

BACKSTEPPING-BASED TRAJECTORY TRACKING FOR UNDERWATER GLIDERS

Demetris Coleman

Smart Microsystems Laboratory
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan 48823
Email: colem404@msu.edu

Maria Castaño

Smart Microsystems Laboratory
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan 48823
Email: castanom@msu.edu

Osama Ennasr

Smart Microsystems Laboratory
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan 48823
Email: ennasros@msu.edu

Xiaobo Tan

Smart Microsystems Laboratory
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan 48823
Email: xbtan@egr.msu.edu

ABSTRACT

Autonomous underwater gliders have become valuable tools for a myriad of applications ranging from ocean exploration to fish tracking to environmental sampling. To be suitable for these types of applications, precise sensing and monitoring is desired, which makes accurate trajectory control important. However, highly nonlinear under-actuated dynamics present significant challenges in control of gliders. In this work a backstepping-based controller is proposed for an underwater glider to track a desired position and heading reference in the sagittal plane with only two control inputs, the buoyancy and center of gravity along the longitudinal direction. In particular, the under-actuation issue is addressed by exploiting the coupled dynamics and introducing a new modified error that combines the tracking errors of heading and position references. In addition, an auxiliary system is incorporated to account for input constraints. Finally, a sliding mode observer is designed to obtain the estimates of body-fixed velocities, to facilitate practical implementation of the designed controller. The effectiveness of the proposed control scheme is demonstrated via simulations and its advantages are shown via comparison with a PID controller.

1 INTRODUCTION

Underwater gliders realize horizontal travel by shifting the center of gravity and changing the buoyancy in tandem. The concept was first introduced by Henrey Stommel [1] and motivated several gliders, most notably SLOCUM [1], Spray [2], and Seaglider [3]. These vehicles are known for high energy efficiency, allowing them to have exceptionally long operation time. The latter has inspired the development of other underwater vehicles that exploit gliding. One example is the gliding robotic fish [4], which combines the gliding mechanism with the tail-actuated swimming [5] to realize both high energy-efficiency and high maneuverability. It has demonstrated promise in environmental sensing applications [6]. Given the myriad of applications in oceanography, marine science, water quality monitoring, and surveillance, it is of interest to realize precise trajectory tracking for underwater gliders and other gliding type-vehicles. However, control of gliders presents a significant challenge, due to their highly nonlinear and under-actuated dynamics.

Early work in control of gliders saw the use of PID controllers for their simplicity [2], [3]. In the past 2 decades, more advanced control methodologies have been proposed for steady-

state glide stabilization, position control, path following, and trajectory tracking in literature. Many of these can be seen in the review by [7]. Sliding mode control is often adopted because of its robustness to disturbances. Mat-Noh et al. used a linearized glider model to compare an Integral Super Twisting Sliding Mode controller to several other sliding mode variants for stabilizing a gliding path between 30 degrees and 45 degrees [8]. Yang and Ma used sliding mode control to track trajectories of the pitch angle and ballast mass [9]. A number of other control methods have been proposed for underwater gliders. For example, Leonard and Graver [10], [11] use a linear quadratic regulator on linearized dynamics to control the magnitude of velocity on a steady state glide path. Mahmoudian and Woosely develop an efficient path planning strategy that concatenates equilibrium turning and gliding motions, then implement the strategy using PID controllers to reach specified center of gravity and center of buoyancy [12]. Neural network-based control was used to implement a self-tuning PID controller to track the velocity along a single axis in the inertial frame [13]. Isa and Arshad analyzed the use of a neural network as a model predictive controller and a gain tuner algorithm to control pitch angle and linear velocities of a linearized glider model [14]. Nag et al. [15] compared fuzzy logic control against PID for pitch and depth tracking. Zhang et al. [16] used nonlinear passivity-based control to stabilize the glide path of a glider with a tale.

The aforementioned control approaches have largely focused on stabilization based on linearized models, or heading and velocity control. In fact, most work on underwater glider control focus on controllers designed to reach a desired pitch angle, velocity, or specified depth [17]. The control of gliders has progressed, but a systematic model-based control approach for trajectory tracking that accommodates nonlinear under-actuated dynamics and input constraints for such systems is still lacking. In this paper, a backstepping-based trajectory tracking controller is proposed for sagittal plane motion of underwater gliders, including the heading, horizontal position, and vertical position (depth). Backstepping-based control design presents a practical and promising approach because it offers a systematic framework that guarantees the stability of the closed-loop system, and allows the accommodation of input constraints.

Limited work has been reported in on backstepping-based control of underwater gliders [17]. In [17] the authors proposed an adaptive backstepping control for tracking of the roll angle, pitch angle, and velocity of an underwater glider. In addition, most works stray away from the position tracking problem because of the difficulty in measuring horizontal-plane position underwater. However, this is still a viable topic of interest for applications where there are relative positions such as target tracking. Our proposed trajectory tracking scheme in this paper, on the other hand, addresses simultaneous tracking of heading, depth, and horizontal position that are of direct relevance to various sampling and target tracking applications for the un-

derwater environment. The proposed scheme accommodates the under-actuation nature of the glider by exploiting coupling of the dynamics, and it incorporates input constraints via an auxiliary system. In particular, inspired by [18], a new error coordinate dependent on both the heading and the horizontal position error is designed, such that the vehicle's pitch velocity can be used as a virtual input to regulate the aforementioned modified error. In this manner, the controller is able to handle the tracking of both heading and horizontal trajectories, apart from tracking the reference in the depth direction. In addition, to compensate for the error due to the difference between feasible and "desired" inputs, an auxiliary system is incorporated. Furthermore, to implement the trajectory tracking control, a sliding mode observer is implemented to estimate the body-fixed velocities which are otherwise not directly accessible. The rest of the paper is organized as follows. The dynamic model of a glider in the sagittal plane is described and the error dynamics are derived in Section 2. In Section 3 the proposed controller scheme is presented. Simulation results are provided in Section 4, followed by concluding remarks in Section 5.

2 System Modeling

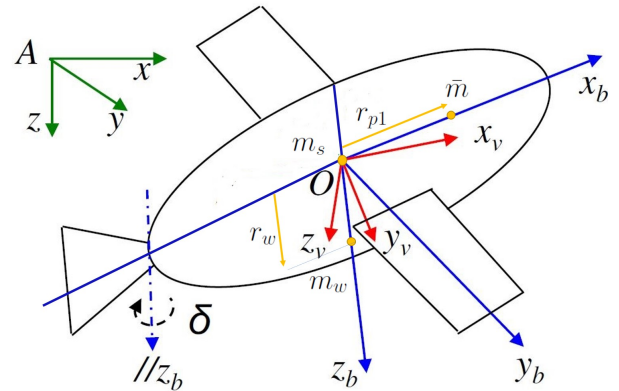


Figure 1. Illustration of the reference frames and mass distribution for a glider [4].

2.1 Glider Model

The glider in this work, pictured in Fig. 1, has two relevant reference frames. The first is the inertial frame, represented by A_{xyz} . The A_z axis is along the direction of gravity, and A_x and A_y are defined in the horizontal plane, with the origin A as a fixed point in space. The body-fixed frame is denoted by $O_{x_b y_b z_b}$ with the origin O at the geometry center of the glider body. The O_{x_b} axis is along the body's longitudinal axis pointing to the head,

the O_{z_b} axis is perpendicular to the O_{x_b} axis in the sagittal plane of the glider pointing towards the vehicles's underbelly, and O_{y_b} axis is formed according to the right-hand orthonormal principle. The glider is modeled as a six degree-of-freedom (DOF) rigid body with an internal moving mass (for adjusting center of gravity) and a ballast water tank (for buoyancy control). The internal movable mass, which is restricted to the longitudinal axis, has direct influence of the pitch angle, and through the coupling of dynamics, influences motion in other degrees of freedom as well. The negative net buoyancy is given as the sum of the uniformly distributed stationary mass m_s (including the water in tank), internal movable mass $\bar{m}$, and non-uniformly distributed mass m_w minus the mass m of the water displaced by the vehicle. This can be expressed as $m_0 = m_s + \bar{m} + m_w - m$ where $m_0 < 0$ cause the vehicle to float and $m_0 > 0$ causes the vehicle to sink. Here m_0 is effectively determined by the water pumped in and out of the ballast tank.

In this paper, the aim is to control the motion in the sagittal plane, which is the primary mode of locomotion for gliders. In this case, the control inputs are the negative net buoyancy m_0 and the distance r_{p1} of a movable mass from the center of gravity. The six dimensional state vector consisting of the position $[x, z]$ and pitch orientation θ of the vehicle given in the inertial frame and the body-fixed linear velocities $[v_1, v_3]$ and pitch angular velocity ω_2 is given by

$$X = [x, z, \theta, v_1, v_3, \omega_2]^T \quad (1)$$

There dynamic equations are given by [4]

$$\dot{X} = \begin{bmatrix} v_1 \cos \theta + v_3 \sin \theta \\ -v_1 \sin \theta + v_3 \cos \theta \\ \omega_2 \\ \frac{1}{m_1 + \bar{m}} (- (m_3 + \bar{m}) v_3 \omega_2 - m_0 g \sin \theta + L \sin \alpha - D \cos \alpha) \\ \frac{1}{m_1 + \bar{m}} ((m_1 + \bar{m}) v_1 \omega_2 + m_0 g \cos \theta + L \cos \alpha - D \sin \alpha) \\ \frac{1}{J_2} (M_2 + (m_3 - m_1) v_1 v_3 + m_w g r_{w3} \sin \theta - \bar{m} g r_{p1} \cos \theta) \end{bmatrix} \quad (2)$$

where $\alpha = \arctan \frac{v_3}{v_1}$ is the angle of attack, and m_1 and m_3 are components of the added mass due to surrounding fluid. The hydrodynamic forces of lift, drag and pitch moment are given as

$$\begin{bmatrix} L = \frac{1}{2} \rho V^2 S (C_{L0} + C_L^\alpha \alpha) \\ D = \frac{1}{2} \rho V^2 S (C_{D0} + C_D^\alpha \alpha^2) \\ M_2 = \frac{1}{2} \rho V^2 S (C_{M0} + C_{M_P}^\alpha \alpha + K_{q2} \omega_2) \end{bmatrix} \quad (3)$$

where constants with 'C' in their notations are hydrodynamic coefficients, ρ is the density of water, $V = \sqrt{v_1^2 + v_3^2}$ is the total magnitude of the velocity, S is the characteristic area of the vehicle and K_{q2} is a rotational damping coefficient. For convenience, we will refer to the accelerations as

$$\begin{bmatrix} \dot{v}_1 \\ \dot{\omega}_2 \\ \dot{v}_3 \end{bmatrix} = \begin{bmatrix} f_1(v_1, \omega_2, v_3) + u_1 g_1(\theta) \\ f_2(\theta, \omega_2, v_1, v_3) + u_2 g_2(\theta) \\ f_3(v_1, \omega_2, v_3) + u_1 g_3(\theta) \end{bmatrix} \quad (4)$$

where $u_1 = m_0$ and $u_2 = r_{p1}$.

2.2 Error Dynamics

The problem of trajectory tracking involves controlling a vehicle to follow a time-dependent path. In this work, aim is to have the pose $P = [x, z, \theta]^T$ follow a trajectory in the inertial coordinate system given by the desired path $P_d(t) = [x_d(t), z_d(t), \theta_d(t)]^T$. To solve this problem, the inertial frame error $P_e(t) = [x_e, z_e, \theta_e]^T$ is define as

$$P_e(t) = \begin{bmatrix} x - x_d \\ z - z_d \\ \theta - \theta_d \end{bmatrix} \quad (5)$$

where we drop the time dependent notation for convenience. The derivative of $\dot{P}_e = \dot{P} - \dot{P}_d$ is given by

$$\dot{P}_e = \begin{bmatrix} \dot{x} - \dot{x}_d \\ \dot{z} - \dot{z}_d \\ \dot{\theta} - \dot{\theta}_d \end{bmatrix} \quad (6)$$

Since the velocity dynamics in equation (4) are given in the body-fixed frame, we denote R_B^I as the rotation matrix from the inertial frame to the body-fixed frame. R_I^B is the inverse of R_B^I defining the rotation matrix from the body-fixed frame to the inertial frame, and therefore, $\dot{P} = R_I^B(\theta) [v_1, v_3, \omega_2]^T$ is given by the first three entries of $\dot{X}$.

With this formulation, trajectory tracking becomes a stabilization problem with respect to the error dynamics. The control objective is now to drive the kinematic error vector P_e to a region around the origin $[0, 0, 0]^T$.

3 Control Design

3.1 Overview of Control Design

Given the error dynamics and problem formulation, there are a few choices of Lyapunov functions that may, at first, seem like good candidates for the design of a backstepping-based trajectory tracking controller for underwater gliders. For instance, a natural

choice would be to use the function $V = \frac{1}{2}(x_e^2 + z_e^2 + \theta_e^2)$. However, due to the under-actuated nature of the system and coupled effects of the inputs, this choice can lead to an over constraint on the design of the control law. The choice $V = \frac{1}{2}(x_e^2 + z_e^2)$ only tracks the position variables and requires the derivatives of the accelerations and inputs. The choices $V = \frac{1}{2}(z_e^2 + \theta_e^2)$ and $V = \frac{1}{2}(x_e^2 + \theta_e^2)$ allow for design of controllers that are only capable of tracking one position variable along with the pitch angle which can be done reasonably well by a PID controller.

In order to achieve tracking for the entire set of references (heading and horizontal and vertical positions), inspiration is taken from [18] and a new variable is defined

$$\begin{aligned} y_e &= \theta_e - c \sin(\theta) \tanh(x_e) \\ &= \theta - (\theta_d + c \sin(\theta) \tanh(x_e)) \end{aligned} \quad (7)$$

to be used in the Lyapunov analysis. The motivation for this variable comes from the fact that the system does not have direct control over translation in the horizontal direction. Instead, it achieves forward translation by conversion of vertical velocity via the lift force which is indirectly dependent on the pitch angle. Since this is the only means to achieve translation, it is a natural way to correct for the error in the x position. y_e is chosen to allow the pitch angle to be used to minimize x_e , but still allow tracking of θ_d when x_e is small enough. In particular, $x_e \Rightarrow 0$ and $y_e \Rightarrow 0$ implies that $\theta_e \Rightarrow 0$. In (7), c is a positive constant specifying a maximum perturbation from θ_d due to x_e . Since $|\sin(\theta) \tanh(x_e)| < 1$, to keep the magnitude of the error x_e from completely dominating the pitch error θ_e , c should be thought of as the maximum control authority dedicated to minimizing x_e . The $\sin \theta$ term plays two roles. First, when θ is 0 or simply very small, the ability to control translation in the longitudinal direction is lost, so less weight is placed on the correction angle. This also mitigates the possibility that the summation of θ_e and the correction angle will negate each other to make $\theta = 0$ and cause bad tracking. However, this also means that the vehicle cannot minimize x_e if θ_d and θ are zero, meaning it is not expected to correct the x position during a flat dive trajectory. Secondly, the sign of the correction angle is determined by both the orientation and the position error. As it turns out, the sign of the correction angle should be the product of the sign of x_e and θ which is handled by the signum-like properties of the hyperbolic tangent and sin functions.

We choose the Lyapunov function $V = \frac{1}{2}(z_e^2 + y_e^2)$ to design the control laws for u_1 and u_2 . $\dot{V}$ can be made negative definite by choosing $\dot{y}_e = -k_y y_e$ and $\dot{z}_e = -k_z z_e$.

$$\begin{aligned} \dot{V} &= z_e(-v_1 \sin \theta + v_3 \cos \theta - \dot{z}_d) + y_e(\omega_2 - \omega_{2d} \\ &\quad - c(\sin(\theta) \operatorname{sech}(x_e)^2 \dot{x}_e + \cos(\theta) \dot{\theta} \tanh(x_e))) \end{aligned} \quad (8)$$

We choose the virtual inputs as α_z and α_y and define new states

ζ_1 and ζ_2 as the difference between the virtual inputs and their desired values α_{zd} and α_{yd} .

$$\begin{aligned} \alpha_z &= -v_1 \sin \theta + v_3 \cos \theta \\ \alpha_{zd} &= \dot{z}_d - k_z z_e \\ \alpha_y &= \omega_2 - c(\sin(\theta) \operatorname{sech}(x_e)^2 \dot{x}_e + \cos(\theta) \dot{\theta} \tanh(x_e)) \\ \alpha_{yd} &= \omega_{2d} - k_y y_e \\ \zeta_1 &= \alpha_z - \alpha_{zd} \\ \zeta_2 &= \alpha_y - \alpha_{yd} \end{aligned} \quad (9)$$

To incorporate the additional states, a new Lyapunov function and its derivative are defined as

$$\begin{aligned} V_A &= \frac{1}{2}(z_e^2 + y_e^2) + \frac{1}{2}(\zeta_1^2 + \zeta_2^2) \\ \dot{V}_A &= z_e(\zeta_1 - k_z z_e) + y_e(\zeta_2 - k_y y_e) + \zeta_1 \dot{\zeta}_1 + \zeta_2 \dot{\zeta}_2 \end{aligned} \quad (10)$$

The derivatives

$$\begin{aligned} \dot{\zeta}_1 &= \dot{\alpha}_z - \dot{\alpha}_{zd} \\ \dot{\zeta}_2 &= \dot{\alpha}_y - \dot{\alpha}_{yd} \end{aligned} \quad (11)$$

introduce $\dot{\omega}_2$, v_1 , and v_3 which are directly influenced by the inputs u_1 and u_2 . Using these to achieve stability allows the design of a controller which will be discussed later in the paper. If we choose

$$\begin{aligned} \dot{\alpha}_z &= \dot{\alpha}_{zd} - k_1 \zeta_1 \\ \dot{\alpha}_y &= \dot{\alpha}_{yd} - k_2 \zeta_2 \end{aligned} \quad (12)$$

the derivative of the Lyapunov function becomes

$$\dot{V}_A = z_e(\zeta_1 - k_z z_e) + y_e(\zeta_2 - k_y y_e) - k_1 \zeta_1^2 - k_2 \zeta_2^2 \quad (13)$$

By adding and subtracting the terms $(\frac{1}{4k_z} \zeta_1^2 + \frac{1}{4k_y} \zeta_2^2)$, it can be rewritten as

$$\begin{aligned} \dot{V}_A &= -k_z(z_e^2 - \frac{z_e \zeta_1}{k_z} + \frac{1}{4k_z^2} \zeta_1^2) - k_y(y_e^2 - \frac{y_e \zeta_2}{k_y} + \frac{1}{4k_y^2} \zeta_2^2) \\ &\quad - \zeta_1^2(-k_z + \frac{1}{4k_1}) - \zeta_2^2(-k_y + \frac{1}{4k_2}) \\ &= -\frac{1}{4k_z}(z_e - \frac{\zeta_1}{2k_z})^2 - \frac{1}{4k_y}(y_e - \frac{\zeta_2}{2k_y})^2 - \zeta_1^2(\frac{1}{4k_z} - k_1) \\ &\quad - \zeta_2^2(\frac{1}{4k_y} - k_2) \end{aligned} \quad (14)$$

From this it is easy to see that $\dot{V}_A < 0$ for $\frac{1}{4} < k_1 k_\theta$ and $\frac{1}{4} < k_2 k_\theta$. By Lyapunov's stability theorem, the system is asymptotically stable about the point $(z_e = 0, y_e = 0)$.

3.2 Synthesis of Controller

The controller design follows from the final steps of the stability analysis in the previous section. Using Eqs. (4) and (12) control laws are created for u_1 and u_2 . From Eq. (12) the derivatives of the virtual inputs are

$$\begin{aligned}\dot{\alpha}_z &= -\dot{v}_1 \sin \theta - v_1 \cos(\theta) \dot{\theta} + \dot{v}_3 \cos \theta - v_3 \sin(\theta) \dot{\theta} \\ &= -\dot{v}_1 \sin \theta + \dot{v}_3 \cos \theta + \gamma_z \\ \dot{\alpha}_y &= \dot{\omega}_2 - c(\sin(\theta)((\text{sech}(x_e))^2 \ddot{x}_e) - 2\dot{x}_e^2 \tanh(x_e) \text{sech}(x_e)^2 \\ &\quad - \dot{\theta}^2 \tanh(x_e)) + \cos(\theta)(\dot{\omega}_2 \tanh(x_e) + \dot{\theta} \text{sech}(x_e)^2 \dot{x}_e) \\ &= \dot{\omega}_2(1 - c \cos(\theta) \tanh(x_e)) - c \sin(\theta) \text{sech}(x_e)^2 (\dot{v}_1 \cos \theta \\ &\quad + \dot{v}_3 \sin \theta) + \gamma_y \\ \ddot{x}_e &= \dot{v}_1 \cos \theta - v_1 \sin(\theta) \dot{\theta} + \dot{v}_3 \sin \theta + v_3 \cos(\theta) \dot{\theta} - \ddot{x} \\ &= \dot{v}_1 \cos \theta + \dot{v}_3 \sin(\theta) + \gamma_x\end{aligned}\quad (15)$$

where γ_x , γ_y , and γ_z are the state-dependent terms that are not functions of the inputs. Plugging Eq. (4) into Eq. (15) and moving γ_y and γ_z to the left hand side of the equations yield

$$\begin{aligned}-(f_1 + g_1 u_1) \sin \theta + (f_3 + g_3 u_1) \cos \theta &= \dot{\alpha}_{zd} - k_1 \zeta_1 - \gamma_z \\ (f_2 + g_2 u_2)(1 - c \cos(\theta) \tanh(x_e)) - c \sin(\theta) \text{sech}(x_e)^2 ((f_1 + \\ g_1 u_1) \cos \theta + (f_3 + g_3 u_1) \sin \theta) &= \dot{\alpha}_{yd} - k_2 \zeta_2 - \gamma_y\end{aligned}\quad (16)$$

where g_i and f_i are short-hand for the corresponding functions in Eq. (4). Using these equations, the input can be calculated as

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \Psi_1 & \Psi_2 \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}\quad (17)$$

where

$$\begin{aligned}\Gamma_1 &= \frac{1}{(-g_1 \sin \theta + g_3 \cos \theta)} (\dot{\alpha}_{zd} - k_1 \zeta_1 - \gamma_z \\ &\quad + f_1 \sin \theta - f_3 \cos \theta) \\ \Gamma_2 &= \dot{\alpha}_{yd} - k_2 \zeta_2 - \gamma_y - f_2(1 - c \cos(\theta) \tanh(x_e)) \\ &\quad + c \sin(\theta) \text{sech}(x_e)^2 (f_1 \cos \theta + f_3 \sin \theta) \\ \Psi_1 &= -c \sin(\theta) \text{sech}(x_e)^2 (g_1 \cos \theta + g_3 \sin \theta) \\ \Psi_2 &= g_2(1 - c \cos(\theta) \tanh(x_e))\end{aligned}\quad (18)$$

We will denote this controller as c1 hereafter.

3.3 Synthesis of Controller Incorporating Input Constraints

To take control constraints into consideration, we take inspiration from [19] and [20] to introduce artificial state variables to retain information about the input saturation. The vector $[\chi_z, \chi_y]$ is generated by the difference between saturated inputs that are actually applied to the system and the inputs u_1 and u_2 generated by the controller. We desire to make use of the input saturation in such a way that the controller can provide more meaningful input to the system (e.g. calculate inputs within achievable limits) while still achieving acceptable performance. To design the new controller, let the following equations define the generation of $[\chi_z, \chi_y]$.

$$\begin{aligned}\dot{\chi}_z &= -k_{\chi_z} \chi_z - A(\hat{u}_1 - u_1) \\ \dot{\chi}_y &= -k_{\chi_y} \chi_y - \Psi_2(\hat{u}_2 - u_2) \\ A &= (-g_1 \sin \theta + g_3 \cos \theta)\end{aligned}\quad (19)$$

where $\hat{u}_1$ and $\hat{u}_2$ play the role of the saturated input. By solving for these, the aim is to design a controller that tends to compute inputs that lie within the saturation limits. In addition, let

$$\begin{aligned}\tilde{\alpha}_z &= \alpha_z - \chi_z \\ \tilde{\alpha}_y &= \alpha_y - \chi_y\end{aligned}\quad (20)$$

where $\tilde{\alpha}_z$ and $\tilde{\alpha}_y$ are virtual input errors. With this with the stability analysis proceeds as in Eqs. (8)-(10). In Eq. (11), it can be assumed that in practical implementation, an achievable input is applied to the system and let Eq. (11) be represented by

$$\begin{aligned}\dot{\zeta}_1 &= \dot{\tilde{\alpha}}_z - \dot{\alpha}_{zd} \\ \dot{\zeta}_2 &= \dot{\tilde{\alpha}}_y - \dot{\alpha}_{yd}\end{aligned}\quad (21)$$

Using the derivative of (20), a substitution is made to arrive at

$$\begin{aligned}\dot{\zeta}_1 &= \dot{\alpha}_z - \dot{\chi}_z - \dot{\alpha}_{zd} \\ \dot{\zeta}_2 &= \dot{\alpha}_y - \dot{\chi}_y - \dot{\alpha}_{yd}\end{aligned}\quad (22)$$

The choice

$$\begin{aligned}\dot{\alpha}_z &= \dot{\alpha}_{zd} + \dot{\chi}_z - k_1 \zeta_1 \\ \dot{\alpha}_y &= \dot{\alpha}_{yd} + \dot{\chi}_y - k_2 \zeta_2\end{aligned}\quad (23)$$

leads to the same results as the stability analysis of the original controller presented in the previous section. Control laws for $\hat{u}_1$ and $\hat{u}_2$ can now be designed using Eqs. (2), (19), and (23). Substituting Eqs. (2) and (19) into Eq. (23) yields

$$Au_1 - f_1 \sin(\theta) + f_3 \cos(\theta) + \gamma_z = \alpha_{zd} - k_1 \zeta_1 - k_{\chi_z} \chi_z - A(\hat{u}_1 - u_1) \quad (24)$$

$$\begin{aligned} & \Psi_1 u_1 + \Psi_2 u_2 + f_2(1 - c \cos(\theta) \tanh(x_e)) \\ & - c \sin(\theta) \operatorname{sech}(x_e)^2 (f_1 \cos \theta + f_3 \sin \theta) + \gamma_y \\ & = \alpha_{yd} - k_2 \zeta_2 - k_{\chi_y} \chi_y - \Psi_1(\hat{u}_1 - u_1) - \Psi_2(\hat{u}_2 - u_2) \end{aligned} \quad (25)$$

Substituting Γ_1 for u_1 and solving for $\hat{u}_1$ and $\hat{u}_2$ yields

$$\begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \Psi_1 & \Psi_2 \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} - \begin{bmatrix} \frac{k_{\chi_z} \chi_z}{(-g_1 \sin \theta + g_3 \cos \theta)} \\ k_{\chi_y} \chi_y \end{bmatrix} \quad (26)$$

where Ψ_1 , Ψ_2 , Γ_1 , and Γ_2 have the same definitions as Eq. (18). We will denote this controller as c2 throughout the rest of the paper.

It is worth noting that the design does not stop the controller from exceeding the input constraint values as they are not directly incorporated. It, instead, uses saturation as feedback to penalize the controller for generating control signals beyond the constraints and drives the difference between the controller generated inputs and saturated inputs to zero. For this reason, the new control laws for $\hat{u}_1$ and $\hat{u}_2$ are treated as a replacement for the original controller u_1 and u_2 in Eq. 20. If there is no occurrence of saturation, the controller acts as the original design c1 in Eq. (17).

3.4 Sliding Mode Observer Design

A challenging problem for the underwater gliders is to measure the linear velocities which are highly dispersed through out the dynamic equations. In order to implement the control design, a sliding mode observer based on results by Yaun et al. [21] is used to estimate the body-fixed velocities. This is done by first deriving the velocity v_z along the A_z axis described in Section 2 via the measurements of the depth z . This will be taken as the measurement v_z . A sliding function

$$s = \dot{z} - \dot{\hat{z}} = v_z - \hat{v}_z \quad (27)$$

is then defined, where $\hat{v}_z$ is the estimate of v_z . The sliding surface is guaranteed if $s\dot{s} < 0$. Note that

$$\begin{bmatrix} v_x \\ v_z \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = R_I^B \begin{bmatrix} v_1 \\ v_3 \end{bmatrix} \quad (28)$$

which implies inertial velocity dynamics become

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_z \end{bmatrix} = R_I^B \begin{bmatrix} \dot{v}_1 \\ \dot{v}_3 \end{bmatrix} + \dot{R}_I^B \begin{bmatrix} v_1 \\ v_3 \end{bmatrix} \quad (29)$$

Using this result, the dynamics of the inertial velocity estimate become a function of the body fixed velocity estimates and the measured state variables defined as

$$\begin{bmatrix} \dot{\hat{v}}_x \\ \dot{\hat{v}}_z \end{bmatrix} = R_I^B \begin{bmatrix} \hat{v}_1 \\ \hat{v}_3 \end{bmatrix} + \dot{R}_I^B \begin{bmatrix} \hat{v}_1 \\ \hat{v}_3 \end{bmatrix} + \begin{bmatrix} k_x \operatorname{sgn}(s) \\ k_z \operatorname{sgn}(s) \end{bmatrix} \quad (30)$$

With this, the sliding function dynamics become

$$\begin{aligned} \dot{s} &= -(\dot{\hat{v}}_1 + \tilde{v}_3 \omega_2) \sin \theta + (\dot{\hat{v}}_3 - \tilde{v}_1 \omega_2) \cos \theta - k_z \operatorname{sgn}(s) \\ \dot{\hat{v}}_1 &= f_1(\tilde{v}_1, \omega_2, \tilde{v}_3) + u_1 g_1(\theta) \\ \dot{\hat{v}}_3 &= f_3(\tilde{v}_1, \omega_2, \tilde{v}_3) + u_1 g_3(\theta) \end{aligned} \quad (31)$$

where $\tilde{v}_i = v_i - \hat{v}_i$, $i = 1, 3$. Choosing the gain

$$k_z > |-(\dot{\hat{v}}_1 + \tilde{v}_3 \omega_2) \sin \theta + (\dot{\hat{v}}_3 - \tilde{v}_1 \omega_2) \cos \theta| \quad (32)$$

ensures that s reaches the sliding surface. Following the results of Yaun et. al [21], invoking Phillipov's theory of equivalent dynamics on Eq. (30) and locally linearizing them around the inertial frame velocities yield.

$$\dot{\hat{v}}_x = \left(\frac{\partial f_x}{\partial v_x} - \frac{k_x}{k_z} \frac{\partial f_z}{\partial v_x} \right) \tilde{v}_x \quad (33)$$

where $(f_x, f_z) = (v_x, v_z)$. The convergence of v_x can be ensured by designing k_x such that $\frac{\partial f_x}{\partial v_x} - \frac{k_x}{k_z} \frac{\partial f_z}{\partial v_x} < 0$.

4 Simulation Results

Model parameters for simulation are shown in Table 1 with limits of ± 0.1 kg on m_0 and ± 7 mm on r_{p1} . Two scenarios are chosen to show the effectiveness of the proposed controller with (c2) and without (c1) consideration of saturation and compare them to a PID controller that was tuned with the matlab PID tuner and then refined through simulation runs to give good tracking performance. The PID controller is chosen as the baseline method over an LQR due to the fact that it is simple to implement, used in many existing gliders, and the fact that linear control laws based on local linearization have good performance only near the gliding equilibrium. [22]. The trajectory in both simulations are generated by a virtual copy of the vehicle using the same parameters as the actual vehicle.

Table 1. Simulation Parameters

Parameter	Value	Parameter	Value
m_1	8 kg	S	0.019 m ²
m_3	10.8 kg	$C_{M_P}^\alpha$	0.5665
$\bar{m}$	1.6 kg	C_{L0}	0.074606
J_2	0.08 m/s ²	C_L^α	0.45275
g	9.81 kg	C_{D0}	0.45275
r_{w3}	0.005 m	K_{q2}	-0.8
c	$\frac{\pi}{6}$ radians	C_{M_0}	0.0075719
ρ	997 kg/m ³		

4.1 Simulation 1: Mismatched Initial Conditions

The first scenario has the vehicle at an offset from the trajectory with no disturbances. The initial conditions of the state vector $[v_1(0), v_3(0), \omega_2(0), x(0), z(0), \theta(0)]$ for the vehicle and virtual copy are $[0.001, 0.001, 0, 0, 2, 0]$ and $[0.001, 0.001, 0, 0, 0, 0]$. The x - z path are plotted for the PID and the proposed controller with (c2) and without saturation considerations (c1) in Fig. (2). The evolution of the control is also plotted for each controller in Fig. (3) as well as the velocity estimates associated with c2. It can be seen that, after stabilizing z_e , the PID tracks θ_d fairly well and maintains a steady offset in x . c1 and c2 both converge to the path. Of course, they initially induce an error on the pitch angle to minimize the error x_e along the horizontal direction and then stabilizes the pitch error. The main difference between c1 and c2 is the control evolution. It can be seen that c2 produces less aggressive behavior, while achieving similar tracking performance. This is desirable since the rate of change of the inputs has direct implications in the energy expenditure of underwater gliders.

4.2 Simulation 2: Constant Disturbance

The second scenario keeps the same initial conditions, but adds a constant disturbance of -0.05 N to the drag D and -0.01 N on the lift L terms in the dynamic equations. It can be seen in Fig. (4) that the velocity estimation error is increased by noise. This slightly affect the performance of the c1 and c2 as they have small steady state offsets from and desired pitch angle. The PID on the other hand is diverging from the path, but tracks z_d and θ_d well. Fig. (5) shows similar results to that of simulation 1 as well as the ability of c1 and c2 to take advantage of the force to reduce control effort.

5 Conclusion and Future Work

In this paper the design of a backstepping-based trajectory tracking controller for underwater gliders was presented. In par-

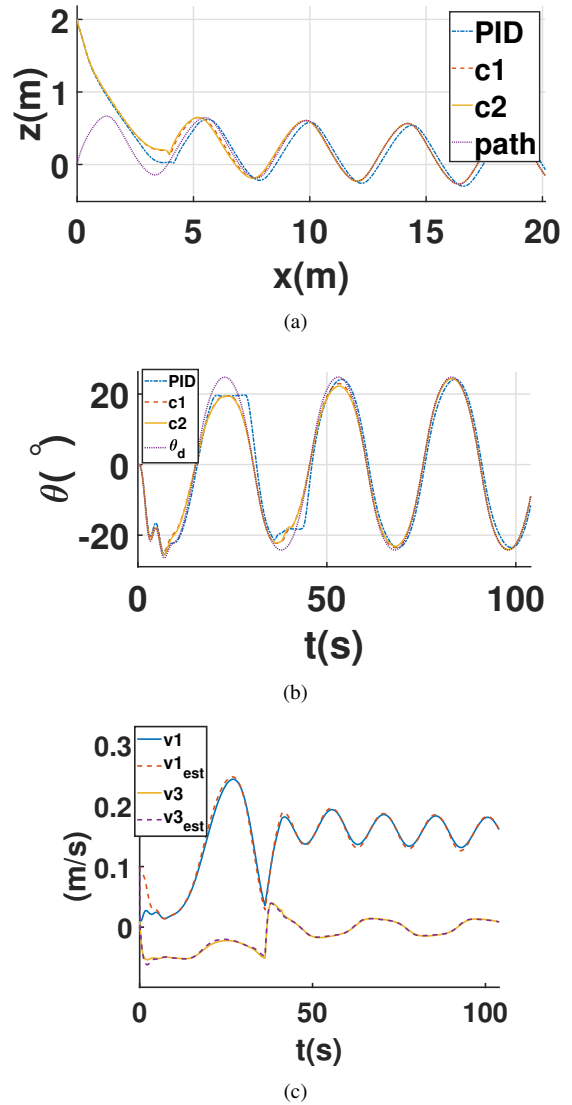
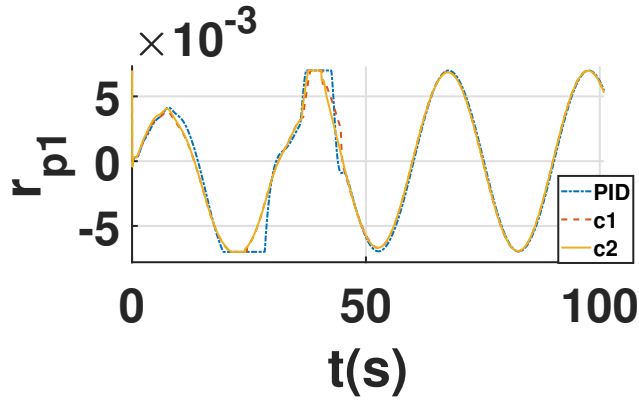
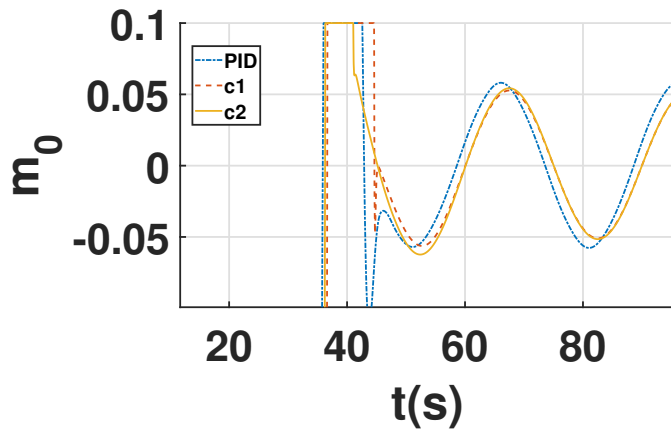


Figure 2. Resultant and reference trajectories of (a) xz-plane path and (b) pitch of the proposed controller, with (c2) and without (c1) saturation consideration, and the PID with mismatched initial conditions. (c) depicts the estimated and actual velocities.

ticular, a choice of a Lyapunov function that allowed the controller to achieve tracking of a sagittal plane position and heading was presented. The proposed controller is able to make use of the pitch angle to control both orientation and horizontal translation. This control scheme is applicable to both hybrid gliders and gliding robotic fish, where it can be used to mitigate energy consumption, and purely buoyancy-propelled gliders, where there is no direct control for horizontal velocity. A simple modification that uses saturation as feedback, producing a slightly less aggressive controller while achieving similar performance is also



(a)



(b)

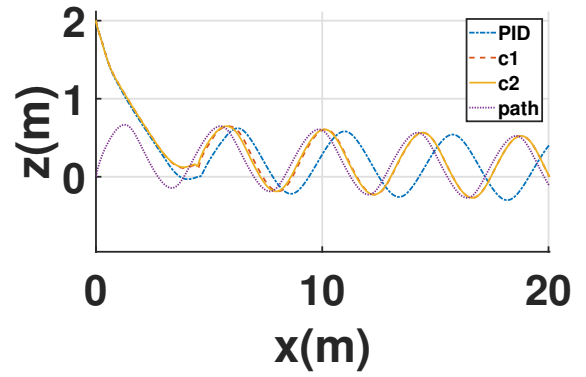
Figure 3. Control input (a) $u_2 (r_{p1})$ and (b) $u_1 (m_0)$ of the proposed controller with (c2) and without (c1) saturation consideration and PID during simulation 1.

presented. In this implementation, only magnitude saturation is considered by providing upper and lower limits. However, this scheme can also capture input saturation due to rate limits if the inputs can be measured.

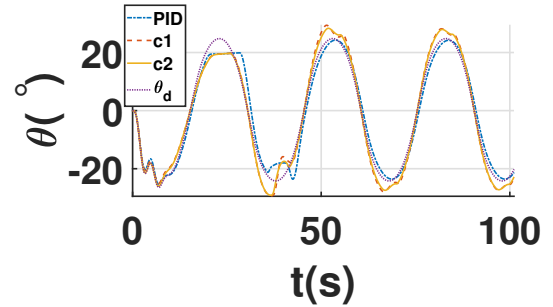
Future work will include estimating disturbances in the observer design, followed by estimating the x position via dead reckoning and sporadic surface measurements. Next, experimental validation of both the observer and controller designs will be performed. Finally, the design approach will be extended to tracking control of a glider in full 3D motion.

ACKNOWLEDGMENT

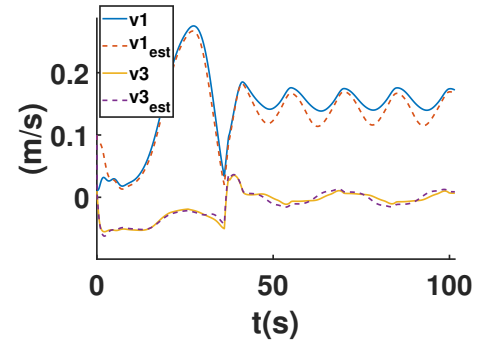
This work was supported in part by the National Science Foundation (ECCS 1446793, IIS 1715714).



(a)



(b)



(c)

Figure 4. Resultant and reference trajectories of (a) xz -plane path and (b) pitch of the proposed controller, with (c2) and without (c1) saturation consideration, and the PID with mismatched initial conditions and constant disturbance. (c) depicts the estimated and actual velocities.

REFERENCES

- [1] Stommel, H., 1989. "The slocum mission". *Oceanography*, **2**(1), pp. 22–25.
- [2] Sherman, J., Davis, R. E., Owens, W., and Valdes, J., 2001. "The autonomous underwater glider "spray"". *IEEE Journal of Oceanic Engineering*, **26**(4), pp. 437–446.
- [3] Sliwka, J., Clement, B., and Probst, I., 2012. "Sea glider guidance around a circle using distance measurements to

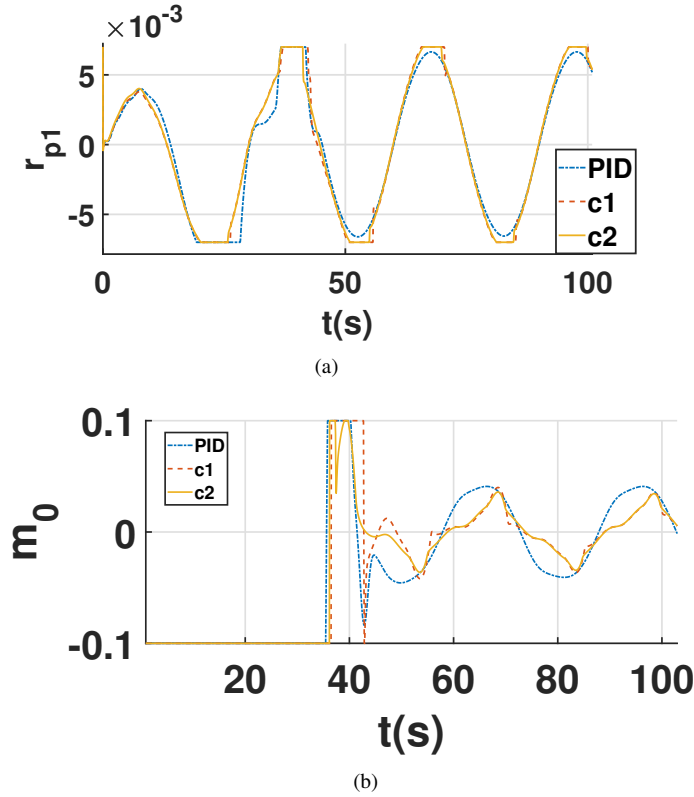


Figure 5. Control input (a) u_2 (r_{p1}) and (b) u_1 (m_0) of the proposed controller with (c2) and without (c1) saturation consideration and PID during simulation 2.

- a drifting acoustic source". In *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, IEEE, pp. 94–99.
- [4] Zhang, F., 2014. "Modeling, design and control of gliding robotic fish". Dissertation, Michigan State University. Electrical Engineering.
- [5] Wang, J., and Tan, X., 2013. "A dynamic model for tail-actuated robotic fish with drag coefficient adaptation". *Mechatronics*, **23**(6), pp. 659–668.
- [6] Zhang, F., Ennasr, O., Litchman, E., and Tan, X., 2016. "Autonomous sampling of water columns using gliding robotic fish: Algorithms and harmful-algae-sampling experiments". *IEEE Systems Journal*, **10**(3), pp. 1271–1281.
- [7] Ullah, B., Ovinis, M., Baharom, M. B., Javaid, M., and Izhar, S., 2015. "Underwater gliders control strategies: A review". In *Control Conference (ASCC)*, 2015 10th Asian, IEEE, pp. 1–6.
- [8] Mat-Noh, M., Arshad, M., Mohd-Mokhtar, R., and Khan, Q., 2017. "Control of an autonomous underwater glider using integral super-twisting sliding mode control (istsmc)". In *Underwater System Technology: Theory and Applica-*

- tions (USYS), 2017 IEEE 7th International Conference on, IEEE, pp. 1–6.
- [9] Yang, H., and Ma, J., 2010. "Sliding mode tracking control of an autonomous underwater glider". In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Vol. 4, IEEE, pp. V4–555.
- [10] Graver, J. G., and Leonard, N. E., 2001. "Underwater glider dynamics and control". In *12th International Symposium on Unmanned Untethered Submersible Technology*, pp. 1742–1710.
- [11] Leonard, N. E., and Graver, J. G., 2001. "Model-based feedback control of autonomous underwater gliders". *IEEE Journal of oceanic engineering*, **26**(4), pp. 633–645.
- [12] Mahmoudian, N., and Woolsey, C., 2008. "Underwater glider motion control". In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, IEEE, pp. 552–557.
- [13] Dong, E., Guo, S., Lin, X., Li, X., and Wang, Y., 2012. "A neural network-based self-tuning pid controller of an autonomous underwater vehicle". In *the Proceedings of the International Conference on Mechatronics and Automation*, Chengdu, China, pp. 5–8.
- [14] Isa, K., and Arshad, M. R., 2013. "Neural network control of buoyancy-driven autonomous underwater glider". In *Recent Advances in Robotics and Automation*. Springer, pp. 15–35.
- [15] Nag, A., Patel, S. S., and Akbar, S., 2013. "Fuzzy logic based depth control of an autonomous underwater vehicle". In *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, 2013 International Multi-Conference on, IEEE, pp. 117–123.
- [16] Zhang, F., Tan, X., and Khalil, H. K., 2012. "Passivity-based controller design for stablization of underwater gliders". In *American Control Conference (ACC)*, 2012, IEEE, pp. 5408–5413.
- [17] Cao, J., Cao, J., Zeng, Z., and Lian, L., 2016. "Nonlinear multiple-input-multiple-output adaptive backstepping control of underwater glider systems". *International Journal of Advanced Robotic Systems*, **13**(6).
- [18] Do, K. D., and Pan, J., 2009. *Control of Ships and Underwater Vehicles: Design for Underactuated and Nonlinear Marine Systems*. Springer Science & Business Media, ch. 6, pp. 109–124.
- [19] Farrell, J., Polycarpou, M., and Sharma, M., 2003. "Adaptive backstepping with magnitude, rate, and bandwidth constraints: Aircraft longitude control". In *American Control Conference*, 2003. Proceedings of the 2003, Vol. 5, IEEE, pp. 3898–3904.
- [20] Farrell, J., Polycarpou, M., and Sharma, M., 2004. "On-line approximation based control of uncertain nonlinear systems with magnitude, rate and bandwidth constraints on the states and actuators". In *American Control Conference*, 2004. Proceedings of the 2004, Vol. 3, IEEE, pp. 2557–

2562.

- [21] Yuan, J., Wu, Z., Yu, J., and Tan, M., 2017. “Sliding mode observer-based heading control for a gliding robotic dolphin”. *IEEE Transactions on Industrial Electronics*, **64**(8), pp. 6815–6824.
- [22] Do, K. D., Pan, J., and Jiang, Z.-P., 2004. “Robust and adaptive path following for underactuated autonomous underwater vehicles”. *Ocean Engineering*, **31**(16), pp. 1967–1997.