Bridging XLUUV and MASS – Technologies for Autonomous Multi-Domain Operations

Carsten Rethfeldt[1], **Alexander Klein**[2], **Malte Riesner**[3],
Martin Kurowski[1], **Jannis Stoppe**[2], **Svenja Schubert**[3], **and**
Torsten Jeinsch[1]

[1]Institute of Automation, University of Rostock, Germany
[2]German Aerospace Center, Germany
[3]TKMS GmbH, Germany

E-mail: carsten.rethfeldt@uni-rostock.de

Abstract.

The development of extra-large unmanned underwater vehicles (XLUUV) has made significant progress. In that context, the paper describes firstly the evolution of the innovative technological vehicle basis MUM (Modifiable Underwater Mothership). A MUM full scale demonstrator of 25 meters length has been developed and is currently in production. MUM has been designed to be operated autonomously. This includes both the execution of mission-specific tasks as well as transit (surfaced and submerged) and port operations. In order to achieve a successful system realization, situation awareness capabilities and safe autonomous navigation become mandatory, requiring specific technologies that bridge XLUUV and MASS. This connection to MASS is shown by explaining the developments in the areas of maneuverability and automated vehicle guidance as well as reliable environmental perception, which are essential for the realization of vehicle autonomy, especially in surface operation. On the one hand, the article describes the methods for designing the MUM demonstrator's actuator configuration and maneuvering system ensuring the necessary maneuvering capabilities and, on the other hand, outlines the selection process for optical sensor systems and AI-based evaluation methods in a dynamic maritime environment enabling reliable obstacle avoidance on the XLUUV. The study concludes with simulations of the performance of the maneuvering system of the MUM demonstrator configuration and the results of the implemented visual perception system using recorded sensor data in a port scenario.

1 Introduction

1.1 Motivation

The market for unmanned underwater vehicles (UUV) is growing. New developments reach higher levels of vehicle autonomy and extend their mission capabilities consistently. Such autonomous underwater vehicles can fulfill a wide range of mission types, depending mostly on the main vehicle particulars and system configuration. In common usage, it has become established to distinguish between AUVs (autonomous underwater vehicles), LUUVs (large uncrewed underwater vehicles) and XLUUVs (extra large uncrewed underwater vehicles). In principle, each of these three vehicle types can operate in an autonomous manner. Thus, the vehicle type does not classify the level of automation, but rather the vehicle size. No clear definition exists. However, the term AUV usually refers to small autonomous underwater vehicles

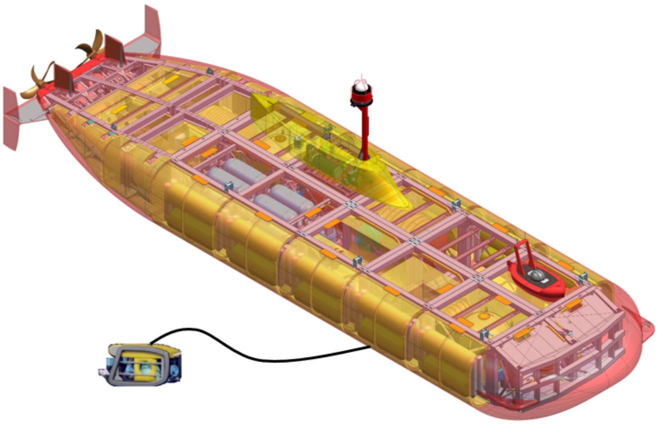


Figure 1: Isometric view of the MUM demonstrator applied for ROV operations

of a length less than approximately 5 meters. Vehicles larger than AUVs and smaller than approximately 15 meters are usually called LUUV. These vehicles can often be launched and recovered by a service vessel. Provided with the necessary energy capacity, they can start their mission from harbor. However, this requires their safe participation in maritime transport. Often, LUUVs do not have the required performance in surfaced mode to operate and maneuver in waves and wind and are not safely visible for other participants in maritime traffic. They are usually not equipped with a navigation mast that brings navigation and communication systems into a reasonable height above the sea surface. For this reason, most LUUVs have to be transported to the operational sea area by a service vessel. The main dimensions of such LUUVs are often constituted to fit into standard container sizes, thus allowing easy transport by truck or ship. Several designs include a small payload bay to integrate medium size payloads, e.g. small remotely operated vehicles (ROVs).

Vehicles of the XLUUV type are not limited in size. Current developments reach lengths of more than 25 meters and a width of more than 6 meters [1]. While such dimensions allow submarine-like applications, which is why the most unmanned vehicles are attracting increased interest for military applications, XLUUV are well suited for a wide range of civil applications, e.g. protection of critical maritime infrastructure or maintenance work on subsea structures and systems without the need for a surface vessel. Within the civilian context, the concept of a subsea shuttle tanker was presented as a potential alternative to subsea pipelines, umbilicals, and tanker ships [2]. The proposed design has a length of 100–200 m and a diameter of 10–20 m. However, the size of the vehicles also leads to new technical and legal challenges in the operation of these unmanned vehicles. Recent innovations aim to develop new approaches towards modularity. A leading example of this is the Modifiable Underwater Mothership (MUM), where the overall aim is to develop a novel class of underwater vehicles to cope with a wide range of operations [3]. The modular vehicle design breaks with the traditional concept of a fixed hull structure. Its shape is completely variable in consequence of the mission dependent assembly and arrangement of basic and mission specific modules, as illustrated exemplarily in Fig. 1.

A comparison between XLUUVs and the smaller AUVs and LUUVs shows significant differences. Usually, AUVs are deployed for scientific or commercial purposes and can be equipped with specialized payloads for a specific monitoring mission. They are designed for short to medium missions at depths of up to 6,000 m and are able to operate autonomously for limited periods of time and space. The authors of [4] provide a comprehensive overview of different vehicle platforms with a focus on under-ice applications. What all these vehicles have in common is the fact that a ship is required for deployment, allowing the AUV to be launched and recovered using the ship's crane or A-frame or a specialized launch and recovery system (LARS). This can be quite challenging depending on the weather conditions [5]. In contrast to that, XLUUV are designed for longer, complex missions in which more or larger sensor arrays or simply heavy payloads can be transported. In the case of payloads that are transported for use in the operational area, interaction with the environment may be necessary, such as placing a station on the seabed or using a drilling rig to extract raw materials [3]. Another key difference is the way the vehicles are deployed and recovered, as XLUUVs are not deployed from a ship in the area of operation as usual, requiring a transition to port-to-port missions instead.

Port-to-port missions generally impose strict requirements on the vehicle technologies used for autonomous vehicle guidance and the potential remote access by a supervisor. Basically, a port-to-port mission can be divided into several phases. These include the ship-like surface passage in public fairways or in restricted and safety-critical areas such as ports, the usually submerged transfer to the area of operation, and the execution of the mission itself while submerged or surfaced. The different mission phases are subject to specific requirements, which can be roughly grouped into *localization and navigation*, *vehicle guidance and control* and *operational safety*, of which the first two challenges will be addressed in this contribution. In terms of vehicle operational safety, fault tolerance has to be considered with regard to sensor and actuator faults, as these can restrict both navigation and vehicle control [1].

1.2 Related Work

Precise localization and navigation are key elements for autonomous vehicles in ports, as they ensure that the vehicles follow their path and thus are able to avoid collisions. In ports in particular, buildings and environmental and weather conditions such as wind, currents, tides and sedimentation make navigation difficult since individual sensors may be limited. The navigation solution will have to be robust against increased sensor data variance and sensor failures. In addition, advanced sensor setups such as lidar, radar and cameras have to be used to ensure reliable environmental perception and localization of structures on the surface. Current development presented in literature mostly deals with environment perception for uncrewed surface vessels (USVs) from an AIs point of view given certain factors based on angle or weather conditions [6]. These solutions often lack the connection within a holistic system, as they only solve partial problems of detecting objects using a single camera. The requirements for safe navigation also include hardware requirements, creating a complex system that further imposes challenges on the software pipeline extending the proposed methods. Besides the more technical aspects, the solution must be integrated into an existing legal framework that has so far been more focused on human navigation. The standard rule set is defined by the International Regulations for Preventing Collisions at Sea (COLREGs) [7]. Especially Rule 5 requires a lookout on each vessel to asses a situation and minimize collision risks. Various organizations are in the process of modernizing these rules especially for the MASS topic [8], but they define the base capabilities of a system's sensor functions and decision process, which rely on a single or a crew of experienced seafarers.

Automated vehicle guidance and control in public fairways or in restricted and safety-critical areas such as ports are particularly important and comparable to aspects of the development of functionalities of MASS. This concerns autonomy and decision-making in dynamic environments. Vehicles must be able to react to unexpected obstacles, avoiding collisions with fixed structures and moving objects. Numerous studies on path and trajectory planning can be found in the literature in order to resolve encounter situations collision-free in principle and to adapt the vehicle paths in accordance with the rules [9]. In addition, interaction with the port infrastructure is necessary, e.g. by detecting and engaging target points. This includes automated docking [10] or the precise approach of platforms, buoys or underwater stations during submerged missions. Essential for these operations is equipping the MASS or the XLUUV with high-performance and potentially redundant propulsion and steering devices to ensure a necessary maneuvering capability. The evaluation of the capabilities of a maritime vehicle to perform a specific operation is derived from Dynamic Positioning (DP) of ships, called DP capability [11]. Following on from this, the authors of [12] present a framework that can be used to determine whether an ROV motion system has sufficient propulsion capacity to withstand environmental and operational loads while maintaining a given position or motion. For the modular MUM XLUUV, the methodology can be used to find a suitable configuration for a defined operation scenario. In that context, the work [13] presents results of maneuvering capability calculations comparing two MUM variants concerning their actuator configuration. Once a suitable configuration has been found, the automated maneuvering system can be applied. Here, the actuator allocation, i.e. the distribution of force and torque specifications to the available actuators [14], forms the basis for all higher automation functions. In contrast to ships, specific requirements of multi-domain operation have to be integrated into the allocation for XLUUV.

1.3 Contribution

In the context of MASS, this contribution introduces the innovative vehicle platform MUM and illustrates the modular approach as well as exemplary application scenarios in Section 2.

Section 3 addresses the methods for designing the basis of an automated maneuvering system focusing on actuator allocation. Here, the previous work on actuator allocation is extended by formulating the problem in the context of multi-domain operations navigating in surfaced and submerged mode. The comprehensive application of propulsion and steering devices, which are specifically designed for surfaced or submerged operation, poses challenges in terms of methodological implementation of the actuator allocation.

In Section 4, the contribution will be extended by the work on environment perception specifically for surface operation, introducing a novel optical sensor system. In contrast to today's standard sensor systems, the work shows that long ranges in object detection can be achieved by applying high-resolution cameras and demonstrates the feasibility of monocular distance estimation.

Both approaches are validated in simulation and real-world experiments in Section 5. The contribution closes with a conclusion and an outlook.

2 Modifiable Vehicle Basis

2.1 Application Scenarios for MUM

As aforementioned, MUM is an XLUUV designed for civil operations. Its main areas of application are dedicated to offshore and scientific scenarios. Operations at sea are highly affected by meteorological conditions. Fair weather conditions usually only occur on a few days a year. Severe wind and waves encumber work at sea and lead to enormous safety requirements. Although modern offshore supply vessels (OSV) or research ships are optimized for such operating conditions, wind and waves remain a major challenge for operations at sea. Numerous offshore and research operation are performed below the water surface. Whether maintenance work at pipelines via a remotely operated vehicle (ROV) or seismic measurements at the sea bottom, the required systems and measuring instruments are usually still launched and recovered by surface ships. Even if the actual control of the subsea ROV and sensors might be carried out by experts at an onshore station, where data transfer is realized via satellite communication, the operation of the surface vessel still requires a certain amount of crew. The research project MUM2 has the goal to develop an unmanned and autonomous vehicle that will fully replace manned OSV or research ships for many mission types. This approach considerably reduces the influence of weather conditions and thus maximizes the number of operation days per year.

One of the most complex, but also innovative applications of MUM is the multi-vehicle operation, in which a payload ROV carries out underwater work in a joint mission together with the MUM carrier vehicle, as sketched in Fig. 1. This is also intended to enable a transition from a human-centered approach to a human-supervised scenario. MUM may operate submerged or on the water surface, requiring a high degree of vehicle autonomy of MUM and ROV, especially for coordinated vehicle movement and automated docking and undocking processes. A typical mission scenario could be described as follows:

1. MUM starts from its base, e.g. in a port, and navigates through the confined areas in surfaced operation.
2. MUM travels autonomously to the operational sea area preferably while submerged.
3. Once the operation area has been reached, MUM releases its communication buoy to set up a satellite communication to a command and control center.
4. Via satellite communication, the command and control center is able to supervise the operation and offer the possibility to change the mission parameters of MUM and the ROV.
5. After the operation, MUM retracts the communication buoy and travels autonomously back to its base.

To minimize the risk of collisions, the autonomous transit should be carried out in submerged mode. However, depending on the sea area, this is not always possible. Especially when moving in navigation channels near ports with intensive maritime traffic, the ability to conduct transits at surface based on autonomous collision avoidance is essential and bridges the gap between the required XLUUV and MASS technology.

2.2 Modularity and Evolution of MUM

The exemplary mission discussed previously is only one of many possible use cases. The payload outfit of MUM is highly adaptable according to the requirements of multi-domain missions. The vehicle is fully modular and based on ISO container dimensions of 10-ft and 20-ft containers. Therefore, the entire vehicle can be dismounted and transported to a new base via truck, train or container ship. Modules are differentiated into basic modules and mission modules. Basic modules comprise systems that are mandatory for the vehicle operation, such as energy supply systems, energy storage or propulsion and maneuvering systems. Mission modules can be almost freely chosen by the operator, such as an ROV module, an AUV launch and recovery module, a communication buoy module, a drilling module and many more [3, 15]. Depending on the number and size of the mission modules carried, MUM can vary

Table 1: Main particulars of the MUM demonstrator

Properties	**Value**
Classification	XLUUV
Dimensions (Demonstrator)	26 m length, 7 m beam, 3.5 m height
Energy system	Electric, Fuel Cell supported by battery
Communication	RF Com (LOS), (tethered) SATCOM buoy, UW Com (EvoLogics acoustic modem)
Maximum speed	4 knots in 2 kn current
Maneuvering capabilities	Dynamic positioning (in current up to 2 kn)
Payload capacity	Flexible / Mission adaptive

in length and displacement. Within the MUM2 project, the project partners develop and manufacture a full scale demonstrator with the main particulars shown in Table 1

The MUM demonstrator is equipped with three payload modules, namely the communication buoy module, the swim-out-modem module (launch and recovery module for AUVs) and the ROV module. Complex underwater operations may require a remote operator, e.g. underwater maintenance work with an ROV. The communication buoy is able to generate a control and communication gateway via satellite connection to allow data transfer and remote operation of the total vehicle or single payload systems.

Based on the variety of payload requirements, MUMs length may change up to a maximum length of 50 meters. A change in length significantly affects the propulsion and maneuvering behavior of the vehicle meaning that a new configuration has to be evaluated using suitable methods, for example as presented in [13]. However, according to the changing inertial and hydrodynamic properties, a changing vehicle shape also affects the vehicle guidance and automated maneuvering control system.

3 Automated Maneuvering System for Multi-domain Operations

3.1 Modularity on the Operational Level

The highly automated operation of MUM and the diverse range of tasks to be performed by the vehicle implicate varying sets of requirements and performance levels for the control system. As a consequence of the modular structure and shape due to the mission-dependent assembly and arrangement of modules, the dynamic behavior varies significantly between different configurations, requiring the control system to adapt accordingly. Considering these highly variant vehicle characteristics, standardized procedures for design and parametrization of the automated motion control system are non-existent, as pointed out in [16]. Instead, the authors in [16] present an approach towards a generalized design framework. By designing their methods to be applicable to generic configurations of a MUM vehicle, they create an iterative control system design loop for generating and parameterizing the motion model and the control system, thus making the process automatable. In this contribution, we adopt the generalized modeling and control concept. One key aspect of the design loop is the definition of different control modes for different operating ranges of the vehicle in accordance with the envisioned set of tasks to be performed. Apart from surfaced and submerged transit, a MUM mission plan might additionally require the vehicle to be capable of dynamically positioning, which is not only relevant in the mission area, but especially for automated docking procedures and slow-speed maneuvering in the light of port-to-port missions.

Basically, the MUM demonstrator operates in six degrees of freedom (DoF) when submerged, resulting in the state vector $\mathbf{x} = [\boldsymbol{\eta}^T \ \boldsymbol{\nu}^T]^T$, where $\boldsymbol{\eta} = [x \ y \ z \ \phi \ \theta \ \psi]^T$ corresponds to the position and attitude in the Earth-fixed frame. The vector $\boldsymbol{\nu} = [u \ v \ w \ p \ q \ r]^T$ describes the vehicle's translational and rotational velocities in body-fixed coordinates and the vector $\boldsymbol{\tau} = [X \ Y \ Z \ K \ M \ N]^T$ contains actuator-generated forces and moments acting in total in the respective DoF. The distinction between surfaced and submerged motion means that degrees of freedom are limited in both transit and DP mode. In transit mode at high(er) speed, surfaced or submerged, the focus is on energy efficiency. For operation at the water surface, the state vector is reduced to three DoF (u, v, r) and (x, y, ψ), respectively. These vehicle state definitions also apply to DP at speed about zero and for docking procedures with slow speed maneuvering, as these operations can be carried out either surfaced or submerged, as well. The focus here is on control quality, meaning that fast alteration of actuator forces and directions have to be expected. These qualitative requirements have to be satisfied by the maneuvering system including the actuator allocation and the available propulsion, steering and ballasting systems.

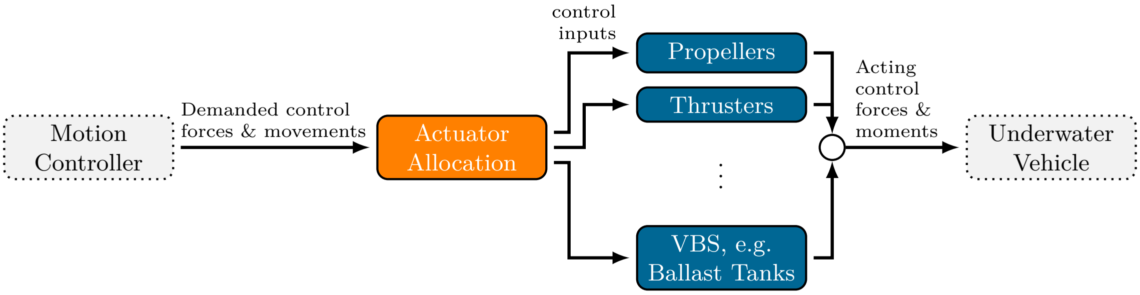


Figure 2: Maneuvering system structure focussing on actuator allocation interfacing the higher-level motion control and the vehicle's drives.

3.2 Maneuvering System Structure

Fundamental for the control system generalization is the separation of higher-level motion control and lower-level actuator allocation, where the former generates vehicle-fixed abstract forces and moments that are to act on the vehicle in order to meet the control objective and the latter distributes these demands among the available propulsion and maneuvering actuators. As clearly conveyed in the illustration of the control system structure in Fig. 2, the allocation serves as the interface between the motion controller and the actuators. This is advantageous especially in the context of adaptability to diverse vehicle and actuator configurations. On one side, the allocation layer allows flexible reconfiguration without necessarily requiring an adaptation of the motion controller. On the other side, the same motion controller design can be reused across different actuator configurations, as long as the allocation layer is properly adapted.

Despite variable vehicle and actuator configurations, MUM's modular system approach envisages a specific set of configurable actuator types. The MUM demonstrator vehicle as shown in Fig. 3 is one exemplary configuration in which all of the envisaged actuator types are represented. At this stage, the actuator types involved are conventional propulsion and maneuvering devices such as propellers, rudders, maneuvering thrusters, pump jets, and diving planes, as well as variable buoyancy systems (VBS) such as water ballast tanks and movable trim bodies. In favor of a more straightforward handling of system variability, the classical allocation method employed for MUM in [16] disregards actuator dynamics and handles actuator saturation only implicitly. However, when combining conventional propulsion and maneuvering units with VBS in a unified allocation problem, as proposed in [17] for the MUM concept, actuator dynamics cannot be left out of consideration anymore. To this end, the approach in [17] draws from techniques used in dynamically positioning (DP) surface vessels, see [14], which employ an optimization algorithm targeting a power-optimal actuator usage considering their physical characteristics and limits explicitly, thereby factoring in the different operating characteristics of the actuators.

3.3 Actuator Allocation

A common formulation of this actuator allocation strategy is given in [18] as

$$\boldsymbol{f}^* = \underset{\boldsymbol{f},\boldsymbol{s}}{\arg\min} \quad \boldsymbol{f}^T\boldsymbol{Q_f}\boldsymbol{f} + \boldsymbol{s}^T\boldsymbol{Q_s}\boldsymbol{s} \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{T}\boldsymbol{f} = \boldsymbol{\tau}_d + \boldsymbol{s} \tag{1b}$$

$$\boldsymbol{L}\boldsymbol{f} \leq \boldsymbol{k}. \tag{1c}$$

According to the cost function (1a), the optimization problem minimizes the approximated power demands of the actuators ($P \approx \boldsymbol{f}^T\boldsymbol{Q_f}\boldsymbol{f}$) as well as the deviation $\boldsymbol{s}$ between demanded forces and moments $\boldsymbol{\tau}_d$ and the actual forces and moments $\boldsymbol{\tau}$ ($\boldsymbol{s}^T\boldsymbol{Q_s}\boldsymbol{s}$). The positive-definite diagonal matrices $\boldsymbol{Q_f}$ and $\boldsymbol{Q_s}$ serve as the respective weighting factors. For the calculation of $\boldsymbol{s} = \boldsymbol{\tau} - \boldsymbol{\tau}_d$, the equality constraint (1b) incorporates an actuator configuration matrix $\boldsymbol{T}$ that relates the forces and moments $\boldsymbol{\tau}$ generated by the actuators to their individual forces $\boldsymbol{f}$ exerted in the respective actuator-fixed coordinates. The inequality constraints (1c) represent the actuator limitations such as maximum thrust or limited rate of change. As outlined in [19], the formulation (1) of the allocation problem can be implemented and solved as a quadratic programming (QP) optimization problem which has benefits regarding computational effort and guarantees a solution, rendering this approach popular for, e.g., industrial DP applications [20, 21]. The

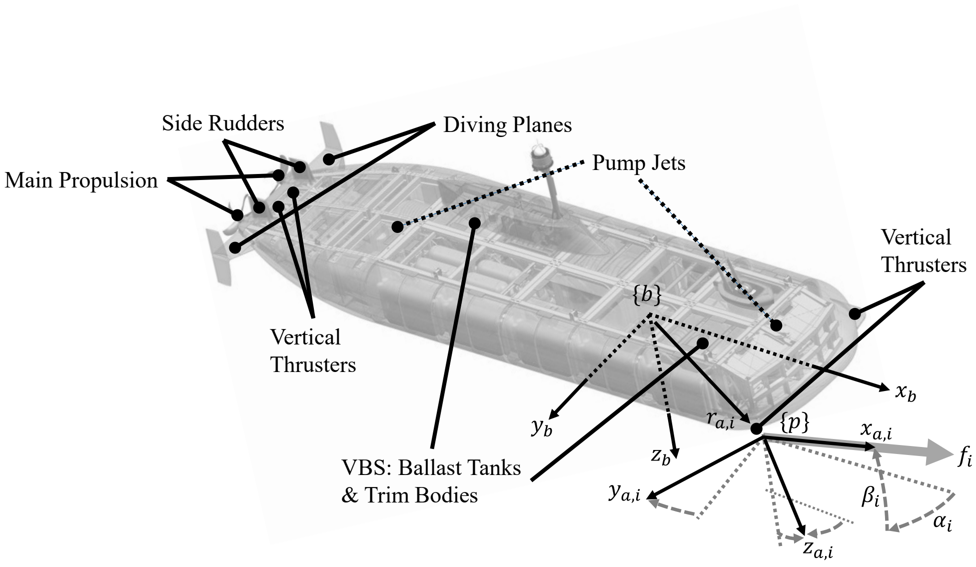


Figure 3: Actuator configuration of the MUM demonstrator, body-fixed reference frame $\{b\}$ of the vehicle, actuator-parallel reference frame $\{p\}$ of a generic actuator, and actuator force vector $\boldsymbol{f}_i$.

integration of conventional propulsion units such as rotatable and non-rotatable thrusters [22], pumpjets [18], or cycloidal propellers [23] in QP-based allocation algorithms has been shown in literature. Additionally, water inflow was taken into consideration in the allocation approach in [19]. When considering control surfaces for QP-based formulations, available approaches focus on combinations of propellers and rudders [14, 24], but not on standalone rudders or diving planes as in the case of MUM. As proposed by [17], VBS can be regarded as a type of maneuvering actuator allowing for integration in the QP-based allocation solution for combined utilization with conventional actuators.

3.3.1 Actuator Types and Modeling Targeting a joint actuator allocation algorithm for the propulsion and maneuvering actuators of MUM as well as for its VBS, the necessary actuator modeling is presented in the following and the incorporation of these actuator models and characteristics in the formulation of the allocation problem is shown.

Main Propulsion and Maneuvering Thrusters For the MUM vehicle class, non-rotatable, fixed-pitch propellers are responsible for the main propulsion, while thrusters can be utilized for slow-speed maneuvering. Both types can be modeled in a similar way starting from considering a force magnitude f_i, i.e. the thrust, which relates to the actuator's control input, i.e. the propeller speed n, through some function $f_i = h(n)$ which incorporates the thruster geometry and characteristics. For DP applications, environmental influences such as propeller inflow are commonly neglected. For static behavior, simple models such as the open-water characteristics

$$f = \rho D^4 K_f |n| n \tag{2}$$

in [25] relate the thrust f and n quadratically using the propeller diameter D, fluid density ρ, and a scaling term K_f. With (2), the control input to the thruster can be calculated from the thruster force given by the actuator allocation.

Given the force magnitude f_i of the thruster i, the vector $\boldsymbol{f}_i^p = [f_i, 0, 0]^T$ describes the actuator's force in the actuator-parallel reference frame $\{p\}$, whose x-axis is defined to be aligned with the direction of the force. Depending on type and orientation of the thruster, it generates f_i at an azimuth angle α_i and an elevation angle β_i with respect to the body-fixed reference frame $\{b\}$, as visualized in Fig. 3. The angles α_i and β_i correspond to the fixed orientation of a non-rotatable propeller or thruster defined by the actuator configuration. In order to express the actuator force in $\{b\}$, the rotation matrix $\boldsymbol{R}(\alpha_i, \beta_i)$ transforms the vector $\boldsymbol{f}_i^p$ using

$$\boldsymbol{f}_i = \boldsymbol{R}(\alpha_i, \beta_i) \boldsymbol{f}_i^p \tag{3}$$

into the force vector $\boldsymbol{f}_i = [f_{x,i}, f_{y,i}, f_{z,i}]^T$ in $\{b\}$ which results in

$$\boldsymbol{f}_i = \begin{bmatrix} f_{x,i} \\ f_{y,i} \\ f_{z,i} \end{bmatrix} = \begin{bmatrix} \cos\alpha_i \cos\beta_i \\ \sin\alpha_i \cos\beta_i \\ -\sin\beta_i \end{bmatrix} f_i, \tag{4}$$

see [26]. Depending on the actuator position $\boldsymbol{r}^b_{a,i} = [x_i \quad y_i \quad z_i]^T$ on the vehicle, the actuator forces $\boldsymbol{f}_i$ produce forces and moments $\boldsymbol{\tau}_i = [X_i \quad Y_i \quad Z_i \quad K_i \quad M_i \quad N_i]^T$ in six DoF acting on the vehicle in $\{b\}$. They are related by $\boldsymbol{\tau}_i = \boldsymbol{T}_i \boldsymbol{f}_i$, where

$$\boldsymbol{T}_i = \begin{bmatrix} \boldsymbol{I}_{3\times3} \\ \boldsymbol{S}(\boldsymbol{r}^b_{a,i}) \end{bmatrix} \quad \text{with} \quad \boldsymbol{S}(\boldsymbol{r}^b_{a,i}) = \begin{bmatrix} 0 & -z_i & y_i \\ z_i & 0 & -x_i \\ -y_i & x_i & 0 \end{bmatrix} \tag{5}$$

is the actuator configuration matrix for actuator i making use of the skew-symmetric matrix $\boldsymbol{S}(\boldsymbol{r}^b_{a,i})$ for the cross product $\boldsymbol{r}^b_{a,i} \times \boldsymbol{f}_i$ yielding the roll, pitch and yaw moments in $\boldsymbol{\tau}_i$.

While the actuator dynamics can be modeled efficiently using linear differential equations for approximation [27], the thruster's physical limits are commonly treated in terms of static lower and upper limits n_{min} and n_{max} as well as a maximum rate of change $\dot{n}_{max}$ of the propeller speed n for implementation in the allocation algorithm. Considering the time interval Δt between consecutive optimization calls, the attainable minimum and maximum propeller speeds $\underline{n}$ and $\overline{n}$ in the next optimization run can be expressed by

$$\underline{n} = \max(n_{min}, n - \dot{n}_{max}\Delta t) \tag{6}$$

$$\overline{n} = \min(n_{max}, n + \dot{n}_{max}\Delta t), \tag{7}$$

which enables us to calculate corresponding dynamic thrust limits $\underline{f}$ and $\overline{f}$ using the relation (2) between speed and thrust at every sampling instant for the next optimization run [18]. Since the allocation formulation (1) works with bounds (1c) on the actuator force, the thrust bounds (6) and (7) can be integrated straightforwardly into the allocation algorithm.

Azimuth Thrusters Thrusters that are rotatable about the azimuth angle expand the set of MUM actuator types by an actuator that is particularly useful for maneuvering. As before, we consider fixed-pitch propellers and thrusters in zero-inflow condition. Considering the two pump jets installed on the demonstrator, which we treat as rotatable thrusters, we confine the modeling and implementation of this actuator type to positive thrust, only. Following [18], the same static behavior (2) of non-rotatable thrusters relating generated thrust and propeller speed applies to rotatable thrusters. While the azimuth angle α is considered as a second control input of the rotatable thruster, we assume the elevation angle β to be zero, so that (4) simplifies to two thrust components

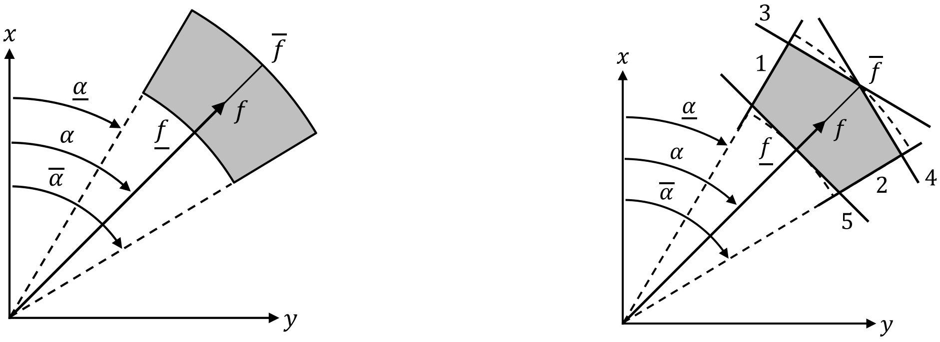
$$\boldsymbol{f}_i = \begin{bmatrix} f_{x,i} \\ f_{y,i} \end{bmatrix} = \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \end{bmatrix} f_i. \tag{8}$$

in the body-fixed reference frame $\{b\}$. With only two of three force components in $\boldsymbol{f}_i$, we construct the configuration matrix the same way as for non-rotatable thrusters using (5), but omit the third column accordingly resulting in a 6×2 configuration matrix $\boldsymbol{T}_i$ for the azimuth thruster. Similar to non-rotatable thrusters, the physical limits in terms of propeller speed and thrust are treated following (6), (7), and (2). Additionally, the maximum rate of change $\dot{\alpha}_{max}$ of the azimuth angle limits the attainable minimum and maximum azimuth angle in the next optimization run according to

$$\underline{\alpha} = \alpha - \dot{\alpha}_{max}\Delta t \tag{9}$$

$$\overline{\alpha} = \alpha + \dot{\alpha}_{max}\Delta t. \tag{10}$$

As outlined in [18], the thrust region formed by the bounds on thrust and angle corresponds to an annulus sector as depicted in Fig. 4a. This does not meet the requirements of convexity, rendering it unusable in a QP formulation. Instead, the thrust region can be approximated by a set of linear inequalities which define an area of attainable thrust vectors $\boldsymbol{f}_i$. For the approximation variant shown in Fig. 4b, the five



(a) Cylindrical thrust region.

(b) Approximation by straight lines.

Figure 4: Approximation of the dynamic thrust region representing the physical limits of the azimuth thruster.

corresponding inequalities

$$\sin(\underline{\alpha})f_x - \cos(\underline{\alpha})f_y \leq c\,|\cos(\underline{\alpha})|\,(\overline{f} - \underline{f}), \tag{11}$$

$$-\sin(\overline{\alpha})f_x + \cos(\overline{\alpha})f_y \leq c\,|\cos(\overline{\alpha})|\,(\overline{f} - \underline{f}), \tag{12}$$

$$\cos(\underline{\alpha})f_x + \sin(\underline{\alpha})f_y \leq \cos(\underline{\alpha} - \alpha)\overline{f}, \tag{13}$$

$$\cos(\overline{\alpha})f_x + \sin(\overline{\alpha})f_y \leq \cos(\overline{\alpha} - \alpha)\overline{f}, \tag{14}$$

$$-\cos(\alpha)f_x - \sin(\alpha)f_y \leq -\underline{f}, \tag{15}$$

express the lines 1 to 5 in this order. The right-hand side terms in line 1 and 2 are necessary for the allocation to be able to rotate the actuator at zero thrust by allowing small negative thrust values. They do not noticeably influence the QP problem solution, but it has to be rechecked regarding the constraints on angle and thrust magnitude afterwards. For further details on the derivation of the inequalities and their implementation in the constraints (1c) of the allocation algorithm, refer to [18].

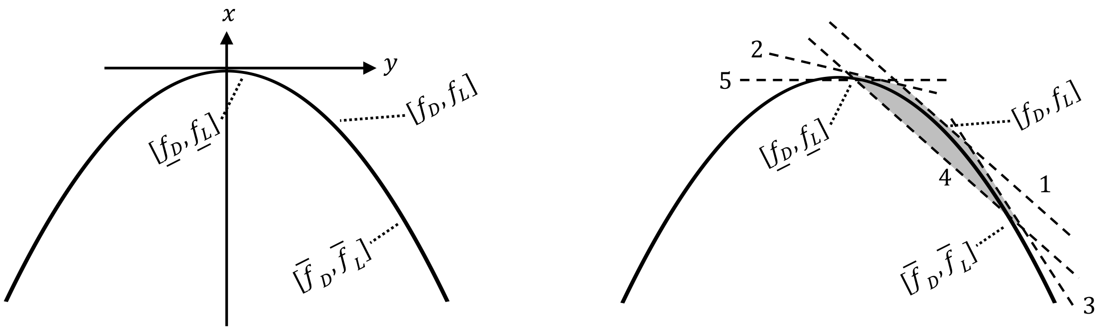
Control Surfaces MUM vehicles can be equipped with control surfaces such as side rudders or diving planes, as is the case with the demonstrator. They generate lift and drag forces depending on the control input, i.e. the rudder deflection, as well as the inflow and the rudder characteristics. When treating control surfaces as actuators, several model simplifications are commonly agreed upon [28]. We neglect any drift angle, assume the inflow due to the vehicle's forward motion to be acting in negative x-direction of the body-fixed coordinate system, and neglect current, inferring that drag and lift act along the axes of the body-fixed coordinate system, that the angle of attack between control surface and inflow corresponds to the deflection of the control surface, and that the inflow velocity corresponds to the vehicle's forward velocity. Moreover, neither of the demonstrator's control surfaces are situated aft of the main propulsion, meaning that there is no direct inflow from the main propulsion toward the side rudders or diving planes. Based on these assumptions and considering control surfaces only for forward motion, we can adhere to [28] and express drag and lift as

$$f_D = -\frac{1}{2}\,\rho\,A\,C_D\,u_f^2\,\delta^2 \tag{16}$$

$$f_L = -\frac{1}{2}\,\rho\,A\,C_L\,u_f^2\,\delta \tag{17}$$

depending on deflection δ, vehicle forward velocity u_f, water density ρ, area A of the control surface, and rudder-specific lift and drag coefficients C_L and C_D. Manipulating the control input δ changes both drag and lift according to these linear and quadratic dependencies of lift and drag on deflection δ. While drag acts along the vehicle's x-direction, side rudders and diving planes differ in the direction of lift which acts along the vehicle's y- or z-direction, respectively, yielding force vectors

$$\boldsymbol{f}_{sr} = \begin{bmatrix} f_{sr,x} \\ f_{sr,y} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{f}_{dp} = \begin{bmatrix} f_{dp,x} \\ f_{dp,z} \end{bmatrix} \tag{18}$$



(a) Drag-lift curve and bounds for a given inflow.

(b) Approximation by straight lines.

Figure 5: Approximation of the curve segment of attainable drag and lift representing the physical limits of the control surface for a given inflow.

in the body-fixed reference frame $\{b\}$ for a single side rudder (sr) or a single diving plane (dp). Thus, similar to rotatable and non-rotatable thrusters, we construct the configuration matrix of a control surface using (5), but omit the third or second column according to the control surface type resulting in a 6×2 configuration matrix $\boldsymbol{T}$ for either one of the control surface types.

We follow previous strategies, e.g. (6) and (7) for thrusters, to express the physical limits of control surfaces in terms of the attainable minimum and maximum rudder deflection in the next optimization run according to

$$\underline{\delta} = \max(\delta_{min}, \delta - \dot{\delta}_{max}\Delta t) \tag{19}$$

$$\overline{\delta} = \min(\delta_{max}, \delta + \dot{\delta}_{max}\Delta t) \tag{20}$$

using the maximum positive and negative rudder deflection $\underline{\delta}$, $\overline{\delta}$ and its maximum rate of change $\dot{\delta}_{max}$. Using the relations (16) and (17), we can calculate dynamic bounds $\underline{f}_D$, $\overline{f}_D$ for drag and $\underline{f}_L$, $\overline{f}_L$ for lift at every sampling instant for the next optimization run. For implementation of the physical limits in the allocation, we need to bear in mind that one control input, i.e. deflection δ, manipulates two force components f_D and f_L at the same time. The bounds on deflection and the resulting upper and lower drag and lift bounds correspond to a segment on the curve of the quadratic relation between lift and drag, as indicated in 5a. This is in contrast to rotatable thrusters, where the bounds on the thrust components form an annulus sector. For approximation of the non-convex constraints on the control surfaces, we express a region of feasible force vectors that approximates the curve segment tightly, as shown in 5b. Defining the coefficient

$$C_{LD} = \frac{2\,C_D}{\rho\,A C_L^2\,u_f^2} \tag{21}$$

that relates drag and lift using $f_D = -C_{LD}\,f_L^2$, we construct five inequalities that circumscribe the feasible area. Line 1 expressed by

$$f_D + (2\,C_{LD}\,f_L)\,f_L \leq C_{LD}\,f_L^2 \tag{22}$$

describes the tangent to the quadratic function at the current drag and lift values $[f_D, f_L]$. Line 2 and 3 are tangents at the lower and upper bounds $\left[\underline{f}_D, \underline{f}_L\right]$ and $\left[\overline{f}_D, \overline{f}_L\right]$, respectively, expressed by

$$f_D + (2\,C_{LD}\,\underline{f}_L)\,f_L \leq \underline{f}_d + 2\,C_{LD}\,\underline{f}_L^2, \tag{23}$$

$$f_D + (2\,C_{LD}\,\overline{f}_L)\,f_L \leq \overline{f}_d + 2\,C_{LD}\,\overline{f}_L^2. \tag{24}$$

Line 4 intersects the lower and upper bounds $\left[\underline{f}_D, \underline{f}_L\right]$ and $\left[\overline{f}_D, \overline{f}_L\right]$ according to

$$-f_D + \frac{\overline{f}_D - \underline{f}_D}{\overline{f}_L - \underline{f}_L}\,f_L \leq -\underline{f}_D + \frac{\overline{f}_D - \underline{f}_D}{\overline{f}_L - \underline{f}_L}\,\underline{f}_L. \tag{25}$$

Lastly, with line 5 we impose a constraint solely on the drag which ensures that drag is never positive, i.e. $f_D \leq 0$. Generally defined by

$$f_D \leq \max(\underline{f}_D, \overline{f}_D), \tag{26}$$

the inequality for line 5 changes depending on the upper and lower bounds on the lift, i.e. if $\operatorname{sgn} \underline{f}_L \neq \operatorname{sgn} \overline{f}_L$, to the expression

$$f_D \leq 0. \tag{27}$$

Approximating the curve segment of feasible force vectors using a tightly encompassing region allows the allocation algorithm to deviate from the curve segment of feasible drag and lift forces. Therefore, after solving the QP problem, the constraints on drag and lift as well as on the control surface deflection need to be rechecked, as was the case for the approximated thrust region of the azimuth thruster.

Variable Buoyancy System Apart from conventional maneuvering actuators, MUM features variable buoyancy systems (VBS) such as water ballast tanks and movable trim bodies. While VBS are typically utilized to maintain neutral buoyancy, they might also be incorporated in automatic motion control tasks such as pitch and depth control. In contrast to the conventional actuators, the VBS exert gravitational forces, i.e. ballast water weight or trim body weight, meaning that their direction relative to the vehicle can change depending on the current orientation of the vehicle. While the water weight f_{BW} of the ballast tank can be regarded as its manipulated variable, the trim body has to be moved in order to alter its influence on the vehicle, i.e. to generate different roll or pitch moments. Thus, the position $\boldsymbol{r}^b_{TB}$ in the vehicle corresponds to the control input of the trim body. By imposing lower and upper limits on the water weight and on the trim body position and considering maximum rates of change of water weight and trim body position, the VBS's physical limits can be formulated in a way similar to non-rotatable thrusters in order to regard them in the allocation problem.

The integration of the VBS actuator type into the allocation formulation (1) has been shown previously in [17], to which the reader is referred for further information.

3.3.2 Combined Actuator Allocation Having defined the actuator types and modeled their characteristics as well as their effect on the vehicle, we can formulate the combined actuator allocation. The actuator types defined in this work form the generalized actuator variables

$$\boldsymbol{u} = \begin{bmatrix} \cdots f_{thr,i} & \cdots & \boldsymbol{f}_{azi,j} & \cdots & \boldsymbol{f}_{rud,k} & \cdots & f_{BW,l} & \cdots & \boldsymbol{r}^b_{TB,m} & \cdots \end{bmatrix}^T \tag{28}$$

and the generalized actuator configuration matrix

$$\boldsymbol{T} = \begin{bmatrix} \cdots \boldsymbol{T}_{thr,i} & \cdots & \boldsymbol{T}_{azi,j} & \cdots & \boldsymbol{T}_{rud,k} & \cdots & \boldsymbol{T}_{BW,l} & \cdots & \boldsymbol{T}_{TB,m} & \cdots \end{bmatrix} \tag{29}$$

yielding the forces and moments $\boldsymbol{T}\boldsymbol{u} = \boldsymbol{\tau}$ acting on the vehicle. Adapting the allocation formulation (1) accordingly, we can use the generalized variables to construct the optimization problem

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}, \boldsymbol{s}}{\arg\min} \quad \boldsymbol{u}^T \boldsymbol{Q_u} \boldsymbol{u} + \boldsymbol{s}^T \boldsymbol{Q_s} \boldsymbol{s} \tag{30a}$$

$$\text{s.t.} \qquad \boldsymbol{T}\boldsymbol{u} = \boldsymbol{\tau}_d + \boldsymbol{s} \tag{30b}$$

$$\boldsymbol{L}\boldsymbol{u} \leq \boldsymbol{k}. \tag{30c}$$

A choice of very high weights in $\boldsymbol{Q_s}$ compared to $\boldsymbol{Q_u}$ reflects the main optimization goal of minimizing the deviation $\boldsymbol{s}$ between demanded and generated forces and moments acting on the vehicle.

Similar to the concatenated matrices and vectors in (28) and (29), the inequality constraints (30c) with $\boldsymbol{L}$ and $\boldsymbol{b}$ can be constructed by concatenating the corresponding matrices and vectors of the individual actuators. Therefore, different actuator configurations with different combinations, positions, and physical specifications of the actuators can be implemented intuitively, given that all actuator types in scope are modeled and adapted to the allocation problem.

During operation, the allocation can be easily adjusted so that certain actuators are excluded from the generation of $\boldsymbol{\tau}$. For example, setting the maximum rates of change of the thrusters to zero and resetting their states $\boldsymbol{f}_{Thr}$ to zero, as well, alters the inequality constraints (30c) of the algorithm such that the thrusters can't contribute to $\boldsymbol{\tau}$. Based on this, allocation modes can be defined and implemented. One may wish to exclude the vertical thrusters and pump jets from the allocation during transit, as the actuators are much less effective due to the higher vehicle velocities and the resulting inflow effects. While dynamically positioning about zero velocity of the vehicle, the control surfaces are basically ineffective, so that they should be excluded from the allocation in order to avoid unreasonable usage. Additionally, this represents an intuitive way to implement allocation reconfiguration in case of actuator failure during operation. Faulty actuators can each be excluded individually, so that the motion control task stays operative using the remaining functional actuators.

4 Environmental Perception

4.1 Requirements

MASS must replace the human lookout with a reliable sensor system in order to enable autonomous navigation in harsh marine conditions as a prerequisite for achieving a high level of autonomy. The selected systems must contend with physical, technical, and legal constraints, which create mutual dependencies that require consideration in software and hardware integration. The performance of AI methods e.g. depends on object distance, camera resolution and field of view, which can lead to challenges in detecting other parties and obstacles at varying ranges. However, there is no clear performance indicator for functional collision detection, as laws and regulations are vague on these technical aspects. From an engineering perspective, this leaves uncertainties concerning the question of *what* has to be solved. Laws and regulations for operating seafaring vessels imply that an autonomous surface vessel requires perception capabilities of a human, but do not define them in values and tolerances.

This creates a strong dissonance in the discussion in which machine vision, using machine learning models to infer the current environmental state, is compared to the more intertwined biological processes of humans.

A holistic approach that can fully mimic human perception and reasoning is therefore beyond the scope of this work. Thus, the focus is primarily on **visual perception** for **surface travel during daytime**. A simplistic but feasible approach is to design a camera system that mimics the visual system of the human lookout as a basis. This poses the challenge of comparing the complex biological visual system of humans to a camera system. Both work by generating a projected image from incoming light, but differ in how this information is collected. A camera sensor constructs an image by counting photons for each pixel, thus generating a single snapshot of the scene over the exposure time. In contrast, the human eye employs photoreceptor cells rather than pixels, with the key distinction being their inhomogeneous distribution, as they are most densely concentrated in the fovea centralis. To compensate for the lower resolution in the peripheral regions and to continuously update the visual projection, the human eye performs constant micro-movements, thereby enhancing information acquisition over time.

The first directly measurable metric relevant when selecting a camera system is visual acuity, which quantifies the sharpness and clarity of vision, specifically the ability to resolve two closely spaced points. It is typically expressed as the angular size of the smallest resolvable detail [29]. For humans with perfect vision under ideal lighting conditions and precise focus on a specific object, the limit of visual acuity is approximately 28 seconds of arc (or 0.0078°) [30]. Furthermore, the Field of View (FoV) for humans can be defined as the spatial extent within which an object can be fixated without head movement. Studies indicate that the binocular FoV of a human covers approximately 180° horizontally and 130° vertically [31, 32].

Although humans posses stereo vision for depth perception, similar to technical systems, it is ineffective for large distances. This limitation arises from the challenge of projecting a three-dimensional object space onto a two-dimensional space with finite resolution, further discussed in Section 5.2.2 on monocular depth estimation. To compensate for this perceptual limitation, the human visual system employs known object sizes as reference cues for depth estimation.

4.2 Hardware

Applying the maximum acuity, i.e. an angular resolution $\alpha = 0,0078^\circ$, for the FoV of an ideal camera system without overlap given values of $FoV_h = 180^\circ$ and $FoV_v = 130^\circ$ for horizontal and vertical FoV, the system would require a pixel resolution of 23077×16667 or 384.6 Megapixels. This estimate does not consider lens distortion or the simple fact that the projection on a flat imagining sensor yields non linear acuity for pixels if a pinhole camera model is assumed. Extending camera perception to achieve a full 360° horizontal field of view would require doubling the pixel count.

The biggest limitation of a system with such capabilities is not necessarily the available image sensors, but data management. A fluent monochrome video at 30 frames per second would yield a data bitrate of 92.32 Gbit/s, which would then require further processing. A generic Ethernet connection would not be suitable for such an application, but is necessary for MUM because the modularity and size of the vehicle class requires the data processing unit to be housed in a pressure-resistant container that may be placed

Table 2: Specification of a single ATL196S camera used in the mum perception system.

Sensor	Sony IMX367 CMOS
Shutter	Global
Sensorsize	$21,7mm$
Resolution	4416×4428
Pixel Size	$3.45\mu m \times 3.45\mu m$
Framerate	27.9 FPS
Connection	5GB POE

quite far from the cameras themselves. Given this environment, the camera system was constrained to a reduced feature set for the selection process.

In order to increase the sight distance, the camera system will be mounted on top of a 7 m tall retractable mast that will be fully extended during surface travel. Its primary function is to detect potential collisions along the path of travel and transmit the footage upon request to an operator in a command and control center. Varying sea states and vehicle movements introduce dynamics in the system have to be compensated for. Since lighting can vary significantly due to operation under different daytime and weather conditions, the camera should deliver an acceptable image even in poor lighting conditions. Based on these considerations and on discussions with experts, the following requirements were placed on the system:

- The cameras should be connect via Ethernet, preferably Power over Ethernet which allows for easy integration and expansion using common network infrastructure.
- The system has to be robust an easy to accommodate in a pressure container.
- Strong external forces such as wind and waves cause the vehicle to tilt. The overall viewing angle must therefore compensate for up to $\pm 20°$ of external rotation.
- The camera requires a large-format imaging sensor as well as a global shutter to reduce motion blur and function in low-light conditions.
- The horizontal field of view should be in the range of at least 120° to 180°.
- All obstacles, including incoming vessels, must be reliably detectable at distances of multiple kilometers.
- For optimal AI-based obstacle detection the camera must produce an image in RGB or an adjacent spectral color space.
- For smooth object identification and tracking, the system must achieve a processing speed of at least 5 Hz, with 30 Hz being desirable.

For the basic camera setup, there are essentially two distinct lines of thought. The first option is to build a single- or multi-sensor platform placed on a pan-tilt unit, as shown in [33]. This configuration offers two key advantages:

1. It significantly reduces the required bandwidth.
2. The camera can mimic human head or eye movements to scan the environment in specific patterns.

For the MUM system, this setup was discarded during the design phase mostly due to its mechanical complexity. Instead, the implemented solution employs a multi-view camera system with intentionally overlapping fields of view. While this method drastically increases the bandwidth, it avoids potential occlusion, simplifies the mechanical integration, and allows the observation of the complete field of view at all times. Following this, the selected MUM camera system consists of three Lucid Vision ATL196S cameras [34] using 18 mm lenses. The specific camera parameters are listed in Table 2. In combination, the perception system achieves a horizontal field of view of $\sim 130°$ with some overlap of the individual view cones. The cameras use the Precision Time Protocol [35] to synchronize the shutter timings, ensuring that frames are synchronized before further processing. This feature requires each camera to interleave the network packet so that all frames arrive at the same time, which in turn limits the cameras' network bandwidth. This reduces the possible frame rate of the cameras to about $\frac{1}{3}$ of the maximum possible frame rate specified in the data sheet, i.e. to ~ 9 FPS.

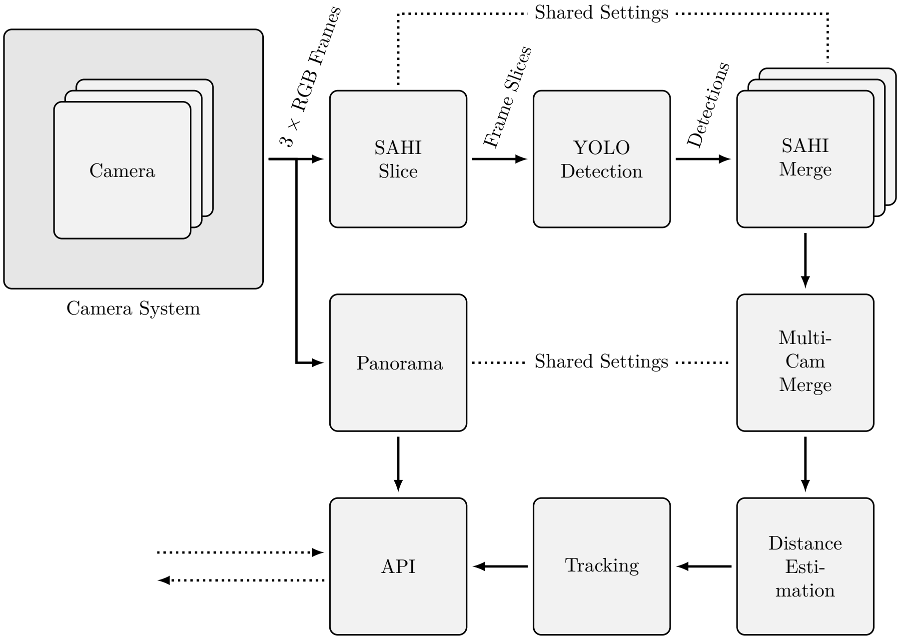


Figure 6: Software pipeline

4.3 Software

The visual perception module plays a crucial role in enhancing collision avoidance capabilities. However, it is essential to recognize that this module is just one component of a broader, more comprehensive collision avoidance system that encompasses traditional methods such as sonar, radar, and the Automatic Identification System (AIS). Given the potential unreliability of AI methods and their experimental nature, the visual perception system is designed to be complemented by AI rather than to be solely dependent on it for accurate collision detection.

The system is tailored to address the specific requirements of the MUM setup, especially the relatively large image size. The architecture of the visual perception system is containerized, allowing access and integration with higher-level systems via a well-defined API. This modular design ensures scalability, flexibility, and ease of maintenance.

Machine learning models are often deployed as micro services, e.g. using specialized frameworks such as Triton Inference Server [36] or Ray [37]. They are designed as designated service architectures, handling incoming requests and spreading the workload in an optimized fashion for distributed computing. The request-based API of such service-based inference frameworks is not suitable for the real-time requirements of autonomous systems such as MUM, especially since onboard processing cannot be further distributed beyond the limited resources used in situ. An alternative framework for a more integrated system would be a pure media pipeline such as gstreamer [38], where information from a media source is processed and transformed in subsequent steps until the results reach an end node. This type of processing is more media-related, having an appropriately optimized pipeline and leveraging metadata to store additional information, enabling strong performance on embedded devices [39]. Other approaches rely on individual pipelines acquiring and processing a single image, with large portions of pre- and postprocessing being negligible in comparison to the execution time and complexity of the machine learning algorithm.

While this may be generally applicable to the given scenario, the overall requirements differ, as the software must process large amounts of data streamed from the camera – up to 5 Gbit/s of raw data – while synchronizing frames and preprocessing the footage. The main goal of the software pipeline is to detect objects, estimate their distance, and track them across multiple frames and camera images, ultimately providing this data via an API. Additionally, the recorded frames should be provided as a panoramic image to a command and control center in case a human wants to inspect the current state and/or has to intervene.

The full pipeline is shown in Fig. 6. It is built on the idea that each module in the pipeline at runtime only depends on a single input x and produces some output y, or – in case of the camera system – is a pure source that only provides data, so that each connection in the figure represents an independent process on the host machine that only becomes active when new data is available and can be restarted at any time in the event of a failure without causing a complete collapse. Data exchange takes place using shared memory and always contains the timestamp of the recording of the source data. This allows for the measurement of pipeline delay and decouples the detections from the original video data, while still allowing for later reassembly of the metadata. A design decision of this particular architecture was that frame drops are to be expected and accepted when there is insufficient computing power is available, e.g. the AI process is slower than new data being fed by the camera module. This means each processing node always works on the most recent data available instead of trying to always process a full input buffer.

Generally, the pipeline works as follows:

- The **camera system** initializes the three mast cameras described in 4.2 and handles their warm-up as well as the frame buffer synchronization. During runtime, the camera images are received and preprocessed. As each frame consists only of raw footage, debayering with edge-aware demosaicing is applied to convert from mono to RGB color space. Further steps include converting from uint8 to float16 for the later inference part and loading the data from RAM to the GPU. The pipline then splits between the visualization path and the machine learning path.
- In the visualization path, the images are stitched together based on their 6D pose and then stored in virtual memory so that the API can access the data.
- The second path handles data extraction such as object detection, distance estimation and tracking. General object detectors are trained on a fixed image size, which requires resizing the input image. This carries the risk of missing distant or small objects. The pipeline therefore uses Slicing Aided Hyper Inference (SAHI) [40] to slice the recorded frames into smaller chunks and create a resized version of the the full image. This process takes place in the **SAHI** module, which also combines the data into a large batch for optimized inference.
- The **YOLO Detection** module uses a modified YOLOv8 network [6, 41] (ScatYOLOv8CBAM + SAHI) which is based on the YOLOv8 architecture [42]. The inference results are bounding boxes for each potential obstacle for each frame slice.
- Based on the previous slicing settings, the pipeline runs a **SAHI Merge** module for each camera. Each module uses a merging process based on non-maximum suppression to reduce multiple detections of the same object in multiple slices to a single bounding box that describes the object location the image.
- Afterwards, **Multi-Cam Merge** transforms the detection coordinates from local camera coordinates to global polar coordinates and applies the same merging algorithm as **SAHI Merge** to embed the detections in a global context.
- Based on the orientation of the MUM, the object detection, and a camera model, the **Distance Estimation** determines where the points of potential obstacles touch the water and then calculates the distance to these determined target points.
- To assess wether an object is moving or static and could cause a collision, the **Tracking** modules uses the Simple Online Realtime Tracking (SOTA) algorithm [43] which only requires observation of the bounding box as input.
- The detected obstacle in global coordinates, including the distance estimate, the tracking ID, and the timestamp of the original frame from which the data was extracted, is stored in a virtual file system in RAM and made available via the **API**.
- A central watchdog that observes the individual modules controls the modules, sends status information and restarts them if necessary, enabling a resilient execution of the data pipeline.

4.4 Object Detection for High-Resolution Images

A key challenge in MUM's perception system stems from its high-resolution camera array with a total of 58.8 megapixels, which is used for visual obstacle detection. Conventional detection architectures based on convolutional neural networks (CNN) (e.g. [44, 45]) are inherently resolution-specific in their design. While these models permit limited resolution adjustment, their receptive fields and feature hierarchies are fundamentally optimized for their native training resolution. CNNs operate through progressive feature aggregation, where each layer combines and transforms features from preceding layers. The effective

Figure 7: Annotated objects in a magnified image taken from the MUM dataset aboard a vessel during a port-to-mission in the Baltic Sea.

receptive field at each layer determines the spatial context available for feature extraction, with earlier layers typically processing localized patterns [46]. This architecture creates resolution-sensitive behavior, as the fixed receptive field structure assumes particular spatial relationships between features.

For inference, these models typically resize the images to their optimized resolution. Especially for high-resolution images, this leads to a loss of information in high-frequency components and limits the detection of small objects. Transformer models [47, 48, 49] have shown better performance, partially due to the fact that their architecture has shown to offer superior feature matching in a global context. However, this advantage comes at the cost of significantly higher computational complexity, particularly when processing high-resolution images.

An alternative approach used by the software pipeline in MUM is Slicing Aided Hyper Inference [40]. The algorithm divides high-resolution input images into smaller, overlapping slices while simultaneously maintaining a resized version of the original image at the detector-optimized resolution. This dual-representation approach creates a multi-scale representation improving small object detection. Fig. 7 shows that such small objects constitute the vast majority of possible obstacles during the MUM system's operation.

The detection results are obtained by inferring the detections on each slice using the object detector, accumulating the local results and recalculating the scale and position of each bounding box for a final global merging process based on their intersection over union. This process allows the detection of small as well as larger objects even for pre-trained models without any further steps. For MUM, the model was improved by further fine-tuning based on data from [50, 51] in conjunction with an additional custom dataset recorded with the MUM hardware. All datasets were preprocessed with SAHI, creating overlapping slices for training. The model employed in MUM is the nano variant which has fevers parameters and is faster in execution, with a slight loss of prediction accuracy. A further speed increase is achieved by compiling and optimizing the model during an initialization phase, batching every sliced image for a single inference call.

4.5 Monocular Distance Estimation

Effective path planning in three-dimensional space necessitates precise localization and tracking, particularly when obstacles are anticipated. In the context of MUM, radar sensors are widely regarded as the most reliable medium for this task. However, these sensors are not without limitations. Specifically, objects with low radar reflectivity, such as fiberglass hulls or small underwater targets, may not adequately respond to radio waves, rendering their detection unreliable. To mitigate this issue, supplementary sensor data must be incorporated, even if the measurements show high degrees of uncertainty.

One such unreliable depth estimation can be obtained from the camera and is known in the literature as monocular depth estimation. The reason for the unreliability of such methods is that when projecting a 3D scene onto a 2D camera sensor, a degree of freedom is lost, which makes the inverse transformation an underdetermined system. Similar to humans, AI methods [52] can use scene information and an accumulated knowledge base to fill in the lost information and produce estimates of how far away an object might be. Usually, such methods do not estimate the "real" distance, but rather a unitless disparity map that shows how far a pixel is from the camera relative to every other pixel in the image. Utilizing precise camera parameters, this disparity map can be transformed into an accurate depth map. However, the computational cost of these methods is substantial, rendering them infeasible within the constraints of MUM's computational power.

To improve computational efficiency, MUM employs a model-based algorithm that provides a more efficient approach to object distance estimation. Changes in height due to wave movement are negligibly small compared to the detected object distances from the vehicle, so it can be assumed that both the vehicle and all detected obstacles are accurately localized at the same height, i.e. the sea surface, and thus the z-axis (height) is eliminated. Building upon this assumption, the degrees of freedom for localization are reduced to two independent variables describing the location on the water surface. For the implemented model, this surface is represented as the surface of a sphere in 3D space with the radius of earth.

Assuming this holds true and given the precise transformation $\boldsymbol{T}_{CM}$ from the sensor space of a single camera to the global MUM vehicle space, with the axes of the coordinate system aligned to the global world coordinates so that the vehicle's roll, pitch, and yaw angles (e.g., caused by waves) can be included in this transformation, the model distance estimation for each pixel on the sensor can be solved using a simple ray-sphere intersection calculation based on ray tracing. The distance d can be estimated by calculating the intersection of a direction vector $\boldsymbol{v}$, originating from pixel position $\boldsymbol{p}$ and aligned with the camera's line of sight, with a sphere located at $\boldsymbol{s}$ with radius r, as described in [53] and formulated as

$$\boldsymbol{v}_M = \boldsymbol{T}_{CM}\boldsymbol{v}^T \tag{31a}$$

$$t_{ca} = \boldsymbol{v}_M\boldsymbol{s}^T \tag{31b}$$

$$h = \sqrt{\boldsymbol{v}\boldsymbol{v}^T - t_{ca}^2} \tag{31c}$$

$$t_{hc} = \sqrt{r^2 - h^2} \tag{31d}$$

$$d = t_{ca} - t_{hc} \tag{31e}$$

Given equations (31), the distance of every pixel to the water surface can be calculated. With the pixel locations describing the corners of a detected bounding box, the distance of an obstacle can therefore be estimated taking the minium positive distance of all corners. As the software pipeline uses the merged bounding boxes in polar coordinates, the directional vectors of these bounding boxes have to be transformed from the directional angles to Euclidean coordinates obtaining the three dimensional directional vectors $\boldsymbol{v}$ necessary for the calculations.

5 Simulations and Experimental Results

5.1 Maneuvering System Simulation Results

With the actuator allocation laying the foundation for higher-level motion control, we focus our results in the following on tests that showcase the workings of the algorithm. For this, the test procedure specifies control forces and moments that are to act on the vehicle. These demands are fed to the allocation which distributes them among the individual actuators using their respective control inputs. The allocation is formulated for and applied to the envisioned actuator configuration of the MUM demonstrator vehicle as specified in Section 3. The chosen sequence of manually commanded control forces and moments $\boldsymbol{\tau}_d$ in this test corresponds to a meaningful motion sequence of the demonstrator induced by the allocated actuators, demonstrated using a six DoF simulation model of the motion behavior of the demonstrator. This simulation model is based on engineering and design data of the demonstrator provided by TKMS GmbH, i.e. the module configuration, rigid-body and hydrostatic data as well as parameters of the propulsion and maneuvering systems. Moreover, the model incorporates hydrodynamic data, i.e. acceleration-dependent added mass coefficients as well as linear and nonlinear damping coefficients, provided by the project partners of "Technische Universität Berlin" (TU Berlin), chair of Design and Operation of Maritime Systems. The hydrodynamic data describe the hydrodynamic forces and moments during a submerged, steady-state transit operation of the demonstrator at speeds of 6 m/s. For testing of the allocation algorithm in surfaced operation, we adapted the hydrodynamic coefficients heuristically to represent the vehicle's motion behavior at surface. Both the actuator allocation as well as the motion model are implemented and tested in MATLAB/Simulink.

As visualized in Fig. 8, the vehicle initially is surfaced being in forward motion at a surge velocity $u = 2$ m/s. As defined in the design phase of the control system, this higher surge velocity calls for the transit operation mode. Here, the controller demands only a surge force X and a yaw moment N to fulfill control objectives such as desired surge velocity or desired heading. As illustrated in Fig. 9, we manually demand a positive yaw moment at $t = 10$ s in order to initiate a turning maneuver toward starboard. Following the large step in the yaw moment, the allocation quickly acts by rotating the side rudders toward the starboard side. Additionally, it engages the two main propulsion units in opposite

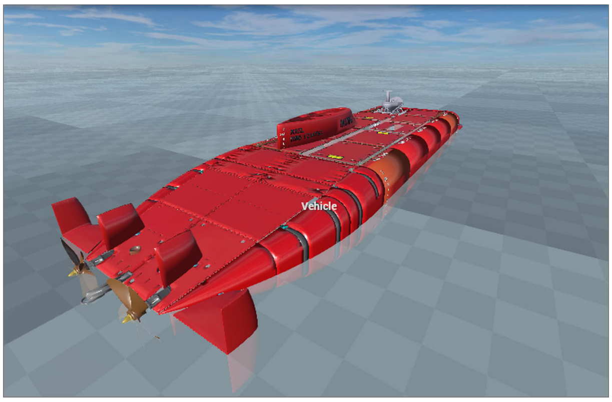


Figure 8: Visualization of the MUM demonstrator at the beginning of the simulated maneuver sequence.

directions for a short amount of time. This is a measure to counteract the limited rotation rate of the side rudders in an effort to generate the large yaw moment demand as quickly as possible. The resulting turn of the vehicle is obvious in the bottom of Fig. 9. Shortly after, the main propulsion acts in unison in forward direction to compensate for the rudder-induced drag. In the course of the maneuver, the rudders gradually loose lift effect due to the vehicle slowing, which in turn is compensated for by the allocation at $t \approx 40\ s$ by utilizing the main propellers in an opposing manner, again. Shortly after, the vehicle is demanded to come to a halt in the form of a negative surge force X commanded at $t \approx 45\ s$, to which the allocation reacts by permitting deviations in the yaw moment N in favor of generating demands in X. With the vehicle slowing further, the allocation eventually switches to DP mode employing actuators that are more favorable for DP operations. While the control surfaces are being excluded from the allocation due to them losing effect at slow velocity, now the pump jets are factored in, instead. Having been omitted at higher transit velocities due to inflow effects, the pump jets now are allocated to generate the yaw moment and help in generating the negative surge force. With the vehicle coming to a halt at $t \approx 85\ s$ with a heading $\psi \approx 30°$, both demands X and N are lifted resulting in neutral allocated control inputs of the actuators.

As a follow-up maneuver, we proceed with submerging the vehicle by demanding a heave force Z at $t = 100\ s$. Consequently, the allocation engages the four vertical thrusters as quickly as possible to their maximum speeds. We assume a neutrally buoyant vehicle at the beginning of the maneuver, which results in the vehicle responding almost immediately to the generated thrust and gaining depth, as can be seen in Fig. 9. In response to the generated heave force, the trim bodies move a few centimeters to the starboard side to compensate for roll moments due to asymmetric lever arms of the thrusters relative to the vehicle's center of gravity. The vertical thrusters are not capable of fulfilling the great heave force demand Z on their own. Consequently, as the maneuver progresses, the allocation orders the ballast tanks to fill up with more water until the demand is met at $t \approx 285\ s$. Accordingly, the trim bodies continue to compensate for roll moments induced by asymmetric lever arms of the tanks filling up. At $t = 300\ s$, the commanded heave force is lifted, i.e. $Z = 0$ N, to which the allocation reacts with high negative speeds of the vertical thrusters in order to compensate for the additional water ballast in the tanks and generate a zero heave force in the aggregate. For the remainder of the maneuver, the allocation orders the ballast tanks to discharge water in order to transfer the forces from the thrusters to the VBS until the vehicle is neutrally buoyant at $t = 500\ s$, again.

5.2 Perception System Experimental Results

5.2.1 Object Detection Within MUM's implemented software pipeline (Section 4.3), a critical remaining question concerns the fundamental performance limits of the system. For the hardware selection, it was assumed that the images would not require mechanical stabilization due to the camera's high vertical field of view, thus reducing mechanical complexity. Analysis of the dataset collected for MUM's visual obstacle detection appears to confirm this assumption. Fig. 10 presents a spatial heat map visualization of the aggregated bounding box distribution across the entire annotated dataset. The dataset was collected during a port-to-port mission, which suggests that other vessels and obstacles are more likely to be encountered at closer ranges, particularly during maneuvers in the harbor area, as opposed to open-sea transits. Upon examining the spatial obstacle distribution, it is notable that a majority of the annotated

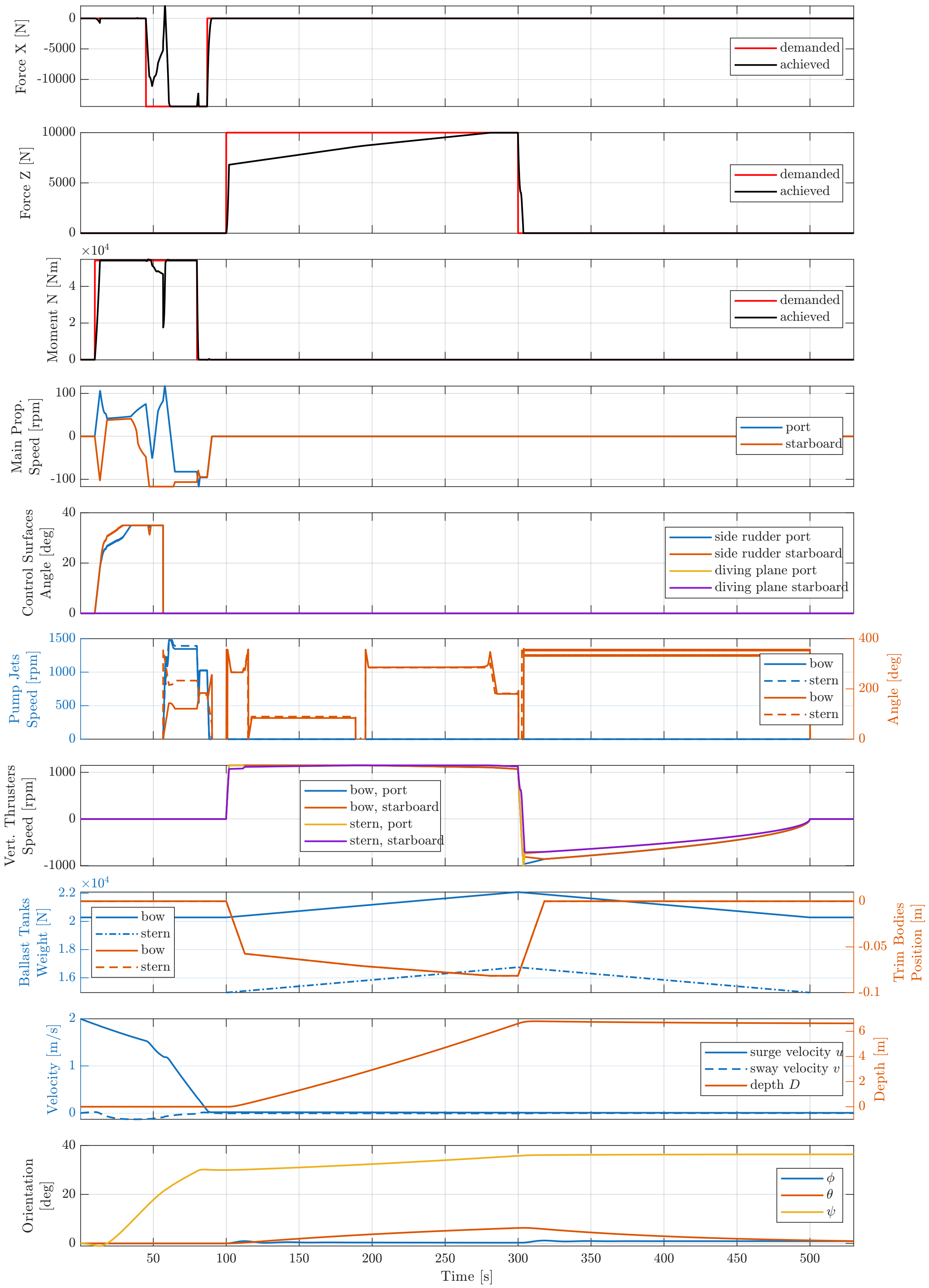


Figure 9: Simulation results of the exemplary maneuver sequence.

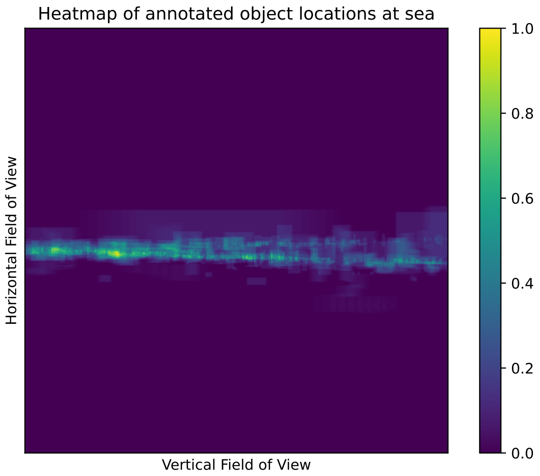


Figure 10: Heat map showing the spatial distribution of annotated obstacles in the custom MUM detection dataset.

objects are situated near the horizon line. This implies that, under normal operating conditions and with proper virtual stabilization, the horizontal field of view can be strategically cropped to minimize the image area requiring further computational processing. Consequently, this reduction in image data leads to lower computational overhead, resulting in faster detection and tracking of potential obstacles. Furthermore, the observed distribution indicates that close-range encounters with other vessels are relatively rare, suggesting that any detected object can be treated as an anomaly deserving special attention, irrespective of whether a collision is imminent.

A crucial aspect of integrating the detector into the hardware pipeline is determining the theoretical maximum detection range, defined as the farthest distance at which objects can be reliably detected within the system's optical and computational limitations. This limitation is particularly pronounced in the detection of small objects, a well-studied problem in the literature [54, 55, 56], as they offer fewer distinctive features for identification and classification. The widely used object detection dataset COCO [57] considers any object with an area smaller than 32 pixels a small object.

Equation (32) computes the projected span i of an object in pixels, given the object's maximum length l_{obj} and the function $r(d_{obj})$, defined in (32b). The function $r(d_{obj})$ calculates the real-world size represented by a single pixel at a distance d_{obj} between object and camera sensor, given the camera's focal length f and the physical size p of a pixel on the camera sensor.

$$r_p(d_{obj}) = \frac{p}{f}\, d_{obj} \tag{32a}$$

$$i = \frac{l_{obj}}{r_p(d_{obj})} \tag{32b}$$

Using the parameters of the camera sensors deployed in MUM's perception system shown in Table 2, the theoretical limits can be calculated.

- An incoming Panamax-sized container ship with a width of 33.5 meters at a distance of 5 kilometers would span $\sim$ 35 pixels and is thus at the borderline of being considered a small object.
- A small pleasure craft with a beam of 4 meters would have to be more than 650 meters away to be considered a small object, improving potential detection and thus reducing the risk of collision.

5.2.2 Monocular Depth Estimation For monocular depth estimation, a static scene was simulated using the algorithm outlined in Section 4.5, with camera parameters specified in Table 2. Although this simulation does not account for finer details such as the uneven ocean surface, it calculates the distance based on simulated light rays intersecting the camera sensor, assuming a realistic camera pose and a spherical Earth model. The results of this simulation for a single camera sensor are presented in Fig. 11.

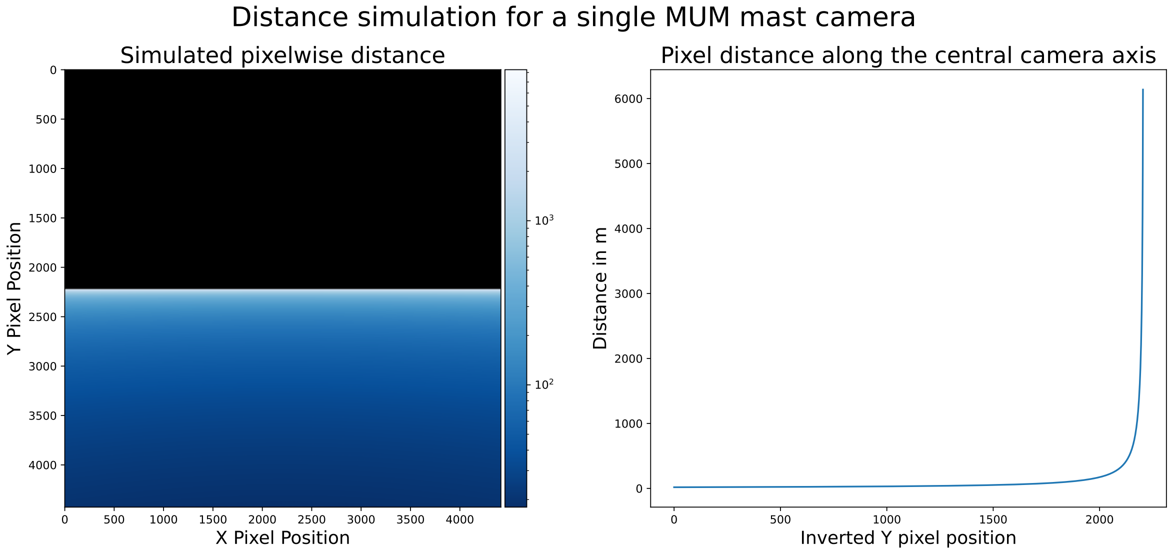


Figure 11: Monocular distance for a high-resolution camera 7 m above sea level. The simulation is based on a ray tracing approach, calculating the intersection of the camera rays with a sphere (earth). This simulation is based on a pinhole camera model parametrized as the one used in the MUM mast camera.

The left figure displays the estimated distance for each spatial pixel of the central camera. The right figure illustrates the distance along the central vertical axis of the camera revealing an exponential relationship between pixel position and estimated distance, with values increasing rapidly as objects approach the horizon. This non-linear relationship gives rise to significant challenges in estimating larger distances, as small errors in object localization propagate exponentially, producing substantially larger errors in distance estimation as the range increases. The cause of this effect lies in the projection of the three-dimensional scene onto a two-dimensional plane, which is further accentuated by the Earth's curvature. In this projected space, parallel lines extending from the camera into the distance converge at a common vanishing point, theoretically located at infinity. Consequently, each point along these parallel lines corresponds to a unique real-world distance, ranging from a finite value to infinity, and is thus preserved in the projection. Furthermore, the Earth's curvature introduces an additional geometric compression, where the apparent height of a target decreases with increasing distance from the vantage point, ultimately reaching a point where the surface is no longer visible at the horizon line.

This limitation explains why distance estimates based on stereo vision, as well as human perception, are generally reliable only at close ranges. Projection-based sensors that provide pixel-wise depth measurements are also constrained by this limitation, and while stereo cameras can theoretically be used for this purpose, they would require an impractically large baseline to achieve accurate results. Even then, their performance would deteriorate at longer ranges due to insufficient spatial resolution, leading to inaccurate estimates. Consequently, the values represented in Fig. 11 are beyond the measurement capabilities of available sensors, which lack the necessary accuracy to provide reliable estimates.

The horizon crowding phenomenon evident in Fig. 10 directly results from this distance-projection relationship. Maritime vessels attempt to maintain a safe distance, and any object farther than ∼100 meters away is projected into the upper half of the vertical space occupied by the sea surface.

The monocular distance estimates in MUM exhibit significant uncertainty especially for distant objects, which must be accounted for by higher-level systems. A primary source of this uncertainty stems from imperfect bounding box detections, which may not precisely encapsulate obstacles, as well as minor localization errors of a few pixels. Additionally, uncertainty in vehicle orientation measurements propagates into the distance estimation, further contributing to the overall error.

6 Conclusion

In summary, this contribution presented current developments within the MUM2 research project. In this context, the innovative modular XLUUV vehicle platform and the applications that can be derived from it have been introduced. The envisaged approach of autonomous port-to-port operations marks a significant challenge. With regard to vehicle control, the actuator allocation was described as an optimization problem, which was identified as an essential component of the vehicle maneuvering system in order to be able to carry out the multi-domain operations of the XLUUV. The proposed strategy adheres to the control system generalization of the modular MUM UUV, enables an intuitive implementation of the actuators, and allows the algorithm to take advantage of the different actuator characteristics and overcome the ambiguity in the utilization of the multitude of actuators. The maneuvering system was tested in extensive simulations using a physical vehicle model. For surface navigation in particular, a camera-based system was presented for reliable environmental perception and as an efficient replacement for the human lookout. The performance of the perception system was evaluated in tests with real data.

The work presented so far provides the crucial basis for developing higher-autonomy functions, where key aspects are automated route or trajectory planning, obstacle and collision avoidance strategies as well as decision-making frameworks for dynamic environments. Since these capabilities are crucial for autonomous port-to-port operations, the project consortium will address these research issues in future developments.

Acknowledgments

The work has been done within the project MUM2, funded by the German Federal Ministry for Economic Affairs and Energy (BMWE) under the registration number 03SX543. The authors wish to express their gratitude for funding the joint research project and the project partners TKMS GmbH, ATLAS ELEKTRONIK GmbH, EvoLogics GmbH, Fraunhofer Institute for Communication, Information, Processing and Ergonomics (FKIE), DLR e.V., Technische Universität Berlin (Design and Operation of Maritime Systems) and University of Rostock (Institute of Automation) for their support and cooperation.

References

[1] S. Ritz, A. Loewe, J. Bauer, and M. Kurowski, “By-Design Risk Mitigation for Large Uncrewed Underwater Vehicles (UUVs),” *Proceedings of the MARESEC 2024*, Nov. 2024, publisher: Zenodo.

[2] Y. Xing, M. C. Ong, T. Hemmingsen, K. E. Ellingsen, and L. Reinås, “Design Considerations of a Subsea Shuttle Tanker System for Liquid Carbon Dioxide Transportation,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 143, no. 4, p. 045001, Aug. 2021.

[3] M. Golz, F. Boeck, S. Ritz, G. Holbach, N. Richter, P.-M. Haselberger, W. H. Wehner, M. Schiemann, E. Rentzow, T. Muller, and T. Jeinsch, “MUM - Large Modifiable Underwater Mother Ship: Requirements and Application Scenarios,” in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*. Kobe: IEEE, May 2018, pp. 1–9.

[4] S. Fan, N. Bose, and Z. Liang, “Polar AUV Challenges and Applications: A Review,” *Drones*, vol. 8, no. 8, p. 413, Aug. 2024, publisher: MDPI AG.

[5] M. Szczotka, “AUV launch & recovery handling simulation on a rough sea,” *Ocean Engineering*, vol. 246, p. 110509, Feb. 2022.

[6] B. Kiefer, L. Žust, M. Kristan, J. Perš, M. Teršek, A. Wiliem, M. Messmer, C.-Y. Yang, H.-W. Huang, Z. Jiang, H.-C. Kuo, J. Mei, J.-N. Hwang, D. Stadler, L. Sommer, K. Huang, A. Zheng, W. Chong, K. Lertniphonphan, J. Xie, F. Chen, J. Li, Z. Wang, L. Zedda, A. Loddo, C. Di Ruberto, T.-A. Vu, H. Nguyen-Truong, T.-S. Ha, Q.-D. Pham, S.-K. Yeung, Y. Feng, N. T. Thien, L. Tian, A. Michel, W. Gross, M. Weinmann, B. Carrillo-Perez, A. Klein, A. Alex, E. Solano-Carrillo, Y. Steiniger, A. B. Rodriguez, S.-Y. Kuan, Y.-H. Ho, F. Sattler, M. Fabijanić, M. Šimunec, and N. Kapetanović, “2nd workshop on maritime computer vision (macvi) 2024: Challenge results,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, January 2024, pp. 869–891.

[7] International Maritime Organization (IMO), “Final Act of the International Conference on Revision of the International Regulations for Preventing Collisions at Sea, 1972, with attachments including the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs),” https://digitallibrary.un.org/record/820, London, 1972, United Nations Document A/CONF.70/29.

[8] P. Margat and M. G. Stadermann, “The application of colregs by autonomous and unmanned vessels: Issues raised by situational awareness, night-time navigation and good seamanship,” in *4th European Workshop on Maritime Systems, Resilience and Security 2024 (MARESEC 24)*, November 2024. [Online]. Available: https://elib.dlr.de/209985/

[9] M. Akdağ, P. Solnør, and T. A. Johansen, “Collaborative collision avoidance for Maritime Autonomous Surface Ships: A review,” *Ocean Engineering*, vol. 250, p. 110920, Apr. 2022.

[10] T. Hahn, R. Damerius, C. Rethfeldt, A. U. Schubert, M. Kurowski, and T. Jeinsch, “Automated maneuvering using model-based control as key to autonomous shipping,” *at – Automatisierungstechnik*, vol. 70, no. 5, pp. 456–468, 2022, publisher: DeGruyter Oldenbourg.

[11] L. Pivano and A. Karlsen, “DNV GL DP Capability Standard ST-0111 - Challenges and Updates One Year from Publishing,” *Proc. of the MTS DP Conference*, pp. 1–8, 2017.

[12] J. A. Ramírez-Macías, R. E. Vásquez, A. J. Sørensen, and S. Sævik, “Motion Feasibility Framework for Remotely Operated Vehicles Based on Dynamic Positioning Capability,” *Journal of Offshore Mechanics and Arctic Engineering*, vol. 143, no. 1, p. 011201, Feb. 2021.

[13] M. Greve, M. Kurowski, S. Ritz, M. Golz, L. N. Vijayasarathi, N. Bayazit, and E. Rentzow, “Design of the Propulsion System for the Autonomous XLUUV MUM,” in *Volume 5A: Ocean Engineering.* Hamburg, Germany: American Society of Mechanical Engineers, Jun. 2022, p. V05AT06A035.

[14] T. A. Johansen and T. I. Fossen, “Control allocation - A survey,” *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[15] S. Ritz, M. Golz, F. Boeck, G. Holbach, E. Rentzow, M. Kurowski, T. Jeinsch, W. H. Wehner, N. Richter, and T. Voß, “Large Modifiable Underwater Mothership: A Case Study for Ocean Bottom Nodes Deployment and Recovery,” in *Proceedings of the SPE Offshore Europe Conference and Exhibition.* Aberdeen, UK: SPE, Sep. 2019, p. D031S013R001.

[16] E. Rentzow, M. Kurowski, and T. Jeinsch, “Generalized Approach for Modeling and Control of Structurally Variable Underwater Vehicles,” in *OCEANS 2024 - Singapore*, 2024, pp. 1–9.

[17] C. Rethfeldt and T. Jeinsch, “Combined Actuator Allocation for Underwater Vehicles with Variable Buoyancy Systems Using QP-Based Optimization,” in *OCEANS 2024 - Halifax*, 2024, pp. 1–7.

[18] E. Ruth, A. J. Sørensen, and T. Perez, “Thrust Allocation with Linear Constrained Quadratic Cost Function,” *IFAC Proceedings Volumes*, vol. 40, no. 17, pp. 337–342, 2007, 7th IFAC Conference on Control Applications in Marine Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015321182

[19] P. Koschorrek and M. Kosch, “An Approach to QP-based Thrust Allocation considering Inflow,” *IFAC-PapersOnLine*, vol. 54, no. 16, pp. 126–131, 2021, 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896321014853

[20] S. Skjong and E. Pedersen, “Nonangular MPC-Based Thrust Allocation Algorithm for Marine Vessels - A Study of Optimal Thruster Commands,” *IEEE Transactions on Transportation Electrification*, vol. 3, no. 3, pp. 792–807, Sep. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7887691

[21] P. Koschorrek, T. Hahn, and T. Jeinsch, “A Thrust Allocation Algorithm Considering Dynamic Positioning and Roll Damping Thrust Demands Using Multi-step Quadratic Programming,” *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 438–443, 2018, 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318321372

[22] C. de Wit, “Optimal Thrust Allocation Methods for Dynamic Positioning of Ships,” Master’s thesis, Delft University of Technology, 2009. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid:4c9685ac-3f76-41c0-bae5-a2a96f4d757e

[23] P. Koschorrek, M. Palm, and T. Jeinsch, “A Dynamic Allocation Strategy for Voith Schneider Propeller,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1127–1132, 2017, 20th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317307425

[24] A. U. Schubert, N. Eisenblätter, R. Damerius, and T. Jeinsch, “Automatic Maneuvering of Vessels with Power-Optimized Thrust Allocation,” in *Conference Proceedings of iSCSS*, vol. iSCSS 2024, 2024. [Online]. Available: http://library.imarest.org/record/11146

[25] Ø. N. Smogeli, “Control of Marine Propellers: from Normal to Extreme Conditions,” Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway, 2006. [Online]. Available: http://hdl.handle.net/11250/237615

[26] F. Dukan, “ROV Motion Control Systems,” Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway, 2014. [Online]. Available: http://hdl.handle.net/11250/239258

[27] M. Kurowski, H. Korte, and B. P. Lampe, “AGaPaS - a new approach for Search-and-Rescue-Operations at sea,” *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 73–78, 2012, 9th IFAC Conference on Manoeuvring and Control of Marine Craft. [Online]. Available: https://doi.org/10.3182/20120919-3-IT-2046.00013

[28] T. I. Fossen, T. A. Johansen, and T. Perez, “A Survey of Control Allocation Methods for Underwater Vehicles,” in *Underwater Vehicles*, A. V. Inzartsev, Ed. Rijeka: IntechOpen, 2009, ch. 7. [Online]. Available: https://doi.org/10.5772/6699

[29] M. F. Deering, “The limits of human vision,” in *2nd international immersive projection technology workshop*, vol. 2, no. 1, 1998.

[30] M. Kalloniatis and C. Luu, “Visual acuity,” *Webvision: The Organization of the Retina and Visual System*, 2007.

[31] H. M. Traquair, *An introduction to clinical perimetry.* London: Kimpton, 1927, ch. Chapter 1 THE NORMAL FIELD OF VISION, pp. 4–5.

[32] H. Strasburger, “Seven myths on crowding and peripheral vision,” *i-Perception*, vol. 11, no. 3, p. 2041669520913052, 2020.

[33] SEA.AI GmbH. Sea.ai sentry: 360° ai awareness for maritime safety. [Online]. Available: https://www.sea.ai/product-sentry/

[34] *ATL196S Camera*, Lucid Vision Labs, 4600 Jacombs Rd #110, Richmond B.C. Canada, V6V 3B1, 3 2025, version 1.41.0.0. [Online]. Available: https://thinklucid.com/de/product/atlas-19-6-mp-imx367/

[35] “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)*, pp. 1–499, 2020.

[36] NVIDIA Corporation, “Triton Inference Server: An Optimized Cloud and Edge Inferencing Solution.” [Online]. Available: https://github.com/triton-inference-server/server

[37] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, W. Paul, M. I. Jordan, and I. Stoica, “Ray: A distributed framework for emerging AI applications,” *CoRR*, vol. abs/1712.05889, 2017. [Online]. Available: http://arxiv.org/abs/1712.05889

[38] “gstreamer: open source multimedia framework,” version 1.24.13. [Online]. Available: https://gstreamer.freedesktop.org/

[39] M. Ham, J. J. Moon, G. Lim, W. Song, J. Jung, H. Ahn, S. Woo, Y. Cho, J. Park, S. Oh, and H.-S. Kim, “Nnstreamer: Stream processing paradigm for neural networks, toward efficient development and execution of on-device ai applications,” 2019. [Online]. Available: https://arxiv.org/abs/1901.04985

[40] F. C. Akyon, S. O. Altinuc, and A. Temizel, “Slicing aided hyper inference and fine-tuning for small object detection,” in *2022 IEEE international conference on image processing (ICIP).* IEEE, 2022, pp. 966–970.

[41] B. Carrillo-Perez, A. B. Rodriguez, S. Barnes, and M. Stephan, “Improving yolov8 with scattering transform and attention for maritime awareness,” in *2023 International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2023, pp. 1–6.

[42] G. Jocher, J. Qiu, and A. Chaurasia, “Ultralytics YOLO,” https://github.com/ultralytics/ultralytics, Jan. 2023, Version: 8.0.0, License: AGPL-3.0, Accessed: 2025-06-10.

[43] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. Ieee, 2016, pp. 3464–3468.

[44] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[45] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[46] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[48] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som *et al.*, "Image as a foreign language: Beit pretraining for vision and vision-language tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 175–19 186.

[49] Z. Zong, G. Song, and Y. Liu, "Detrs with collaborative hybrid assignments training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 6748–6758.

[50] L. Žust, J. Perš, and M. Kristan, "Lars: A diverse panoptic maritime obstacle detection dataset and benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 304–20 314.

[51] L. A. Varga, B. Kiefer, M. Messmer, and A. Zell, "Seadronessee: A maritime benchmark for detecting humans in open water," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 3686–3696.

[52] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 3, pp. 1623–1637, 2020.

[53] Scratchapixel, "A minimal ray-tracer: Ray-sphere intersection," accessed: 2025-06-19. [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-sphere-intersection.html

[54] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "Uav-yolo: Small object detection on unmanned aerial vehicle perspective," *Sensors*, vol. 20, no. 8, p. 2238, 2020.

[55] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "Yolo-z: Improving small object detection in yolov5 for autonomous vehicles," *arXiv preprint arXiv:2112.11798*, 2021.

[56] B. Carrillo-Perez, A. B. Rodriguez, S. Barnes, and M. Stephan, "Enhanced small ship segmentation with optimized scatyolov8+ cbam on embedded systems," in *2024 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2024, pp. 13–18.

[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 2014, pp. 740–755.