Docking MOLA: Apriltag detection, data fusion, and control of a 6-DOF AUV

Juri Hemmi[1], Mentors: Giancarlo Troni[2], and Sebastián Rodríguez-Martínez[3]

***Abstract*— This report describes the development of a perception and localization system for autonomous underwater vehicle (AUV) docking, implemented on the MOLA AUV platform. The system uses Apriltag fiducial markers for visual pose estimation, enabling reliable localization relative to both fixed and dock-mounted tags. Five Python modules were developed and integrated via Lightweight Communications and Marshalling (LCM): Apriltag Detection, Delayed Extended Kalman Filter (EKF), Dock EKF, Docking Controller, and a Diagnostic Plotter. The delayed EKF provides real-time state estimates suitable for use in control. Experiments demonstrated stable vehicle control and centimeter-scale localization accuracy of the dock when the vehicle approached the dock. Although full mechanical docking was not yet achieved, the results validate the perception and localization framework and establish a strong foundation for autonomous underwater docking.**

Fig. 1: *This figure shows the MOLA AUV (the vehicle used for testing) sitting next to the test tank.*

I. INTRODUCTION

Autonomous underwater vehicles (AUVs) can greatly benefit from reliable docking systems. Docking enables efficient deployment and retrieval, persistent underwater operation, and flexible launch from larger vehicles. Firstly, docking would allow for easy deployment and retrieval of the vehicle from a ship. For example, the dock can be lowered into the water and the vehicle can autonomously detach, significantly improving operational efficiency. Lowering the dock to depth could also simplify deep-sea operations and mean that there is less transport required of the vehicle. This reduces energy demands on the AUV, potentially allowing for smaller battery systems as well as shorter transport time. Secondly, docking would allow for persistent sub-sea monitoring. A stationary dock placed on the seafloor could enable frequent missions, allowing the vehicle to return periodically to charge and transfer data. This would support longer-term inspection and mapping studies. Finally, A dock mounted on a larger vessel would allow the smaller vehicle to be transported to a region of interest, from which it can then detach to run local missions. This could be useful if a large vehicle can spend the time and energy traveling to the site and then the smaller more nimble vehicle/s can detach and gather data.

Given these benefits, docking systems are being developed for many types of vehicles. For instance, a docking system has been developed for MBARIs Long Range AUVs (LRAUV). This system consists of a vertical niobium rod with a light. The cylindrical, 3-DOF vehicle uses the light as a guide to drive into the rod. The vehicle can then charge and transfer data through its connection with this rod. This works well, but such a system cannot be used to move the vehicle as the coupling is not robust enough. Another docking system for MBARIs LRAUVs is described in [1]. This system uses an Ultra Short Base Line (USBL) acoustic sensor to help drive the vehicle into a tapered tube which holds the vehicle. Other docking systems discussed in the literature include a helipad style dock described in [2] in which the vehicle drops down into a depression using a guiding light and another, summarized in [3], which uses fiducial markers to help position a vehicle to drop down onto a platform.

[1]University of New South Wales, Sydney, NSW, Australia

[2,3]Monterey Bay Aquarium Research Institute, Moss Landing, CA 95039-9644 USA (e-mail: gtroni@mbari.org; srodriguez@mbari.org)

6-DOF vehicles are quite different from the standard cylindrical vehicles meaning that docking solutions that work for cylindrical AUVs do not work well for more box-like 6-DOF vehicles. However, the docking systems that we can make for 6-DOF vehicles are much more versatile than the other docking solutions. With accurate pose estimation the docking maneuver can be done at low speeds even with cross current and using a more compact dock. As stated in [3], accurate pose estimation is usually achieved through long-range acoustic sensors and close-range optical navigation.

The vehicle that was used for development is called the MOLA AUV shown in Figure 1. It is a suitcase sized 6-DOF vehicle with a tether and can be run via remote control as well as autonomously. Once docking and the Autonomous functions are further developed the goal is to remove the MOLA AUV's tether and have it run missions fully autonomously.

The objective of this work is to develop and validate a perception and localization pipeline to be used for autonomous docking of the MOLA AUV. This report focuses on the perception and localization components of this pipeline. In particular, we use Apriltag fiducial markers to make relative pose measurements using the MOLA AUV's camera. Apriltags offer unique IDs allowing the vehicle to recognize these landmarks and localize itself relative to these fixed references in the environment or on the dock. Compared to more general methods such as SLAM, Apriltags can offer higher certainty in pose estimation and can be used as a global pose measurement to anchor the estimated pose to a preset map.

This report describes the methods used in Section II. Section III describes the results as the MOLA AUVs current docking capability. Section IV discusses the methods used in developing the modules. Section V presents some ideas for where to further develop and improve what has been built. Finally, Section VI concludes the report.

II. METHODS

All software was developed in Python. The overall system was implemented as a set of five modules that communicate via Lightweight Communications and Marshalling (LCM). The system was tested with recorded data and then in real-time using the MOLA AUV.

A. Modules

The Apriltag Detection module receives a camera stream and outputs relative pose measurements between the vehicle and the observed Apriltags. These measurements provide estimates of the vehicle's pose within the Apriltag coordinate frames.

The Delayed EKF module fuses all measurements related to the vehicles pose. This module produces a real-time estimate of the vehicle's pose, compensating for the latency introduced by visual processing.

The Dock EKF module uses the vehicle's estimated pose together with the Apriltag measurements of the dock to provide a filtered estimate of the dock's pose.

The Docking Controller module takes both the vehicle and dock pose estimates and generates motion commands to guide the vehicle into the dock.

Finally, the Diagnostic Plotter receives outputs from all modules and generates real-time 3D plots of the environment, vehicle, and dock. This tool assists in debugging and in understanding the system's perception and localization performance.

Figure 2 shows the flow of information between these modules.

B. Equipment

Testing was conducted using the MOLA AUV shown in Figure 1, a suitcase-sized 6-DOF vehicle equipped with a fiber-optic tether. All computation was run on surface laptops, which transmitted pose and position commands to the vehicle's onboard motion controller.

Experiments took place in a $10m \times 10m \times 20m$ saltwater test tank equipped with Apriltags mounted on one wall. A prototype dock consisting of a rigid frame with Apriltags on its top and front surfaces was suspended in the tank for testing (Figure 3).

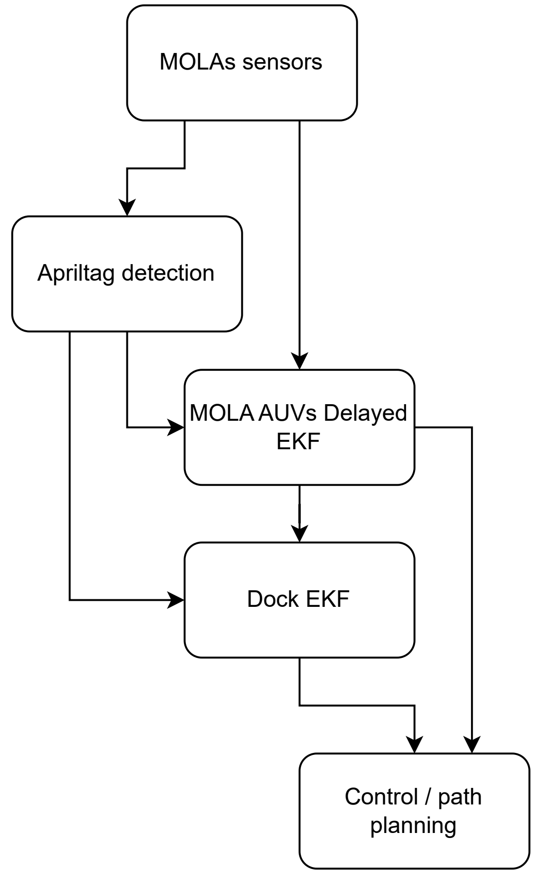

Fig. 2: *This figure shows the relationships between the various modules.*

III. RESULTS

The performance of the perception and localization pipeline was verified through real-time, in-the-loop testing on the MOLA AUV. All modules were validated by operating the vehicle and inspecting the diagnostic plots to confirm that the estimated poses were consistent with ground truth observations. During these tests, vehicle control remained stable when using the estimated poses generated by the system. The vehicle successfully perceived, localized, and approached the dock, achieving centimeter-scale positional accuracy when close to the dock.

At the time of testing, a force controller for physical coupling with the dock had not yet been implemented, and therefore full mechanical docking was not achieved. Further

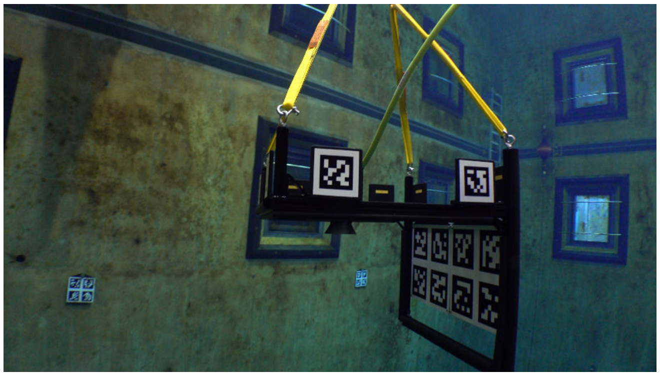

Fig. 3: *This figure shows the prototype dock hanging in the test tank. Note the Apriltags on the dock and the test tank wall behind.*

development of the controller is required to enable this functionality.

IV. DISCUSSION

This section outlines the architecture and implementation of the five developed modules. Section IV-A describes the Apriltag Detection module. Section IV-B outlines the structure of the delayed EKF. Section IV-C details how the dock pose is estimated and filtered. Section IV-D summarizes the docking controller, and Section IV-E presents the diagnostic visualization tools.

A. Apriltag Detection Module

This module ingests images from a video stream either by direct reading, monitoring a directory for new files, or receiving LCM messages containing image paths. The Apriltag-3 algorithm [4] (via its Python bindings [5]) is then used to estimate the relative pose between the camera and each tag detected in the frame.

Detections are categorized into two sets: (i) tags fixed in the global coordinate frame and (ii) tags attached to the dock. Each set is processed separately to compute the vehicle pose in both coordinate frames. Processing time per frame is in the order of 0.5s.

The measurement processing pipeline consists of transforming Apriltag measurements and computing covariances (Section IV-A.1), filtering outliers (Section IV-A.2), and fusing multiple detections to obtain a single pose estimate (Section IV-A.3).

1) Obtaining Covariance and Transforming Apriltag measurements: The Apriltag 3 algorithm outputs the relative pose between the camera and a tag. We require the pose of the vehicle with respect to the relative coordinate frame. We also need to obtain a mean and covariance of the vehicle's pose for each measurement.

Transforming the mean of the measurement into the vehicles pose was done in a standard way using the known pose of each tag.

Unfortunately, the measurements that the Apriltag 3 detection algorithm output did not contain any uncertainty information. To solve this we create positional and rotational covariances for the relative poses starting with two tunable parameters representing the general position and rotation variance. We then expand these into 3×3 diagonal matrices and scale them by the measured distance so that the further away the tag is, the greater the uncertainty.

This gives:

$$\Sigma_{\text{posBase}} = I_3 \times \sigma_{\text{pos}}^2 \times d, \quad (1)$$

$$\Sigma_{\text{rot}} = I_3 \times \sigma_{\text{rot}}^2 \times d, \quad (2)$$

where d is the distance to the tag, and σ_{pos}^2 and σ_{rot}^2 are the tunable variance parameters.

The rotational uncertainty of the measured value Σ_{rot} is simply the same as the rotational uncertainty of the transformed vehicles pose. The final positional covariance of the transformed vehicle pose has two contributing factors. Firstly the direct propagation of positional variance of the measured tag, and secondly, the indirect contribution from rotational uncertainty of the measured tag.

Contributions of the measured positional variance: We can calculate the amount contributed to the positional covariance by the positional uncertainty in the original measurement in the standard way. I.e

$$\Sigma_{\text{posPos}} = R_{rot} \times \Sigma_{\text{posBase}} \times R_{rot}^T \quad (3)$$

Where R_{rot} is the rotation matrix multiplied by the measured position vector in its transformation.

Contributions of the measured rotational variance: Angular uncertainty in the orientation of the detected tag produces translational uncertainty in the derived camera position. We calculate the contribution by constructing a flat covariance matrix that lies perpendicular to the camera-tag direction. We then normalize this matrix, and multiply it by the rotational uncertainty of the measurement $\sigma_{\text{rot}}^2 \times d$ to get the resulting Σ_{posRot}.

Finally the position covariance is calculated as

$$\Sigma_{\text{pos}} = \Sigma_{\text{posPos}} + \Sigma_{\text{posRot}} \quad (4)$$

2) Filtering Outlying Measurements: When measurements of all Apriltags in a frame were plotted, we saw that there were occasionally significant outliers. To filter out these measurements we took all of the computed vehicle poses for each measurement and computed the z scores of each position dimension (x,y,z) and each of the elements of it's rotation matrix. If any of the z scores for a measurement were over a threshold (Eg. 1.8 standard deviations) then we dropped that measurement.

This method works well when a large number of tags are seen. If only 1 or 2 tags are seen, then no measurement is dropped.

3) Fusing Apriltag Measurements: Multiple tag detections per frame were averaged using a GTSAM factor graph [6], which performs maximum likelihood estimation of the vehicle pose given the measurement means and covariances. This produced a single robust pose estimate per frame.

B. Delayed EKF

The Apriltag processing introduces a delay of up to 0.5s. A fixed-delay EKF (0.7 s) was implemented to ensure proper temporal ordering of measurements from AHRS, DVL, depth sensor, uGPS, and Apriltags. A secondary EKF was then periodically cloned and fast-forwarded through the accumulated measurements to the present and was then kept up-to-date using the low-latency AHRS, DVL, and depth sensor inputs. This allowed real-time pose estimates to be provided to the controller while maintaining a best estimate in the delayed EKF. Figure 4 shows how the fast-forwarding process works.

C. Dock EKF

Dock pose estimates were derived by combining Apriltag measurements with the vehicle's estimated pose, resulting in dock pose estimates in the global frame. A simple Kalman filter was used to fuse these measurements. The process model assumed a stationary dock with small process noise. This approach allows the filter to learn the docks pose even if the dock is actually slowly moving.

A benefit of having the docks pose recorded in the global coordinate frame is that if the vehicles estimated pose drifts while on a mission, then there is some loop closure where it re-learns its pose in the GCF then the previous drift will not affect its idea of where the dock is. However, if such a loop closure happens in the vehicles pose when the vehicle is close to the dock, the relative pose estimate between the vehicle and dock can have a sudden jump. This is not good for control and may cause collisions. This issue is currently unsolved but some suggested solutions can be found in V.

D. Docking Controller

The docking controller was developed only to the extent necessary to validate the Apriltag perception, and localization modules. It consisted of two states:

State 1: The vehicle faces the dock and moves towards a pre-docking waypoint in front of the dock. A velocity controller was used which drew the vehicle towards the waypoint while repelling it from collisions with the dock.

State 2: When close enough to the pre-docking pose, the vehicle executes a predefined trajectory into and out of the dock without making physical contact.

E. Diagnostic Plotter

This module provides real-time 3D visualization of tag positions, the dock, vehicle, control commands, and measurements. It uses LCM inputs and matplotlib rendering and the code is intended to be directly changed depending on what the user wants to display. Due to its computational cost, a dedicated laptop was needed for real-time operation.

V. FUTURE WORK

Below are four suggestions for future work.

A. Apriltag corner-based pose estimation:

The Apriltag-3 algorithm can output the corners of tags. The corners of all tags could be used at once with a PnP method to find the relative pose. This would improve the accuracy as it would better utilize the information in the image. It would also do away with the need to fuse individual measurements as is currently done. It would also make the covariance generation and handling easier and more robust. For instance a 2D covariance could be assigned to each of the detected corners and then followed through the relevant transformations

B. Make sure the relative pose between vehicle and dock is continuous:

Suppress or compensate for global Apriltag measurement updates when close to the dock. This could be done simply by ignoring the global Apriltags when close to the dock. Alternatively, when the vehicles EKF makes an Apriltag measurement update it could also update the docks estimated pose. This is easier said than done as this would introduce timing constraints so that the controller doesn't see an impulse in the relative poses.

C. Adaptive EKF delay:

Allow the delayed EKF to adapt its delay based on a real-time estimate of the Apriltag pipeline delay. Alternatively, If it is guaranteed that the Apriltag measurements come in order than the delay could simply be set to the time the last Apriltag measurement came through.

D. Modify the outputs of the Apriltag-3 algorithm:

There has been some discussion around using the 'Passive Correction for Frame Consistency' described in [7]. This modification to the Apriltag-3 algorithm could be implemented and tested with the current pipeline. This addition reduces the angular error of a tag detection however it reduces distance accuracy and may not be of overall benefit.

VI. CONCLUSIONS

Five Python modules were developed to enable robust perception and localization for underwater docking using Apriltag markers. The system fuses tag-based measurements with other sensor data via an EKF, allowing accurate and stable pose estimation suitable for control. While full docking was not yet achieved, the perception and localization components provide a solid foundation for further controller development for autonomous docking.

ACKNOWLEDGEMENTS

The MBARI Summer Internship Program is generously supported through a gift from the Dean and Helen Witter Family Fund and the Rentschler Family Fund in memory of former MBARI board member Frank Roberts (1920-2019) and by the David and Lucile Packard Foundation. Additional funding is provided by the Maxwell/Hanrahan Foundation.

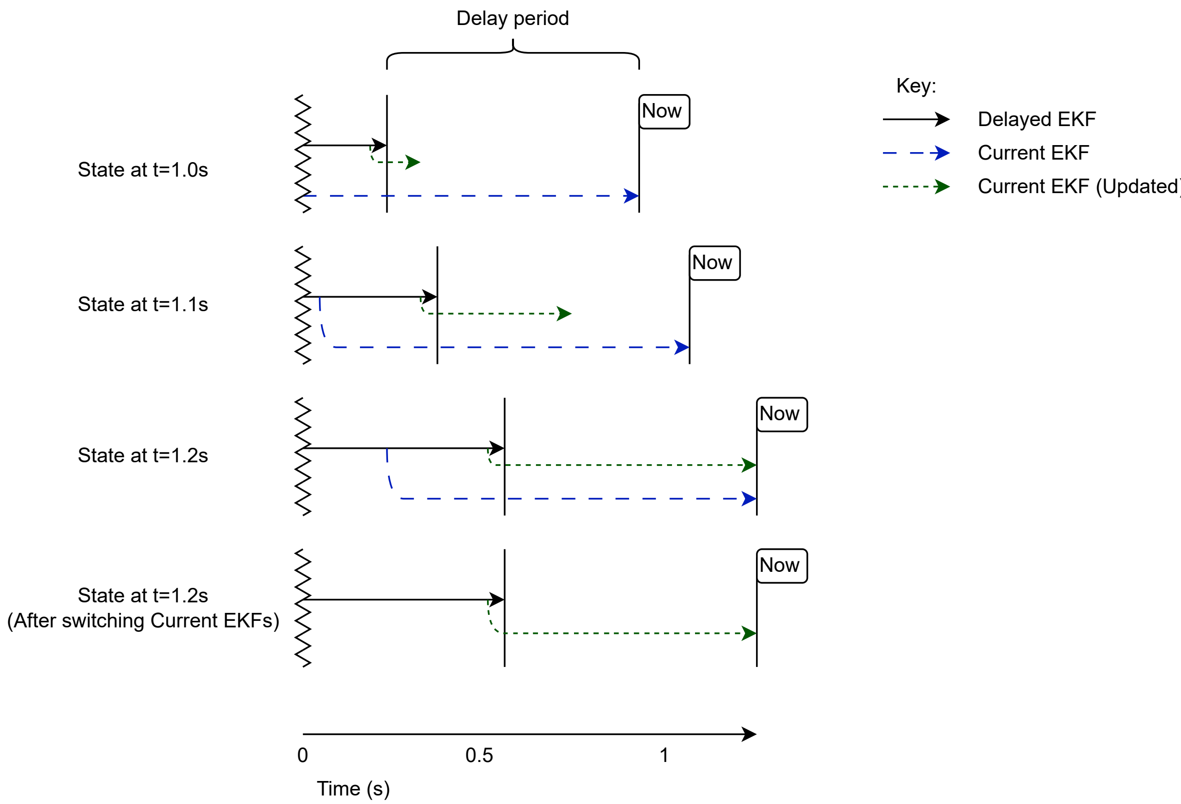

Fig. 4: *The EKF manager uses the best estimate from the delayed EKF and fast-forwards it to the current time.*

REFERENCES

[1] B. Hobson, R. McEwen, J. Erickson, T. Hoover, L. Mcbride, F. Shane, and J. Bellingham, "The development and ocean testing of an auv docking station for a 21 auv," pp. 1 – 6, 11 2007.

[2] J. Zhou, H. Huang, S. H. Huang, Y. Si, K. Shi, X. Quan, C. Guo, C.-W. Chen, Z. Wang, Y. Wang, Z. Wang, C. Cai, R. Hu, Z. Rong, J. He, M. Liu, and Y. Chen, "Auh, a new technology for ocean exploration," *Engineering*, vol. 25, 09 2022.

[3] J. Liu, F. Yu, B. He, and C. Guedes Soares, "A review of underwater docking and charging technology for autonomous vehicles," *Ocean Engineering*, vol. 297, 02 2024.

[4] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2019.

[5] A. Petrov and A. F. Daniele, "lib-dt-apriltags," 2021-10-6. Python bindings for the Apriltag 3 library.

[6] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022.

[7] M. Abbas, S. Aslam, K. Berns, and A. Muhammad, "Analysis and improvements in apriltag based state estimation," *Sensors*, vol. 19, 12 2019.