Waypoint Follower for an Autonomous Underwater Vehicle

Andrew Albano

I. Introduction

An autonomous underwater vehicle (AUV) must be capable of performing a variety of tasks, with a variety of equipment. Changing equipment can produce changes in the system dynamics, and the motion controller must be robust or easily tunable to account for the changes. This prompts the objective of this project, which aims to convert the BlueROV Heavy Configuration from a remote-operated vehicle (ROV) to an AUV, specifically by implementing a motion controller to follow a path. If time permits, the controller and motion planner will incorporate obstacle avoidance, given an occupancy map with information on the 3D scene.

The inherited state of the controller was a proportional controller with hard-coded waypoint poses. The poses were restricted to the $x-y$ plane, and an animated 2D plot was used to visualize the waypoints and the current pose and heading. The primary deliverable is a control algorithm that can reach a desired waypoint position and attitude under the assumption of four degrees of freedom, x, y, z, and ψ (yaw). For testing purposes, this project will develop a simple user interface for generating waypoints and attitudes relative to the NED frame, relative to the last waypoint added, and relative to the current pose, along with an interface for adjusting parameters such as controller gains and saturation limits during live testing. In addition to the goal waypoint, intermediate waypoints will be generated to to form a path from the starting location to the goal location. The goal waypoints and and desired path will be visible in the current RViz window. For tuning purposes, live plots will be used to visualize the reference setpoint and current state of each degree of freedom. The original proportional controller will be used as a baseline for analysis, for which alternative controllers will be compared. Due to the complex dynamics involved and the potential for sensors to change to suit research needs, control algorithms that do not require dynamic models will be investigated with a higher priority.

II. Related Research

A. Waypoint Following Methods

The current state of the project is a functional PID controller (in simulation). Although initial efforts focus on PID control, this section will explore alternatives that could be considered if a dynamics-free approach does not yield reasonable results once tuned. Related research often compares PID controllers with sliding mode control(SMC) [1]. SMC technique is a robust control tool for complex higher order nonlinear systems under the effect of parametric uncertainties and external perturbations [2]. SMC technique provides outstanding performance while dealing with bounded uncertainties/disturbances and unmodelled dynamics [3] over the other established techniques like robust adaptive control, H infinity control, and backstepping control [2]. Most research related specifically to BlueROV2 uses sliding mode control such as [4]–[6].

The author of [1] compares the performance of a classic proportional derivative controller with that of one containing dynamic parameters of the underwater vehicle in simulations. This verified that some knowledge of the dynamics allows for better selection of controller gain coefficients and results in faster convergence to steady state. While the structure of PID control is simple and robust, the authors of [7] found that experimental results agree with the analytical predictions that model-based controllers outperform PD control over a wide range of operating conditions.

Underwater vehicles present complex control-system design problems due to their nonlinear dynamics, uncertain models, and disturbances that are difficult to measure or estimate [4]. A common strategy to address the unpredictable behavior of these vehicles is to use trajectory control techniques with sliding modes.

In [5], the authors developed a simulation model environment of the BlueROV2 in Simulink. Their model is based on Fossen's equations and includes a kinematic model of the vehicle, the hydrodynamics of the vehicle and water interaction, a dynamic model of the thrusters, and, lastly, the gravitational/buoyant forces. Their case study uses a sliding mode controller to follow a trajectory with low error while encountering disturbance from ocean currents. While this study highlights the ability of the sliding mode control to control the vehicle in all 6 degrees of freedom, it did not consider physical implementation on the hardware through ArduSub or the use of MAVlink protocol, and the controller implementation was not validated on the hardware.

Similarly, the authors of [6] developed a sliding mode controller with integral action after determining that the BlueROV2 was unstable when disturbed by a water jet with sliding mode control alone. The sliding mode controller was stable while following a trajectory without the jet disturbance. A linear quadratic regulator controller was also designed based on [8] for comparison. The sliding mode controller yielded better results when tracking a moving reference setpoint, while the linear quadratic regulator was better suited when the reference setpoint was constant.

III. Current State, Planned Tasks, and Milestones

A. Waypoint Manager and Tuning Interface

The current waypoint manager interface is shown in Figure 1 Currently, waypoints can be defined relative to the NED frame, relative to the last waypoint added, and relative to the current pose. Generating a waypoint relative to a latitude and longitude will also be added. Waypoints are published in a geometry_msgs PoseArray format in the NED frame. The waypoints can be visualized individually, simultaneously, and as a path in RViz to verify them before submitting them to the motion controller as is shown in Figure 2. Additionally, a home position can be set to ensure a consistent location is returned to. The waypoint manager also has preset routes such as a square to generate tests patterns quickly. Additional routes will be generated to benchmark the motion controller's performance such as an orbit mode (traveling in a circle with the heading facing the center of rotation), and lawnmower pattern. As a safety precaution, a disable motion controller button was added for emergency shutoff. The controller gains can also be tuned during testing using the interface. To assist with tuning the controller gains the x, y, z, and yaw velocity setpoints and control signal setpoints are plotted on a sliding axis to observe overshoot and settling time during operation as shown in Figure 3. All user interface features have been tested in the simulation environment and are ready for hardware testing.

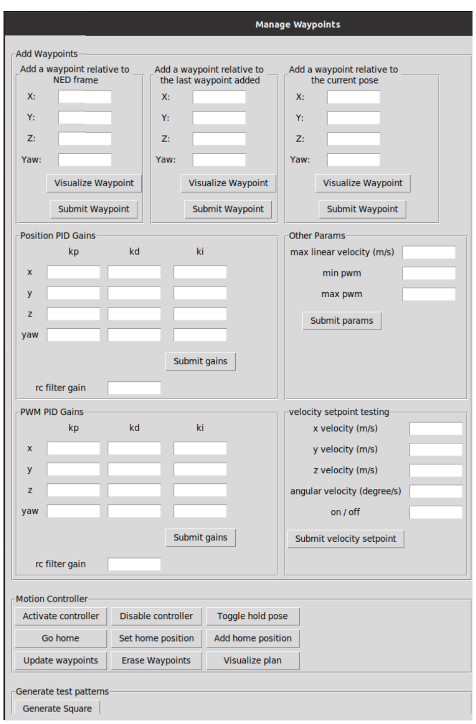

Fig. 1. User interface for managing waypoints and controller gains during testing

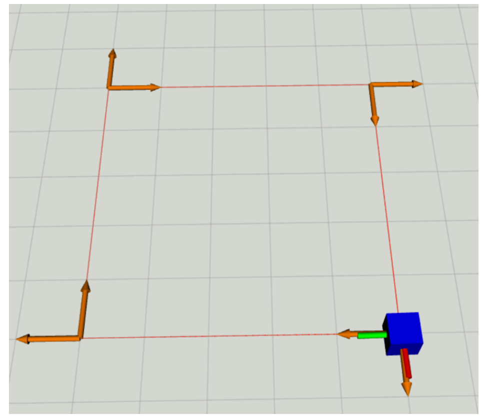

Fig. 2. Sample waypoints and path visualized in RViz from the generate square test pattern

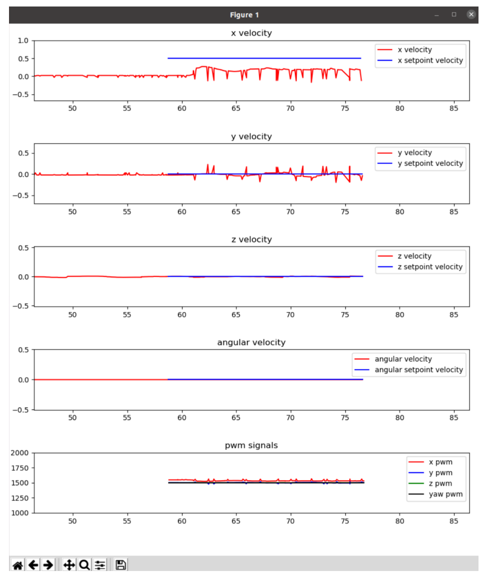

Fig. 3. Sample of live plotting velocity setpoints and current velocity for controller gain tuning. Note that controller gains are not tuned yet.

B. Motion Controller

The initial motion controller only used proportional control based on the position to assign a value for the PWM signal. The updated motion controller contains two control loops. The first is a PID controller for the position which outputs a velocity setpoint that is saturated to a maximum velocity set in the user interface. A second PID controller uses the velocity feedback to adjust the x, y, z, and ψ output signals to reach the setpoint velocity. Due to the control architecture for the BlueROV2, these signals are equivalent to replacing the signal that would otherwise be sent from the joysticks. A thruster allocation matrix is used to convert these signals to usable PWM signals, the details of which are described more fully in the literature in the related works section and are already in use by the BlueROV2. Decoupling each degree of freedom does present an issue worth considering; the BlueROV2 could potentially move in a direction that the sensors do not have line-of-sight information resulting in a potential collision if there is drift from the actual pose in the state estimation or an object not previously accounted for during mapping. An alternative approach is to restrict the BlueROV2 to three degrees of freedom, x, z, and yaw similar to a rotate, translate, rotate controller architecture to ensure there are no obstacles in the line of sight. Both approaches will be tested in simulation and with hardware. Once the PID controllers are benchmarked and if time allows, an implementation of the sliding mode control algorithms presented in either [4], [5], or [6] will be benchmarked using the hardware.

C. Simulation Environment

Simulation tests are currently being conducted using an ArduSub software in the loop (SITL) simulation with the BlueROV2 Heavy. This was used to verify that the waypoint interface and motion controller are compatible with the BlueROV2 software. More research into the back-end of simulation needs to be completed to verify if this is meant to be an accurate representation of the dynamics or if it is only meant to be used as a software emulator. If it is not sufficient, the dynamic models presented in [5] could be used to provide a more reasonable estimation of the dynamics and provide a baseline for the controller gains before tuning with the hardware.

D. Testing

A sample of the current motion controller using the SITL simulation to navigate to a waypoint is shown in Figure 4 using RViz. The controller gains have not been tuned using well known data-driven techniques such as the Ziegler-Nichols method [9].

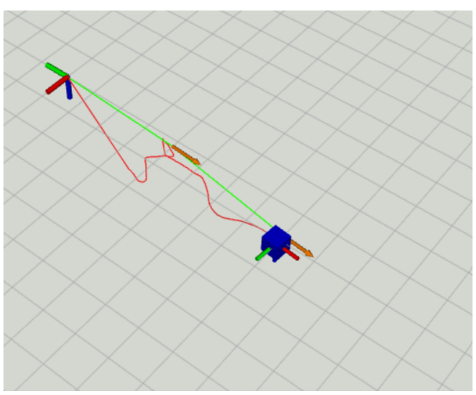

Fig. 4. Sample waypoint following SITL simulation results prior to implementing tuning methods

E. Planned Tasks and Timeline

TABLE I
MILESTONES AND DIVISION OF WORK

Date	Milestone
Jan 31	Research and test ROS nodes
Feb 7	Create waypoint interface and visualize waypoints using RViz
Feb 14	First iteration of control algorithm and set up simple simulation for testing
Feb 28	Second iteration of control algorithm, make compatible with ArduSub SITL simulation, initial tests on hardware to confirm signals, prepare tools for tuning controller gains
Mar 7	Research tuning methods and research ArduSub simulation dynamics
Mar 14	Tune and benchmark 4 dof algorithm and RTR controller on hardware
Mar 28	Evaluate results, decide whether implementation is sufficient or attempt SMC implementation
Mar 28 - End of semester	TBD based on results from Mar 28 testing

REFERENCES

[1] P. Herman, “Decoupled pd set-point controller for underwater vehicles,” *Ocean Engineering*, vol. 36, pp. 529–534, 2009.

[2] S. Gambhire, D. R. Kishore, P. Londhe, and S. Oawar, “Review of sliding mode based control techniques for control system applications,” *International Journal of Dynamics and Control*, vol. 9, p. 363–378, 2021.

[3] V. Utkin, J. Guldner, and J. Shi, *Sliding mode control in electromechanical systems*. Routledge, Taylar & Francis Ltd, 1999.

[4] D. Yoerger and J. Slotine, “Robust trajectory control of underwater vehicles,” *IEEE Journal of Oceanic Engineering*, vol. 10, pp. 462 – 470, 1985.

[5] M. V. Benzon, F. F. Sørensen, U. Esben, J. Jouffroy, J. Liniger, and S. Pedersen, “An open-source benchmark simulator: Control of a bluerov2 underwater robot,” *Journal of Marine Science and Engineering*, vol. 10, 2022.

[6] M. V. Benzon, F. F. Sørensen, J. Liniger, S. Pedersen, S. Klemmensen, and K. Schmidt, “Integral sliding mode control for a marine growth removing rov with water jet disturbance,” *2021 European Control Conference (ECC)*, pp. 2265–2270, 2021.

[7] D. Smallwood and L. Whitcomb, “Model-based dynamic positioning of underwater robotic vehicles: theory and experiment,” *IEEE Journal of Oceanic Engineering*, vol. 29, pp. 169 – 186, 2004.

[8] S. Pedersen, J. Liniger, F. Sørensen, K. Schmidt, M. v. Benzon, and S. Klemmensen, “Stabilization of a rov in three-dimensional space using an underwater acoustic positioning system,” *IFAC-PapersOnLine*, vol. 52, pp. 117 – 122, 2019.

[9] K. Åström and T. Hägglund, “Revisiting the ziegler–nichols step response method for pid control,” *Journal of Process Control*, vol. 14, pp. 635–650, 2004.