ship Design and Operation: An Experiential Learning, IMT
IMT-2-2016	Martin Storheim	Structural response in ship-platform and ship-ice collisions, IMT
IMT-3-2016	Mia Abrahamsen Prsic	Numerical Simulations of the Flow around single and Tandem Circular Cylinders Close to a Plane Wall, IMT
IMT-4-2016	Tufan Arslan	Large-eddy simulations of cross-flow around ship sections, IMT

IMT-5-2016	Pierre Yves-Henry	Parametrisation of aquatic vegetation in hydraulic and coastal research,IMT
IMT-6-2016	Lin Li	Dynamic Analysis of the Instalation of Monopiles for Offshore Wind Turbines, CeSOS
IMT-7-2016	Øivind Kåre Kjerstad	Dynamic Positioning of Marine Vessels in Ice, IMT
IMT-8-2016	Xiaopeng Wu	Numerical Analysis of Anchor Handling and Fish Trawling Operations in a Safety Perspective, CeSOS
IMT-9-2016	Zhengshun Cheng	Integrated Dynamic Analysis of Floating Vertical Axis Wind Turbines, CeSOS
IMT-10-2016	Ling Wan	Experimental and Numerical Study of a Combined Offshore Wind and Wave Energy Converter Concept
IMT-11-2016	Wei Chai	Stochastic dynamic analysis and reliability evaluation of the roll motion for ships in random seas, CeSOS
IMT-12-2016	Øyvind Selnes Patricksson	Decision support for conceptual ship design with focus on a changing life cycle and future uncertainty, IMT
IMT-13-2016	Mats Jørgen Thorsen	Time domain analysis of vortex-induced vibrations, IMT
IMT-14-2016	Edgar McGuinness	Safety in the Norwegian Fishing Fleet – Analysis and measures for improvement, IMT
IMT-15-2016	Sepideh Jafarzadeh	Energy efficiency and emission abatement in the fishing fleet, IMT
IMT-16-2016	Wilson Ivan Guachamin Acero	Assessment of marine operations for offshore wind turbine installation with emphasis on response-based operational limits, IMT
IMT-17-2016	Mauro Candeloro	Tools and Methods for Autonomous Operations on Seabed and Water Coumn using Underwater Vehicles, IMT
IMT-18-2016	Valentin Chabaud	Real-Time Hybrid Model Testing of Floating Wind Tubines, IMT
IMT-1-2017	Mohammad Saud Afzal	Three-dimensional streaming in a sea bed boundary layer
IMT-2-2017	Peng Li	A Theoretical and Experimental Study of Wave-induced Hydroelastic Response of a Circular Floating Collar
IMT-3-2017	Martin Bergström	A simulation-based design method for arctic maritime transport systems

IMT-4-2017	Bhushan Taskar	The effect of waves on marine propellers and propulsion
IMT-5-2017	Mohsen Bardestani	A two-dimensional numerical and experimental study of a floater with net and sinker tube in waves and current
IMT-6-2017	Fatemeh Hoseini Dadmarzi	Direct Numerical Simulation of turbulent wakes behind different plate configurations
IMT-7-2017	Michel R. Miyazaki	Modeling and control of hybrid marine power plants
IMT-8-2017	Giri Rajasekhar Gunnu	Safety and efficiency enhancement of anchor handling operations with particular emphasis on the stability of anchor handling vessels
IMT-9-2017	Kevin Koosup Yum	Transient Performance and Emissions of a Turbocharged Diesel Engine for Marine Power Plants
IMT-10-2017	Zhaolong Yu	Hydrodynamic and structural aspects of ship collisions
IMT-11-2017	Martin Hassel	Risk Analysis and Modelling of Allisions between Passing Vessels and Offshore Installations
IMT-12-2017	Astrid H. Brodtkorb	Hybrid Control of Marine Vessels – Dynamic Positioning in Varying Conditions
IMT-13-2017	Kjersti Bruserud	Simultaneous stochastic model of waves and current for prediction of structural design loads
IMT-14-2017	Finn-Idar Grøtta Giske	Long-Term Extreme Response Analysis of Marine Structures Using Inverse Reliability Methods
IMT-15-2017	Stian Skjong	Modeling and Simulation of Maritime Systems and Operations for Virtual Prototyping using co-Simulations
IMT-1-2018	Yingguang Chu	Virtual Prototyping for Marine Crane Design and Operations
IMT-2-2018	Sergey Gavrilin	Validation of ship manoeuvring simulation models
IMT-3-2018	Jeevith Hegde	Tools and methods to manage risk in autonomous subsea inspection, maintenance and repair operations
IMT-4-2018	Ida M. Strand	Sea Loads on Closed Flexible Fish Cages
IMT-5-2018	Erlend Kvinge Jørgensen	Navigation and Control of Underwater Robotic Vehicles

IMT-6-2018	Bård Stovner	Aided Inertial Navigation of Underwater Vehicles
IMT-7-2018	Erlend Liavåg Grotle	Thermodynamic Response Enhanced by Sloshing in Marine LNG Fuel Tanks
IMT-8-2018	Børge Rokseth	Safety and Verification of Advanced Maritime Vessels
IMT-9-2018	Jan Vidar Ulveseter	Advances in Semi-Empirical Time Domain Modelling of Vortex-Induced Vibrations
IMT-10-2018	Chenyu Luan	Design and analysis for a steel braceless semi-submersible hull for supporting a 5-MW horizontal axis wind turbine
IMT-11-2018	Carl Fredrik Rehn	Ship Design under Uncertainty
IMT-12-2018	Øyvind Ødegård	Towards Autonomous Operations and Systems in Marine Archaeology
IMT-13-2018	Stein Melvær Nornes	Guidance and Control of Marine Robotics for Ocean Mapping and Monitoring
IMT-14-2018	Petter Norgren	Autonomous Underwater Vehicles in Arctic Marine Operations: Arctic marine research and ice monitoring
IMT-15-2018	Minjoo Choi	Modular Adaptable Ship Design for Handling Uncertainty in the Future Operating Context
MT-16-2018	Ole Alexander Eidsvik	Dynamics of Remotely Operated Underwater Vehicle Systems
IMT-17-2018	Mahdi Ghane	Fault Diagnosis of Floating Wind Turbine Drivetrain- Methodologies and Applications
IMT-18-2018	Christoph Alexander Thieme	Risk Analysis and Modelling of Autonomous Marine Systems
IMT-19-2018	Yugao Shen	Operational limits for floating-collar fish farms in waves and current, without and with well-boat presence
IMT-20-2018	Tianjiao Dai	Investigations of Shear Interaction and Stresses in Flexible Pipes and Umbilicals
IMT-21-2018	Sigurd Solheim Pettersen	Resilience by Latent Capabilities in Marine Systems
IMT-22-2018	Thomas Sauder	Fidelity of Cyber-physical Empirical Methods. Application to the Active Truncation of Slender Marine Structures
IMT-23-2018	Jan-Tore Horn	Statistical and Modelling Uncertainties in the Design of Offshore Wind Turbines

IMT-24-2018	Anna Swider	Data Mining Methods for the Analysis of Power Systems of Vessels
IMT-1-2019	Zhao He	Hydrodynamic study of a moored fish farming cage with fish influence
IMT-2-2019	Isar Ghamari	Numerical and Experimental Study on the Ship Parametric Roll Resonance and the Effect of Anti-Roll Tank
IMT-3-2019	Håkon Strandenes	Turbulent Flow Simulations at Higher Reynolds Numbers
IMT-4-2019	Siri Mariane Holen	Safety in Norwegian Fish Farming – Concepts and Methods for Improvement
IMT-5-2019	Ping Fu	Reliability Analysis of Wake-Induced Riser Collision
IMT-6-2019	Vladimir Krivopolianskii	Experimental Investigation of Injection and Combustion Processes in Marine Gas Engines using Constant Volume Rig
IMT-7-2019	Anna Maria Kozłowska	Hydrodynamic Loads on Marine Propellers Subject to Ventilation and out of Water Condition.
IMT-8-2019	Hans-Martin Heyn	Motion Sensing on Vessels Operating in Sea Ice: A Local Ice Monitoring System for Transit and Stationkeeping Operations under the Influence of Sea Ice
IMT-9-2019	Stefan Vilsen	Method for Real-Time Hybrid Model Testing of Ocean Structures – Case on Slender Marine Systems
IMT-10-2019	Finn-Christian W. Hanssen	Non-Linear Wave-Body Interaction in Severe Waves
IMT-11-2019	Trygve Olav Fossum	Adaptive Sampling for Marine Robotics
IMT-12-2019	Jørgen Bremnes Nielsen	Modeling and Simulation for Design Evaluation
IMT-13-2019	Yuna Zhao	Numerical modelling and dynamic analysis of offshore wind turbine blade installation
IMT-14-2019	Daniela Myland	Experimental and Theoretical Investigations on the Ship Resistance in Level Ice
IMT-15-2019	Zhengru Ren	Advanced control algorithms to support automated offshore wind turbine installation
IMT-16-2019	Drazen Polic	Ice-propeller impact analysis using an inverse propulsion machinery simulation approach
IMT-17-2019	Endre Sandvik	Sea passage scenario simulation for ship system performance evaluation

IMT-18-2019	Loup Suja-Thauvin	Response of Monopile Wind Turbines to Higher Order Wave Loads
IMT-19-2019	Emil Smilden	Structural control of offshore wind turbines – Increasing the role of control design in offshore wind farm development
IMT-20-2019	Aleksandar-Sasa Milakovic	On equivalent ice thickness and machine learning in ship ice transit simulations
IMT-1-2020	Amrit Shankar Verma	Modelling, Analysis and Response-based Operability Assessment of Offshore Wind Turbine Blade Installation with Emphasis on Impact Damages
IMT-2-2020	Bent Oddvar Arnesen Haugaløkken	Autonomous Technology for Inspection, Maintenance and Repair Operations in the Norwegian Aquaculture
IMT-3-2020	Seongpil Cho	Model-based fault detection and diagnosis of a blade pitch system in floating wind turbines
IMT-4-2020	Jose Jorge Garcia Agis	Effectiveness in Decision-Making in Ship Design under Uncertainty
IMT-5-2020	Thomas H. Viuff	Uncertainty Assessment of Wave-and Current-induced Global Response of Floating Bridges
IMT-6-2020	Fredrik Mentzoni	Hydrodynamic Loads on Complex Structures in the Wave Zone
IMT-7-2020	Senthuran Ravinthrakumar	Numerical and Experimental Studies of Resonant Flow in Moonpools in Operational Conditions
IMT-8-2020	Stian Skaalvik Sandøy	Acoustic-based Probabilistic Localization and Mapping using Unmanned Underwater Vehicles for Aquaculture Operations
IMT-9-2020	Kun Xu	Design and Analysis of Mooring System for Semi-submersible Floating Wind Turbine in Shallow Water
IMT-10-2020	Jianxun Zhu	Cavity Flows and Wake Behind an Elliptic Cylinder Translating Above the Wall
IMT-11-2020	Sandra Hogenboom	Decision-making within Dynamic Positioning Operations in the Offshore Industry – A Human Factors based Approach
IMT-12-2020	Woongshik Nam	Structural Resistance of Ship and Offshore Structures Exposed to the Risk of Brittle Failure
IMT-13-2020	Svenn Are Tutturen Værnø	Transient Performance in Dynamic Positioning of Ships: Investigation of Residual Load Models and Control Methods for Effective Compensation
IMT-14-2020	Mohd Atif Siddiqui	Experimental and Numerical Hydrodynamic Analysis of a Damaged Ship in Waves
IMT-15-2020	John Marius Hegseth	Efficient Modelling and Design Optimization of Large Floating Wind Turbines

IMT-16-2020	Asle Natskär	Reliability-based Assessment of Marine Operations with Emphasis on Sea Transport on Barges
IMT-17-2020	Shi Deng	Experimental and Numerical Study of Hydrodynamic Responses of a Twin-Tube Submerged Floating Tunnel Considering Vortex-Induced Vibration
IMT-18-2020	Jone Torsvik	Dynamic Analysis in Design and Operation of Large Floating Offshore Wind Turbine Drivetrains
IMT-1-2021	Ali Ebrahimi	Handling Complexity to Improve Ship Design Competitiveness
IMT-2-2021	Davide Proserpio	Isogeometric Phase-Field Methods for Modeling Fracture in Shell Structures
IMT-3-2021	Cai Tian	Numerical Studies of Viscous Flow Around Step Cylinders
IMT-4-2021	Farid Khazaeli Moghadam	Vibration-based Condition Monitoring of Large Offshore Wind Turbines in a Digital Twin Perspective
IMT-5-2021	Shuaishuai Wang	Design and Dynamic Analysis of a 10-MW Medium-Speed Drivetrain in Offshore Wind Turbines
IMT-6-2021	Sadi Tavakoli	Ship Propulsion Dynamics and Emissions
IMT-7-2021	Haoran Li	Nonlinear wave loads, and resulting global response statistics of a semi-submersible wind turbine platform with heave plates
IMT-8-2021	Einar Skiftestad Ueland	Load Control for Real-Time Hybrid Model Testing using Cable-Driven Parallel Robots
IMT-9-2021	Mengning Wu	Uncertainty of machine learning-based methods for wave forecast and its effect on installation of offshore wind turbines
IMT-10-2021	Xu Han	Onboard Tuning and Uncertainty Estimation of Vessel Seakeeping Model Parameters
IMT-01-2022	Ingunn Marie Holmen	Safety in Exposed Aquaculture Operations
IMT-02-2022	Prateek Gupta	Ship Performance Monitoring using In-service Measurements and Big Data Analysis Methods
IMT-03-2022	Sangwoo Kim	Non-linear time domain analysis of deepwater riser vortex-induced vibrations
IMT-04-2022	Jarle Vinje Kramer	Hydrodynamic Aspects of Sail-Assisted Merchant Vessels
IMT-05-2022	Øyvind Rabliås	Numerical and Experimental Studies of Maneuvering in Regular and Irregular Waves

IMT-06-2022	Pramod Ghimire	Simulation-Based Ship Hybrid Power System Conspect Studies and Performance Analyses
IMT-07-2022	Carlos Eduardo Silva de Souza	Structural modelling, coupled dynamics, and design of large floating wind turbines
IMT-08-2022	Lorenzo Balestra	Design of hybrid fuel cell & battery systems for maritime vessels
IMT-09-2022	Sharmin Sultana	Process safety and risk management using system perspectives – A contribution to the chemical process and petroleum industry
IMT-10-2022	Øystein Sture	Autonomous Exploration for Marine Minerals
IMT-11-2022	Tiantian Zhu	Information and Decision-making for Major Accident Prevention – A concept of information- based strategies for accident prevention
IMT-12-2022	Siamak Karimi	Shore-to-Ship Charging Systems for Battery- Electric Ships
IMT-01-2023	Huili Xu	Fish-inspired Propulsion Study: Numerical Hydrodynamics of Rigid/Flexible/Morphing Foils and Observations on Real Fish
IMT-02-2023	Chana Sinsabvarodom	Probabilistic Modelling of Ice-drift and Ice Loading on Fixed and Floating Offshore Structures
IMT-03-2023	Martin Skaldebo	Intelligent low-cost solutions for underwater intervention using computer vision and machine learning

