Towards Robust Autonomous Underwater Docking for Long-Term Under-Ice Exploration

Tom Creutz[1], Bilal Wehbe[1], Sascha Arnold[2], Marc Hildebrandt[1]
[1]DFKI - Robotics Innovation Center
[2] Kraken Robotics
Germany, 28359 Bremen
Email: [1]{firstname.surname}@dfki.de, [2]sarnold@krakenrobotics.com

***Abstract*—Autonomous underwater docking is a necessary requirement for achieving long-term term residency for Autonomous Underwater Vehicles (AUVs). In this work, we propose a docking method that is capable of achieving robust docking by proposing a state machine and extending our localization framework with visual features from Apriltag marker measurements and USBL 2D position measurements. In total we executed 201 docking sequences of which 194 were succesful, thus yielding a total success rate of 96.5%. To further test the robustness of the proposed method, multiple experiments were conducted where feedback from visual markers was actively denied during the approach of the vehicle to its designated docking station. Our results show that the proposed method was able to complete 73 successful docking maneuvers out of 76 attempts (96%) without any visual aiding.**

***Index Terms*—AUV, autonomous docking, under-ice exploration, localization**

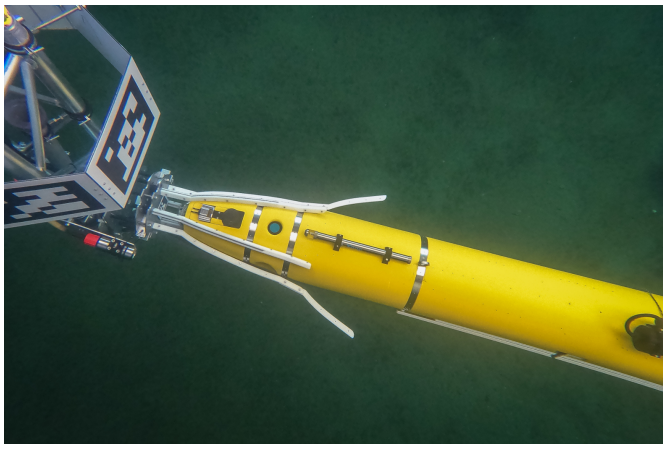

Fig. 1. Exploration AUV *Deepleng* [5] after successful docking.

I. Introduction

Autonomous under-ice exploration has been gaining traction throughout the last 2 decades, with a particular interest into the search of life under the surface of icy moons [1], [2]. When using Autonomous Underwater Vehicles (AUVs) in such scenarios, it is crucial to make sure that the AUV is able to return to its home-base, usually a docking station [3]. This is especially a challenge as the docking station is the only place for the AUV to be retrieved again due to the impossibility to simply surface. When considering long-term operations, the AUV is also supposed to recharge and upload collected data at the docking station.

This work describes our implementation and corresponding experiments towards robustifying autonomous underwater docking as crucial part of long-term under-ice exploration.

For this we extend our existing localization framework [4] by integrating delayed Ultra Short Baseline (USBL) position measurements for homing and visual feature measurements for near-range localization in addition to the already used Inertial Measurement Unit (IMU), Doppler Velocity Log (DVL), actuator based forces and torques and pressure sensor measurement integration. In contrast to pure visual servoing docking methods, our approach leverages all sensor measurements to improve the AUVs global state estimation and then allows to either use the a priori knowledge on the pose of the docking station or the relative docking station pose estimation for finishing the docking process.

Further, we introduce an easily extendable docking algorithm which allows usage of additional perception measures for confirmation of successful docking. We propose a state machine implementation (Fig. 4) that covers the complete progress of homing, approaching and finishing docking. The implementation aims to be easily extendable for future robustness check extensions.

To validate our results, we present a series of real-world experiments in our test basin. For this we used our AUV *Deepleng* [5] (Fig. 1) and a prototype docking station (Fig. 6) which is equipped with a ring of six Apriltag [6] markers for mid-range ($5 - 10$ m) localization and a smaller marker in the center of the docking cone for near-range (< 5 m) localization and docking progress validation. For long-range positioning an Evologics R 18/34 USBL Modem was used.

A. Related Work

Several works have attempted autonomous docking for underwater vehicles using various designs and techniques in the past years. A docking system that relies on USBL aiding was developed in [7]. The AUV was equipped with a reverse USBL in its nose allowing bearing and range measurement with respect to its beacon that is placed at a fixed location in the area

This work was funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK), under the project the EurEx-LUNa (grant no. 50NA2002).

of survey. The docking station is conically shaped with 2 m entrance diameter, compared to a 54 cm diameter of the AUV. A docking method for a hovering type AUV was developed in a previous work [8], which relied on visual aiding from a number markers mounted onto the docking station. A docking method based on acoustic and visual positioning for a hovering type AUV was developed in [9]. This system is comprised of an approach mode relying on acoustic position, ground speed, attitude and depth. As the AUV is in visual range to the station, the docking is switched to visual aiding using LED markers. In [10], two complementary methods were developed to achieve docking, where the first consists of a range-only localization providing feedback for the AUV approach. The second method is based on active visual markers providing high accuracy measurements in close range. A stereo-vision based docking method with active markers was developed in [11]. To test the robustness of this method, a test was conducted in a pool while simulating a switch from day to night. Further readings on AUV docking can be found in [12]–[15].

II. Localization

For the localization of the AUV an Unscented Kalman Filter (UKF) based Inertial Navigation System (INS) was utilized [4]. To improve the docking robustness and reliability, the filter was extended by integrating delayed USBL measurements as well as visual feature measurements. The USBL position measurements ensure successful homing of the vehicle after a long mission duration, that typically suffers from a growing drift of the position estimation during the dead reckoning navigation. USBL measurements need special treatment because they will arrive delayed as they are acoustic measurements that depend on the velocity of sound in the water medium. In distances of multiple kilometers between modem and AUV, this can lead to delays of multiple seconds and thus differences of multiple meters between measured position and actual position.

To further robustify the docking process, especially in terms of the alignment to the docking cone, artificial Apriltag markers are used. In particular their corner features are used as measurements to additionally aid the pose estimation in the proximity of the docking cone. We extended the UKF filter based INS described in [4] by a measurement model for marker based visual features and by the integration of delayed USBL measurements.

A. Visual Feature and Apriltag Marker Integration

In order to aid the pose estimation using the corner features of an Apriltag, the marker poses in the navigation frame have to be known. For this, their absolute poses are initially determined using a calibration process. Since each marker pose will be associated with an uncertainty after the calibration process we temporally augment the state of the filter by the calibrated marker pose in the navigation frame, allowing to reflect its pose and respective uncertainty in the measurement update for each corner feature.

The filter state is augmented by $\mathbf{p}_m^n \in \mathbb{R}^3$ and $\boldsymbol{\phi}_m^n \in SO(3)$, the position and orientation of an Apriltag marker in the navigation frame. The augmentation is removed and the original size of the filter state is restored after the integration of the corner features for this marker. As described in [16] the filter operates on a locally mapped neighborhood of the manifold $SO(3)$ in $\mathbb{R}^3$ and the manifold S^2 in $\mathbb{R}^2$.

The measurement model for the marker based corner features is defined as:

$$\begin{aligned}\hat{\mathbf{z}}_m(t) &= \frac{\mathbf{C}_b^c \mathbf{C}_{n,t}^b \mathbf{C}_{m,t}^n \mathbf{p}_i^m}{|\mathbf{C}_b^c \mathbf{C}_{n,t}^b \mathbf{C}_{m,t}^n \mathbf{p}_i^m|} \boxplus \nu_m \\ &= \frac{\mathbf{C}_{m,t}^c \mathbf{p}_i^m}{|\mathbf{C}_{m,t}^c \mathbf{p}_i^m|} \boxplus \nu_m\end{aligned} \tag{1}$$

where $\mathbf{p}_i^m$ is the position of the marker corner point with index i in the marker frame, $\mathbf{C}_{m,t}^n$ is the coordinate transformation from the marker to the navigation frame (given by the augmented filter state), $\mathbf{C}_{n,t}^b$ is the coordinate transformation from the navigation to the body frame at time t and $\mathbf{C}_b^c$ is the static coordinate transformation from the body to the camera frame. $\nu_m \sim \mathcal{N}(0, \Sigma_m)$ is a zero-mean normal distributed random variable, where Σ_m is the uncertainty of the feature measurement. $\hat{\mathbf{z}}_m(t) \in S^2$, the surface of the unit sphere. The $\boxplus$ operator in equation (1) is a manifold based addition, as defined in [16].

Each corner point of the Apriltag marker is integrated using the same measurement model, while the marker corner point $\mathbf{p}_i^m$ is selected depending on the corner index i of the feature. The corner points are defined by the type and size of the utilized marker. In case of the Apriltag marker, 4 corner points are integrated using the measurement model in equation (1). The corner feature measurements itself are undistorted and normalized from the image domain and then also represented in S^2.

B. USBL Measurement Integration

The second extension to our localization stack was the integration of USBL measurements. As measurements we use 2-dimensional position measurements. USBL measurements have to be treated specially because it uses acoustic signals for positioning, meaning a delay of the measurement on signal arrival at the vehicle. For integrating them into our UKF we chose a similar approach to the proposed method *solution E* from Mandt et al. [17]. During runtime, we buffer all estimated states until a configured duration in the past. This buffer duration can be derived from the maximum range of the USBL and speed of sound. On USBL signal arrival, the measurement timestamp is matched to a past state of the UKF. Then the positional difference between the delayed state and the current state is added to the measurement to transform it into a real-time measurement:

$$\begin{aligned}e_p(t, t-k) &= p(t) - p(t-k) \\ z_m(t) &= z_m(t-k) + e_p\end{aligned} \tag{2}$$

In this case e_p denotes the positional error of the delayed state and current state position $p(t-k), p(t) \in \mathbb{R}^2$ at timestamp $t-k$, with k being the delay between the current time and

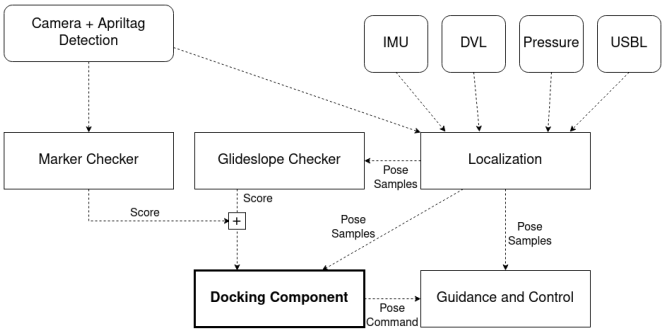

Fig. 2. Simple representation of the relevant software components within the software architecture for the docking algorithm.

the measurement time. The artificial real-time measurement $z_m(t)$ is then calculated by simply adding this positional error e_p to the delayed measurement $z_m(t-k)$. This can be done under the assumption that the error of the pose estimation is not growing substantially in the time between the two states [17]. This assumption is true in the case of the AUV Deepleng, as it is using high-quality dead reckoning sensors. Then the transformed USBL measurement is simply integrated as a 2D measurement into the UKF. The measurement model $\hat{z}_m(t)$ is the current 2D position estimate $p(t)$:

$$\hat{z}_m(t) = p(t) \tag{3}$$

III. Docking Algorithm

To complement the abilities of the AUV to precisely localize itself and return back safely into the docking station, a docking algorithm was implemented. It is used within a Behaviortree-based [18] mission execution framework which starts the docking process as well as handling potential error states.

It is crucial for the docking algorithm to be able to handle potential failure cases that might occur during the docking process and take into account additional observation measures to the reliance to the localization component. With this it is able to detect if it misses the docking cone or gets stuck during the approach.

In figure 2, a simple representation of our AUV software architecture and its relevant components for the docking algorithm are visualized. The localization component was already introduced in the previous section. The details of the docking algorithm, as well as the used guidance and control component are introduced next.

A. Guidance and Control

The guidance system is comprised of a simple point-to-point method based on line-of-sight. Each waypoint command is a 6D pose $(x_d, y_d, z_d, \phi_d, \theta_d, \psi_d)$ that is given as input to the controller, where x_d, y_d and z_d represent the desired position, and ϕ_d, θ_d and ψ_d the corresponding Euler angle orientation. Given the current pose and velocity of the AUV, a control method based on a Proportional-Integral-Derivative (PID) with feedback linearization is used to control the vehicle to the desired position following a line-of-sight guidance scheme as in [19].

B. Docking Algorithm State Machine

The docking algorithm is implemented as a state machine as shown in figure 4. For better reproducibility the algorithm is also summarized in algorithm 1. It consists of 5 states which can be grouped into:

1) Preparation Stage
   - *Wait for Docking Command*
   - *Starting Docking*
2) Docking Stage
   - *Docking in Progress*
   - *Finishing Docking*
   - *Docking Completed*

The algorithm differentiates between the *Docking in Progress* and *Finishing Docking* states, as this allows to alter corresponding docking condition checks and behavior of the vehicle during the approach. In detail this means, that if the vehicle is already inside of the docking cone but not docked yet (as shown in figure 1), only minimal visual monitoring is needed and the vehicle can simply thrust forwards until it is docked. Additionally there is a *Docking Error State* which as of now is a general state that the state machine transitions to on any error occurring during the docking process. In the current iteration of the algorithm there is no difference between different error states.

In the preparation stage and after receiving the docking command, the algorithm initializes necessary parameters. This includes reading of the docking station pose in the local navigation frame as well as parameters for different checks during the docking process. Pose error threshold parameters are used to determine when a waypoint in the trajectory is reached as well as additional parameters for when the vehicle considers itself to transition to the *Finishing Docking* and *Docking Completed* states. Additionally error handling and detection parameters are defined and configured such as detection of the AUV getting stuck in the docking cone. After configuration, the docking trajectory is planned. In the current iteration of the algorithm a simple docking trajectory is planned. For this the vehicle first aligns its orientation, followed by aligning its depth, as well as its horizontal position (y-axis) relative to the docking cone. After alignment the vehicle will move forwards in a previously configured step size, e.g. $0.3m$. Then, the state machine will transition to the *Docking in Progress* state.

During docking, the following sequence is executed:

1) Check, if vehicle is inside of the cone
2) Calculate and check the docking conditions score
3) Check, if current trajectory waypoint was reached (and proceed to next waypoint)
4) Output current waypoint pose command to the controller

In the first step, the algorithm checks if the vehicle is already inside of the cone. In the current iteration of the algorithm this means that the current distance to the docking goal pose

Algorithm 1 Docking Algorithm

Require: Start Docking Command
  state ← *Start Docking*
  Initialize Parameters
  Plan Docking Trajectory
  state ← *Docking in Progress*
  while state == *Docking in Progress* **do**
    Read newest pose sample
    if Docking Conditions score < threshold **then**
      state ← *Docking Error State*
      return
    end if
    if nose inside of cone **then**
      state ← *Finishing Docking*
    end if
    if current waypoint reached **then**
      Proceed to next waypoint in trajectory
    end if
    Write pose command to guidance and control component
  end while
  while state == *Finishing Docking* **do**
    Read newest pose sample
    if Finishing Docking Conditions score < threshold **then**
      state ← *Docking Error State*
      return
    end if
    if Docking Completed **then**
      state ← *Docking Completed*
    end if
    if current waypoint reached **then**
      Proceed to next waypoint in trajectory
    end if

    Write pose command to guidance and control component
  end while

is checked. As the dimensions as well as the absolute pose of the docking station is known, the checking of the distance proofed to be a well-working condition under the assumption that the vehicle localization is accurate enough and the distance parameters are configured properly under consideration of potential error margins. If this check succeeds, the state machine transitions to the *Finishing Docking* state.

After that, the algorithm calculates and checks the docking condition score by summing up the retrieved scores of the corresponding checker components. The implemented docking condition score system is one of the key features of this implementation and is explained in more detail in the following subsection. In the current implementation only two docking conditions are used. First it is checked if the apriltag marker that is placed in the center of the docking cone can be seen by the front camera of the vehicle. The docking condition

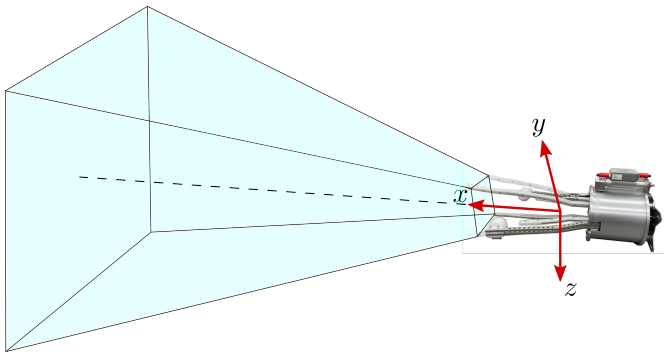

Fig. 3. 3D representation of the docking cone.

score is 1.0, if the vehicle sees the marker in the currently available image frame. If the marker can not be seen, the score degrades by a pre-configured value with each image frame that does not include the apriltag marker. As second condition it is checked whether the vehicle is within a defined glideslope. The glideslope is a pre-configured cone-shaped virtual body as shown in figure 3, which is measured by knowing the dimensions and angles of the docking cone. The docking condition score is 1.0 (and thus holds) if the nose of the vehicle is inside of this virtual body. If the sum of these scores is below the configured docking threshold, the docking process is aborted and the state machine transitions to the *Docking Error State*. In the last two steps the algorithm checks whether the current waypoint of the trajectory was reached and outputs the current pose command to the controller.

The *Finishing Docking* state is structured in a very similar way to its previous state. The core differences are that it allows to utilize different docking condition states and it checks whether docking is completed. In the currently used implementation only the apriltag marker check condition is used. The docking process is considered complete (and thus a transition to the *Docking Completed* state) if the vehicle is within a configured distance threshold to the goal pose and the vehicle is not moving anymore.

C. Docking Condition Score Feature

The key feature of the proposed docking algorithm is the usage of a score-based docking condition system. In detail this means, that docking conditions checks that are used for ensuring success of the docking process are modeled as probability-based components. This means that each docking condition, e.g. the visibility of the docking marker in the center of the docking station, outputs a so-called score value in range $[0, 1]$. The reason for this is the possibility to not only support boolean-typed docking conditions (which translate to scores of 0.0 and 1.0 respectively) but also probability-based docking conditions such as outputs of classification neural networks. The docking scores can then be used by the docking algorithm to determine either a total score (by summing up) or a docking probability (normalization of the total score). Whether the vehicle should proceed docking or abort can then be determined by either tuning and setting a configuration

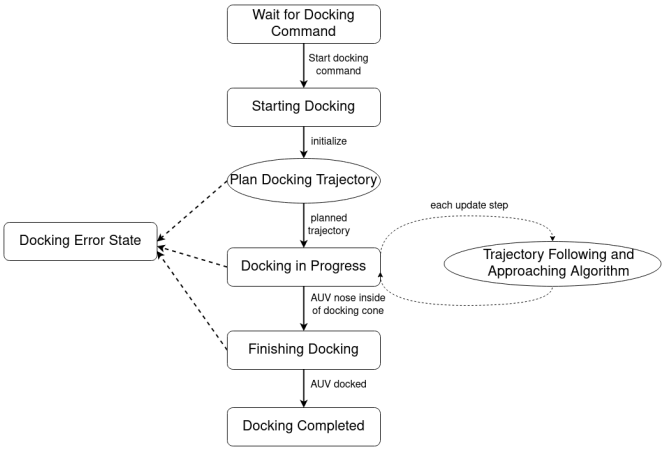

Fig. 4. State Machine representation of the proposed docking algorithm implementation.

TABLE I
COMPARISON BETWEEN LOCALIZATION SOLUTIONS SHOWING THE MEAN ERROR, STANDARD DEVIATION AND MAX ERROR

Version	Mean Error	Standard Deviation	Max Error
Dead Reckoning	5.2759	4.6931	16.04568
Dead Reckoning + Markers	3.1603	2.1335	8.7491
Dead Reckoning + USBL	0.6487	0.9040	5.6105
Full Localization	0.6760	0.9180	5.8587

parameter or also give potential to apply a behavior learning based method.

In context of the docking algorithm these docking conditions are handled as dynamic inputs and can be differentiated in *docking* and *finishing docking* conditions. In the current iteration of implementation the scores are simply summed up. Therefore it is necessary to know the maximum possible score to configure a meaningful threshold. If no threshold was set, it is assumed that all conditions must have the full score (1.0) to continue docking.

IV. Localization Evaluation

For evaluation of the newly implemented measurement integration functionalities an under-ice dataset [20] was used. The dataset [21] was recorded on a recent field trip with the AUV Deepleng to Abisko, Sweden. To show the improvements of state estimation accuracy we compare the pure dead-reckoning mode consisting of IMU, DVL and pressure sensor measurements with integration of only markers and USBL respectively and the full localization integrating both marker measurements and USBL 2D position measurements. For evaluating the position performance of the localization component, we measure mean error, standard deviation and max error compared to the USBL measurements as only pseudo-ground-truth available. The positional errors are given in meters. The results are presented in table I. The corresponding visualization is presented in figure 5.

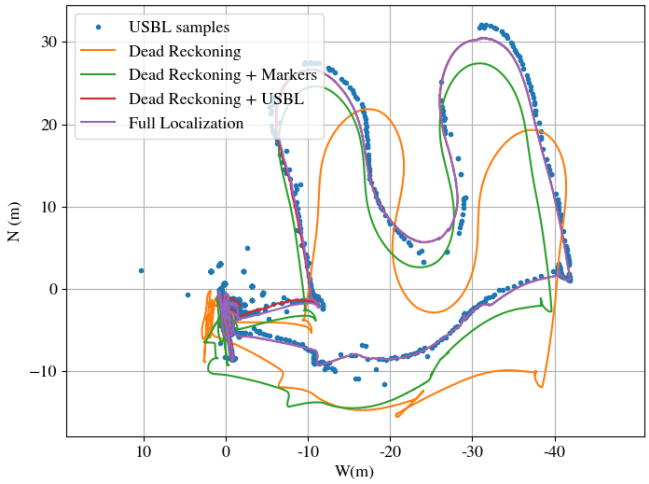

Fig. 5. Comparison of different versions of the pose estimation on Deepleng Abisko Dataset. [20], [21]

As both the values in table I and the visualization in figure 5 show, both versions with integrated USBL outperform the other two versions heavily across all metrics. The version with only the USBL is even slightly better than the one with the marker integration. These results were expected, especially if the same USBL measurements, that are also integrated into the UKF are used as pseudo-ground-truth measurements to evaluate their performance. The difference between the USBL only and the Full Localization versions are observed because of measurement errors in the absolute poses and transformations of the apriltag markers to the USBL base station.

Although it was not possible to evaluate the orientation error of the newly implemented measurement functions with the dataset. Though preliminary results from tests in our basin suggest improvements at least for rather short-term missions. Its impact in long-term missions remain to be evaluated in future work. With the availability and visibility of Apriltag markers after homing, it is even possible to re-initialize the pose estimation to further improve docking success probability.

V. Docking Experiments and Evaluation

To evaluate the performance and robustness specifically, a number of experiments were conducted. The goal of the experiments were to show the robustness of the algorithm and implemented software architecture in a controlled environment. For this a multitude of docking trials with traveling to different waypoints between the individual docking attempts were executed. Following the experiment setup and configurations are described in more detail as well as the different experiments and their outcome.

A. Experiment Setup

The conducted docking experiments focus on evaluating the robustness of the proposed docking algorithm. For this

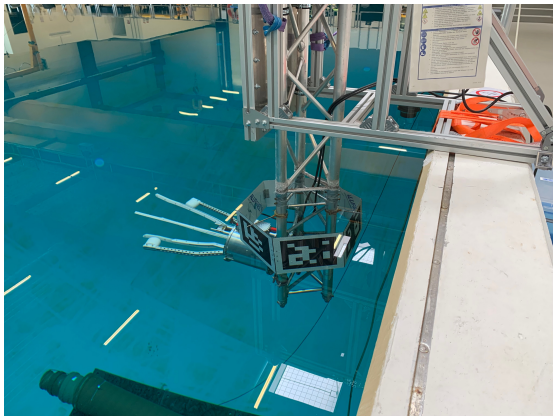

Fig. 6. Docking station mounted in our test basin.

multiple series of waypoint and docking actions were executed. One series consists of multiple sequences in which the vehicle would traverse to a pre-defined waypoint in our basin and then return to the docking station and dock. These sequences are repeated between 2-10 times per series and there were 10 sequences per series, with 5 different waypoints and 2 different target depths. These waypoints were chosen to simulate different approaching angles to the docking station and are shown in figure 7. Additionally there were two types of experiments, one using marker detection integration into the pose estimation as well as for visual docking assurance and the other one was complete artificial marker occlusion meaning that the pose estimator was using only IMU, DVL and pressure samples for dead reckoning and the docking algorithm was blindly docking.

The docking station is mounted at the edge of our test basin as shown in figure 6. We opted to not use USBL position samples for localization in these experiments due to the lack of a reliable outlier detection for our USBL samples.

B. Docking without Marker Occlusions

In the first series of experiments the docking algorithm was evaluated with integration of marker detection into the localization component and the docking algorithm. Across all of the (logged) experiments a total of 125 docking sequences were executed, 121 of them successful, yielding a success rate of 96.8%. In figure 7 we show such a series of docking sequences, consisting of 2x5 waypoints and a total of 20 docking sequences.

Two of the 4 unsuccessful attempts were due to unrecoverable (and unfortunately not retracable) execution errors of the used middleware framework *Robot Construction Kit (ROCK)* [22]. One failed docking attempt was due to the vehicle getting stuck in the arms of the docking cone and not detecting it properly due to misconfigured parameters concerning the stuck detection. And the last failed attempt was due to a DVL outlier during returnal from a waypoint to the docking station without any apriltag markers in sight. Thus the localization component was not able to recover this jump of the position estimation. As this error occurred during one of the very first series of

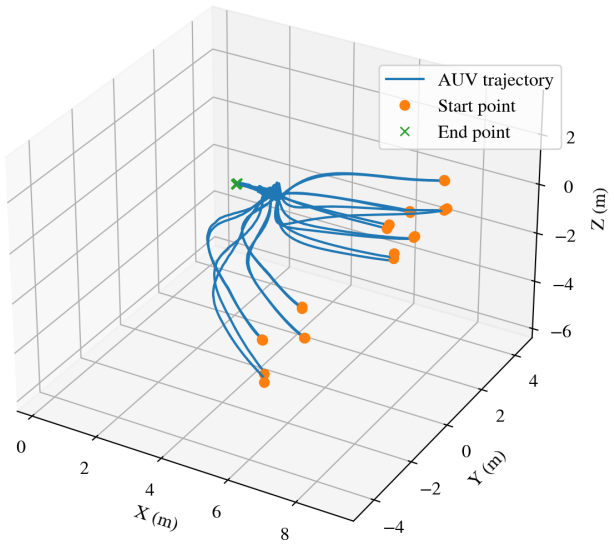

Fig. 7. Standard 2x10 Waypoint docking experiment series without marker occlusion

experiments, this error was resolved by adding a simple DVL measurement outlier rejection which rejects any measurement that would mean a physically impossible increase of the vehicle velocity using information on the hydrodynamic model of the AUV.

Besides of the logged experiments, many more successful docking sequences were executed, possibly pushing the success rate even higher than the logged 96.8%. With fixing of the two docking and localization software related errors, we showed that the proposed docking framework is highly robust on availability of apriltag marker detections and under consideration of different failure cases.

C. Docking with Marker Occlusions

To further show the robustness and reliability of the proposed docking algorithm an artificial marker occlusion component was added to the experiments. In detail this means, that all marker detections are hold back and neither integrated into the localization component nor used for visual confirmation for the docking algorithm. For this, the docking condition score parameters had to be adapted to solely rely on the glideslope score. The goal of these experiments were to evaluate the behavior of the docking algorithm even under increasing positional and rotational drift of the pose estimation. One series was executed until the occurrence of an error which needed intervention of an operator.

In total, 3 series of experiments with artificial marker occlusion were executed during the evaluation process. Again all series contained the 10 different waypoints, with each being traversed up to 10 times. In the first series, the vehicle was able to dock successfully for 25 consecutive times. At the 26th attempt the vehicle got stuck in the docking cone,

was not able to detect this error and therefore the series had to be aborted. In the second series the vehicle was able to successfully dock for 14 consecutive times. In the last attempt the main thruster threw an unrecoverable exception (unrecoverable without completely restarting the system) thus the experiment had to be aborted. In the last series the vehicle was able to dock for 34 consecutive times, failing at the 35th attempt. The reason for this was, again, a poorly configured parameter for the stuck detection as well as a drift in depth estimation due to changes in the atmospheric pressure. In total this yields 73 successful attempts out of 76 total attempts, meaning a success rate of approximately 96%.

VI. Discussion

In this paper we proposed an extension to our localization framework by integrating visual features from apriltag marker measurements and USBL 2D position measurements. Our results showed clearly their influence on the localization performance and improved accuracy by reducing the mean error from 5.2759m to 0.65m and max error from 16m to 5.6m compared to only using the dead reckoning approach. During the docking experiments in our basin, only the marker integration was used, performing really well and not causing any unsuccessful docking attempts. As of now, we still have to evaluate the performance of the localization using the USBL in real-life experiments, especially the interaction between localization and control. While replaying the collected dataset, it was observed that the pose estimates inhibited a jittering effect on retrieval of USBL data, jumping by multiple centimeters, which could lead to unstable behavior of the vehicle during traversal. Additionally it remains to be tested how the usage of the markers influences the orientation estimation during long-term missions as there is no ground truth data available for orientation within the dataset. Though first tests in our basin seem promising in that regard.

In the conducted experiments, we showed that the proposed docking algorithm is highly robust in experiments in our basin. Even on occlusion of markers and on slight positional and orientation drift, the docking algorithm is still able to dock in most cases. Most failures were due to poorly tuned configuration parameters or unpredicted system errors independent from the docking algorithm or localization itself. Still the docking algorithm remains to be tested after traversal of longer range trajectories which can lead to additional drift than easily doable during the experiments within the scope of this paper.

VII. Outlook

In the next steps, the proposed methods and implementations have to be tested in real world long-term missions. For this, a number of experiments are planned in a nearby lake of Bremen as well as another field trip to Abisko to test the AUV and the implemented methods in the under-ice scenario. Further we want to investigate whether the correction of the orientation estimate on returnal to the docking station is accurate enough for docking the AUV after a long-term mission or if the pose estimator has to be re-initialized after returning to the station and visibility of the visual markers.

Additionally, using apriltag markers for visual confirmations works perfectly in clear and perfect conditions under water, though is highly unreliable in darker or turbid water. For this, deep-learning based approaches could help detect the complete docking station as it is a static and well-known object which is more easily observable than potentially small markers that are attached to the docking station.

References

[1] K. Konstantinidis, C. Flores Martinez, M. Hildebrandt, and R. Förstner, "Towards an astrobiological vision for the outer solar system: The europa and enceladus explorer mission designs," in *Workshop on the Habitability of Icy Worlds*, vol. 1774, 2014, p. 4043.

[2] C. Waldmann, J.-P. de Vera, B. Dachwald, H. Strasdeit, F. Sohl, H. Hanff, J. Kowalski, D. Heinen, S. Macht, U. Bestmann *et al.*, "Search for life in ice-covered oceans and lakes beyond earth," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE, 2018, pp. 1–7.

[3] P. W. Kimball, E. B. Clark, M. Scully, K. Richmond, C. Flesher, L. E. Lindzey, J. Harman, K. Huffstutler, J. Lawrence, S. Lelievre *et al.*, "The artemis under-ice auv docking system," *Journal of field robotics*, vol. 35, no. 2, pp. 299–308, 2018.

[4] S. Arnold and L. Medagoda, "Robust model-aided inertial localization for autonomous underwater vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[5] M. Hildebrandt, S. Arnold, P. Kloss, B. Wehbe, and M. Zipper, "From epi-to bathypelagic: Transformation of a compact auv system for long-term deployments," in *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*. IEEE, 2020, pp. 1–6.

[6] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[7] R. S. McEwen, B. W. Hobson, L. McBride, and J. G. Bellingham, "Docking control system for a 54-cm-diameter (21-in) auv," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 550–562, 2008.

[8] M. Wirtz, M. Hildebrandt, and C. Gaudig, "Design and test of a robust docking system for hovering auvs," in *2012 Oceans*. IEEE, 2012, pp. 1–6.

[9] T. Maki, R. Shiroku, Y. Sato, T. Matsuda, T. Sakamaki, and T. Ura, "Docking method for hovering type auvs by acoustic and visual positioning," in *2013 IEEE international underwater technology symposium (UT)*. IEEE, 2013, pp. 1–6.

[10] N. Palomeras, G. Vallicrosa, A. Mallios, J. Bosch, E. Vidal, N. Hurtos, M. Carreras, and P. Ridao, "Auv homing and docking for remote operations," *Ocean Engineering*, vol. 154, pp. 106–120, 2018.

[11] K. N. Lwin, N. Mukada, M. Myint, D. Yamada, M. Minami, T. Matsuno, K. Saitou, and W. Godou, "Docking at pool and sea by using active marker in turbid and day/night environment," *Artificial Life and Robotics*, vol. 23, no. 3, pp. 409–419, 2018.

[12] J. Y. Park, B. huan Jun, P. mook Lee, and J. Oh, "Experiments on vision guided docking of an autonomous underwater vehicle using one camera," *Ocean Engineering*, vol. 36, no. 1, pp. 48–61, 2009.

[13] S. Fan, C. Liu, B. Li, Y. Xu, and W. Xu, "Auv docking based on usbl navigation and vision guidance," *Journal of Marine Science and Technology*, vol. 24, pp. 673–685, 2019.

[14] T. Wang, Q. Zhao, and C. Yang, "Visual navigation and docking for a planar type auv docking and charging system," *Ocean Engineering*, vol. 224, p. 108744, 2021.

[15] A. M. Yazdani, K. Sammut, O. Yakimenko, and A. Lammas, "A survey of underwater docking guidance systems," *Robotics and Autonomous systems*, vol. 124, p. 103382, 2020.

[16] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.

[17] M. Mandt, K. Gade, and B. Jalving, "Integrating DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV Magne Mandt * , Kenneth Gade ** , Bjørn Jalving *** Forsvarets Forskningsinstitutt (Norwegian Defence Research Establishment) P.o. Box 25, No-2027 Kjeller, Norway," *Proceedings of the 8th Saint Petersburg International Conference on Integrated Navigation Systems, Saint Petersburg, Russia*, no. 2027, 2001.

[18] Auryn Robotics, "BehaviorTree.CPP." [Online]. Available: https://www.behaviortree.dev/

[19] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[20] M. Hildebrandt, T. Creutz, B. Wehbe, M. Wirtz, and M. Zipper, "Under-Ice Field tests with an AUV in Abisko/Torneträsk," in *OCEANS 2022-Hampton Roads*. IEEE, 2022.

[21] Hildebrandt, Marc and Wehbe, Bilal and Wirtz, Marius and Creutz, Tom and Zipper, Michael, "Eurex-LUNa Abisko Trials Lawnmower Vectoring." [Online]. Available: https://zenodo.org/record/7035132

[22] DFKI GmbH Robotics Innovation Center, "The Robot Construction Kit." [Online]. Available: https://www.rock-robotics.org/