Evaluation of Dead Reckoning Navigation for Underwater Drones using ROS

Matthew Bellafaire
Department of Electrical and Computer Engineering
Oakland University
Rochester, MI, 48309
Email: mbellafaire@oakland.edu

Timothy Mayer
Department of Mechanical Engineering
Trine University
Angola, IN 56703
Email: tmayer18@aol.com

Edward Corlett
Department of Electrical Engineering
Florida State University Tallahassee, FL 32304
Email: eddycorlett@gmail.com

Osamah Rawashdeh, PhD, PE
Department of Electrical and Computer Engineering
Oakland University
Rochester, MI, 48309
Email: rawashd2@oakland.edu

Abstract

The motivation for this project was the creation of an Autonomous Underwater Vehicle (AUV) navigation system for the National Oceanic and Atmospheric Administration's (NOAA) 2021 *Ocean Observing Prize*, a competition aimed at developing innovative oceanic monitoring platforms. The final AUV must operate for up to 6-months unassisted in deep water. These conditions create challenges for underwater navigation as many conventional navigation sensors, such as GPS and Doppler Velocity Log (DVL), are unavailable for use in this environment. A Dead Reckoning approach was implemented in simulation using the Robot Operating System (ROS) and simulated through the UUV Simulator Gazebo package. In order to validate the algorithm, a small-scale AUV was implemented utilizing mostly COTS components. This work was part of the 2021 installment of an NSF-funded Research Experience for Undergraduates (REU) program in the Electrical and Computer Engineering Department at Oakland University.

Keywords

Autonomous Underwater Vehicle (AUV), ROS, Dead Reckoning, Navigation, REU

1. Introduction

Oceanic research has long been hampered by the limited number of sensors that can be deployed into the open ocean. This is prevalent in hurricane modeling research, where larger data sets would greatly increase our understanding of the ocean's effect on the formation of hurricanes. A proposed method of increasing the available oceanic data is the deployment of many Autonomous Underwater Vehicles (AUVs) capable of autonomously collecting data over a wide area during long missions. Without support ships, these AUVs would be deployed remotely, navigate to the location of interest, and loiter in the area collecting data on oceanic conditions leading up to hurricanes. To promote the development of such AUVs in the ocean, the National

Oceanic and Atmospheric Administration (NOAA) has created the *Ocean Observing Prize*[1]. The goal of the *Ocean Observing Prize* is to create new AUVs which are capable of operating for up to 6 months, unassisted in the open ocean, while recharging from wave power. This competition was the primary motivation for this project and this paper specifically focuses on the navigation aspect of AUVs.

Since the NOAA competition parameters require data to be collected in the top 200 meters of the ocean, many conventional navigation methods that operate based on access to the seafloor are unavailable. As a result, any proposed navigation system for the *Ocean Observing Prize* must contend with navigation based only on sensors that do not require external references. This includes commercially available Attitude and Heading Reference System (AHRS) or Inertial Navigation Systems (INS). Such systems can be used to provide positional data underwater through dead reckoning. In dead reckoning, a starting position is established to use as a reference, and then position change is continuously estimated based on sensor data and control inputs. However dead reckoning and inertial navigation methods suffer from increasing unbounded error over time, which leads to eventual loss of precision in navigation[2]. The unbounded error growth prevents the use of a navigation system based entirely on dead reckoning methods and thus other positioning methods must augment the information provided by dead reckoning.

Another tool available to an AUV operating in deep water is the Global Positioning System (GPS). GPS can provide high accuracy absolute position; however, it is unavailable under the surface of the water due to rapid attenuation of the satellite signal and therefore can only be used intermittently while navigating at the surface[3,4]. Therefore an AUV using GPS for absolute positioning must continually resurface to reorient itself in the global frame.

This paper presents progress in the development of an AUV navigation system for deep-water operation. Our system is built on the Robotic Operating System, which allows for a high degree of flexibility in development, simulation, and testing. We utilize a dead reckoning approach based on a simplified model of the AUV. Section 3 details the design of the navigation system and Test AUV and section 4 shows testing results of the simulation model and physical test platform.

2. Design

At the core of the presented system is the *Robot Operating System* (ROS). ROS was the chosen platform for the navigation system due to its wide range of tools for the development of robotic systems. These tools include packages for common tasks and for computer simulation through Gazebo, both of which expedite the development of navigation systems.

Implementation of the navigation algorithm was initiated in simulation utilizing the *UUV Simulator*[5] package created by Musa Morena Marcusso Manhães with the *LAUV* model. The UUV Simulator package provides a virtual environment through Gazebo to simulate kinematics, sensor inputs, and environments for an AUV. We then used ROS to create an algorithm that would rely on the sensors available to us in the application. Inputs to the navigation algorithm were provided through Gazebo for simulation, which can be mapped to real sensors in the final implementation.

2.1 Navigation

Because the navigation system must be capable of operating in deep water, the design was limited to sensors that did not rely on external references or supporting equipment. The sensors used for input consisted of an Inertial Measurement Unit (IMU), magnetometer, GPS receiver, and pressure sensor. With these inputs available, the system was developed to meet the goal of navigating on an arbitrary path determined by user-specified GPS waypoints. The flow of data through the navigation system, and the various software components, is shown at a high level in Figure 1.

A dead reckoning approach was utilized to facilitate position tracking underwater. At any given time, the AUV's position is represented in the East-North-Up (ENU) coordinate frame in meters. This position is constantly updated by sensor and control inputs through the position tracker. The position is then used by the path planner node to determine the desired heading. At the moment the AUV is powered on, the current position is set to (0,0,0) and all provided waypoints are mapped relative to that starting position.

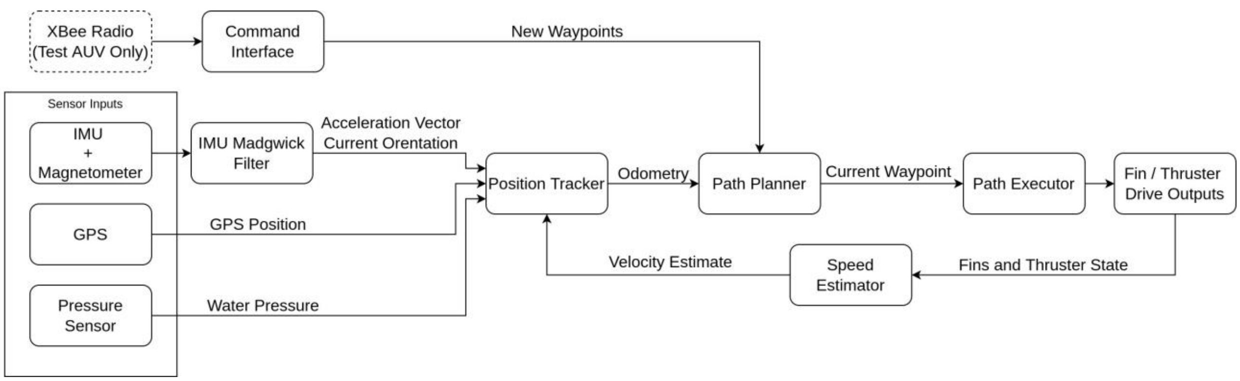

Figure 1: High-level overview of navigation system, inputs, outputs, and data flow

Drone initialization is assumed to occur at the ocean surface. The first GPS fix is recorded and used as a reference for mapping all following GPS points onto the ENU coordinate frame. In order to mitigate the increasing unbounded error of dead reckoning, continual corrections are performed by frequent surfacing operations performed by the AUV. Upon receiving a GPS fix, the current position of the AUV position tracker is recalculated relative to the first GPS coordinate. Conversion of spherical GPS coordinates into the Cartesian ENU coordinate frame is conducted using the Haversine Distance formula[6].

A path planner subroutine was created to track, add, and convert new waypoints into the ENU frame. The path planner can be interfaced with via text console commands, while on the physical test AUV this is performed wirelessly through a XBee radio module. Commands can be provided to add a new waypoint in the ENU frame, add a new waypoint by GPS coordinates, or add a new waypoint relative to the current position of the AUV.

The current heading ψ is provided through the use of the *imu_filter_madgwick*[7] ROS package. The IMU Madgwick Filter performs sensor fusion of the accelerometer, gyroscope, and magnetometer data, providing a more accurate pose estimation than any of the individual sensors could provide alone.

2.2 Localization

The navigation system proposed here utilizes dead reckoning as the only method for determining position underwater. In this method, the last known GPS position is used as a reference point from which the position underwater is tracked based on sensor inputs using the East-North-Up (ENU) coordinate scheme. Position for a given time t is determined by adding the current estimated velocity multiplied by the time step (ΔT) for the x and y-axis based on the AUV's heading (ψ), then adding the result to the previous position estimate as shown in Equation 1.

$$s_x(t) = s_x(t-1) + v_e(t)cos(\psi)\Delta T$$
$$s_y(t) = s_y(t-1) + v_e(t)sin(\psi)\Delta T \tag{1}$$

In the approach utilized here the navigation accuracy of the dead reckoning method is heavily dependent on the accuracy of the velocity estimate. A model was experimentally developed to estimate velocity under the surface at a given time t utilizing the command outputs to the propeller and fins. The model presented in Equation 2 takes into account the drag induced by the fins when being driven out of their zero position which reduces the overall AUV velocity.

$$v_e(t) = v_{cmd}(t) * v_b e^{r_f\sqrt{s_p(t)^2+s_y(t)^2}} \tag{2}$$

The estimated velocity $v_e(t)$ is calculated from the commanded propeller output and the commanded fin positions of the pitch and yaw fins, s_p and s_y respectively, on a range of -1 to 1. $\boldsymbol{v_b}$ represents the velocity of the AUV in m/s in a straight line with maximum propeller output. An exponential relation was identified between the deviation of the fins and the velocity of the AUV, which is approximated through the r_f coefficient. The values for v_b and r_f are determined through a guess-and-check python script used on recorded data for both the physical AUV and simulation model.

2.3 Test AUV

Validation of the system was to be performed with a small test AUV created by modifying an off-the-shelf RC submarine built by *Nautilus Drydocks*. The RC submarine is 42" in length and contains two servos for the yaw and pitch fins with a DC motor to operate the propeller. The modified RC submarine is shown in Figure 2. Modifications were conducted to create a ROS compatible navigation computer capable of running the proposed navigation system with the required sensors, utilizing Commercial-Off-The-Shelf (COTS) components.

Figure 2: Test AUV modified from an off-the-shelf remote-control submarine produced by *Nautilus Drydocks*

The primary controller of the test AUV is a *Raspberry Pi 4* running Ubuntu 20.04 with *ROS Noetic*. For the IMU an MPU9250 was chosen to provide inertial measurements, while an HMC583L Magnetometer provides heading information. An off-the-shelf GPS module was selected utilizing a USB interface that communicates serially at 4800 baud. To issue control commands to the AUV an XBee radio communicates with the Raspberry Pi via USB. Finally, an Adafruit 16-channel servo shield provides interfacing to control fin actuators and the propeller.

The test sub is powered via a 3-cell lithium polymer battery which directly powers the DC motor controller. To power the Raspberry Pi and additional sensors the battery voltage is regulated via a buck converter to bring the voltage down to 5 volts. An additional remote power-on-off switch is included which disables battery power through a radio-controlled relay.

3. Results

3.1 Simulation

Testing of the simulated navigation algorithm was conducted through Gazebo using the *UUV Simulator* package. A test path was created covering 32.12 kilometers with no oceanic currents introduced. The simulated AUV was set at (0,0) and the waypoints were provided as GPS coordinates. Surfacing operations were performed once every 30 minutes of operation. The path taken by the AUV is shown in Figure 3 with both the ground-truth and estimated position shown.

The total time required to complete the course was 2 hours and 57 minutes. In simulation, the ground-truth position of the AUV is provided and is used in the evaluation of the position estimation. Figure 4 shows the error present in odometery as given by the difference between the ground truth and estimated positions at any given time. Over the course of the run the maximum error was 261.62 meters and the average was 82.17 meters. The standard deviation for all samples collected was 64.41 meters.

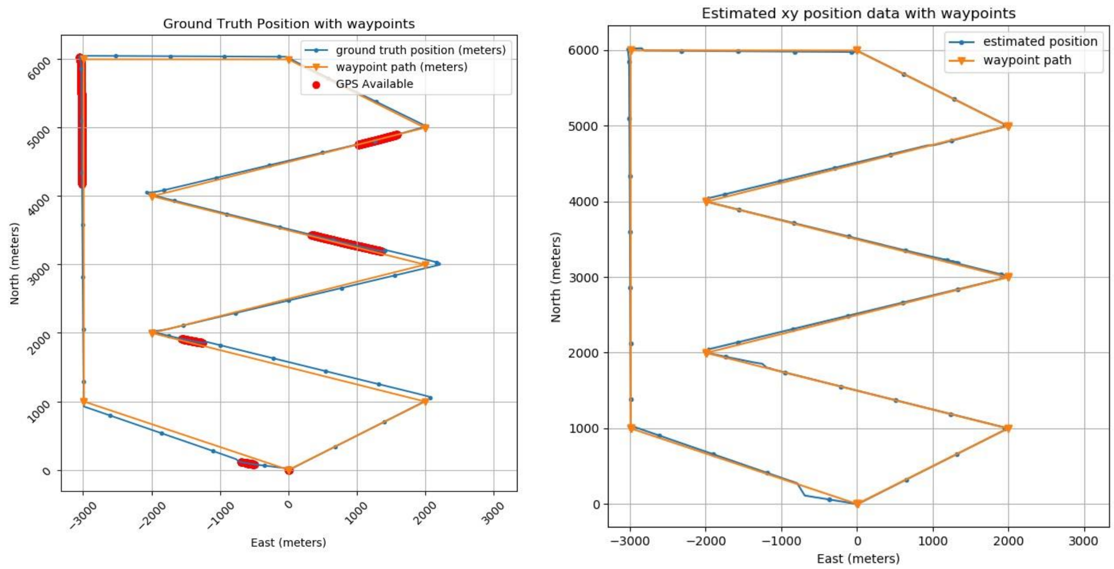

Figure 3: Trajectory of simulated AUV traversing the 32-kilometer path

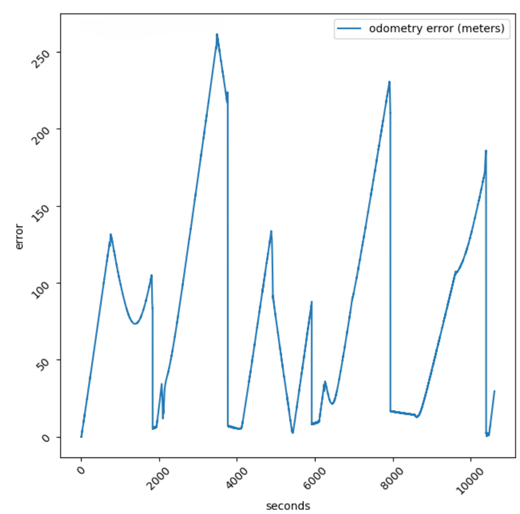

Figure 4: Odometery error of the simulated AUV traversing the 32-kilometer path

3.2 Test AUV

Validation of the algorithm work is ongoing, working to interface the navigation algorithm with a small test AUV capable of limited operation. The results presented here are from a recent run of the Test AUV on the shore of Lake St. Clair in Southeast Michigan. The Test AUV was commanded to travel between 3 GPS coordinates which were entered manually through text commands and then sent via the XBee radio module. The radius of acceptance for the waypoints was 10 meters in this test. The Trajectory of the Test AUV is shown in Figure 5.

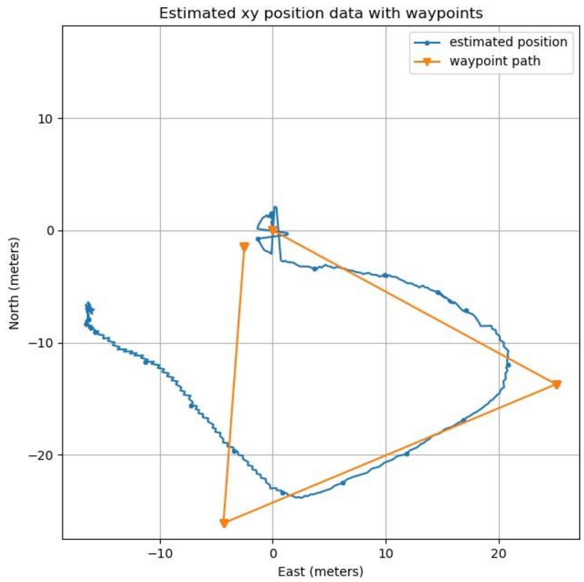

Figure 5: Trajectory of test AUV in one test course

The Test AUV operated at the surface over the full run and was capable of completing the first 2 waypoints without issue. After the third waypoint, the compass heading was incorrect leading to the Test AUV impacting a sea wall at which point the test was halted.

4. Conclusion

In this paper, the progress towards a dead reckoning-based navigation system, built in ROS and simulated in Gazebo along, was presented and validated using a physical AUV platform. The simulation test results show acceptable performance for the intended application of an AUV operating in deep water to collect information about the top layer of the ocean for use in hurricane modeling. Using the simulation environment, and the components already created, we have a viable platform to continue work on developing dead reckoning techniques for our intended application and for similar AUV applications.

5. Future Work

Future work will focus on utilizing the constructed ROS system with localization techniques available in the literature and improving upon current techniques for underwater navigation. Kalman filters and machine learning approaches are primary solutions for addressing challenges in dead reckoning that will be evaluated using the simulation environment and the test vehicle developed as part of this project.

Acknowledgements

This program was supported by the National Science Foundation Grant No EEC-1659650. Additional support was provided by the School of Engineering and Computer Science at Oakland University. In addition, we would like to thank Mark Supal for providing the initial motivation for the project and the Test AUV for testing.

References

[1] U.S. Department of Energy, "Ocean observing prize official rules," Apr 2021.
[2] P. Prusaczyk, J. Panasiuk, and L. Baranowski, "Gps-free navigation based on using inertial and odometry, data fusion and map matching algorithm," AIP Conference Proceedings, vol. 2029, no. 1, p. 020062, 2018.
[3] T. Hyakudome, "Design of autonomous underwater vehicle," International Journal of Advanced Robotic Systems, vol. 8, no. 1, p. 9, 2011.
[4] J. Gonzalez-Garcıa, A. Gomez-Espinosa, E. Cuan-Urquizo, L. G. Garcıa-Valdovinos, T. Salgado-Jimenez, and´ J. A. E. Cabello, "Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions," Applied Sciences, vol. 10, no. 4, 2020.
[5] M. M. M. Manhaes, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV simulator: A˜ gazebo-based package for underwater intervention and multi-robot simulation," in OCEANS 2016 MTS/IEEE Monterey, IEEE, sep 2016.
[6] w.-t. Chris Veness, "Calculate distance, bearing and more between latitude/longitude points," 2020.
[7] "imu filter madgwick - ros wiki," 2020.