Numerical Modelling of an Underwater Micro Glider

1.0 Abstract

Underwater gliders are autonomous devices that have helped researchers collect environmental data on our oceans by performing relatively inexpensive, long-term missions. These gliders can be fitted with a range of measuring equipment, and then broadcast data back to be analysed. The aim of this project was to model a micro underwater glider that could effectively operate in lakes. The glider measures just **0.75m** long, weighs **3.9kg** and is powered by a buoyancy engine with a displacement of **±60ml**. The model shows the expected behaviour of the glider when it is programmed to travel to a relative point. Further to this behaviour, the model shows the user when the glider's route is calculated to be unsuitable, to prevent glider loss.

2.0 Introduction

This report presents the process undertaken to perform numerical modelling of the underwater glider using MATLAB. Initially, previous development of underwater gliders as well as future applications are covered to explain the motivation behind this project. Then the derivation of the equations, on which the MATLAB simulation is based, is discussed from first principles. Finally, the results of this process are discussed to provide insight on the success of the modelling.

3.0 Background

Initially designed in the early 1990's [1], an underwater glider is an Autonomous Underwater Vehicle (AUV) that uses a buoyancy engine, in the form of a gas bladder to vertically propel itself. Wings on the glider's hull create a lift force that pulls the glider forward through the water. This form of movement is very energy efficient, as alternating between a climbing and sinking trajectory requires very small displacement changes at relatively large time intervals. A single glider mission, following this triangular pattern, can last for many months while travelling several thousand kilometres [2]. Their long ranges and lower mission costs, considering both operational and production costs, make them ideal for collecting environmental data on large bodies of water. Currently, the main application for underwater gliders is ocean research, as greater depth means longer glide paths are possible without risking proximity with the seabed. However, this report focuses on the development of a micro glider suitable for use within a lake. This application allows for a smaller buoyancy engine but places a far greater restriction on maximum depth. The motivation behind using a micro glider within a lake is to allow researchers to gather a large dataset, of multiple points autonomously. Collecting large datasets currently requires researchers to manually place probes at multiple points and depths and then repeat this process over the course of weeks or months [3]. Placing a single glider with a preprogramed route that passes through all these points is clearly superior as it greatly reduces the valuable time required by researchers that could instead be used for analysis.

4.0 Method

To calculate the essential differential equations, all forces acting on the glider were resolved. These forces included drag (F_d), lift (F_L), weight (W) and buoyancy (F_b). In this case it has been assumed that all the forces act upon the centre of the glider.

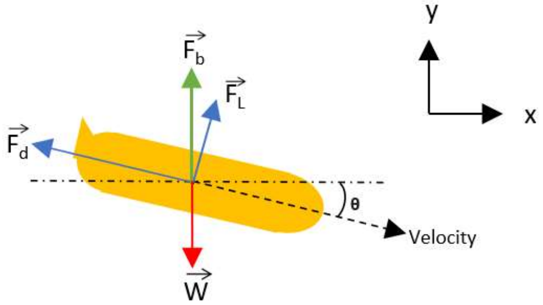


Figure 1 Glider Free Body Diagram

From the free body diagram in **Figure 1**, the forces were decoupled into the X and Y axis to produce equations 1 and 2.

$$F_L \sin(\theta) - F_d \cos(\theta) \tag{1}$$

$$F_b + F_L \cos(\theta) + F_d \sin(\theta) - W \tag{2}$$

When expanding the individual force terms within the resultant force equations above, velocity and theta remain unknown. Theta can subsequently be written in terms of velocity, by applying basic trigonometry.

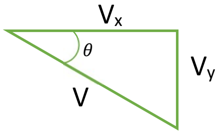


Figure2: Trigonometric relation

$$\theta = sin^{-1}\left(\frac{V_y}{V}\right) \tag{3}$$

$$\theta = cos^{-1}\left(\frac{V_x}{V}\right) \tag{4}$$

Substituting the force equations from the Appendix and setting $V_x = x'$, $V_y = y'$ and $V = \sqrt{V_x^2 * V_y^2}$ into equation 1 and 2, the equations 5 and 6 are derived.

$$\frac{1}{2} * \rho * A * Cl * y'\sqrt{x'^2 * y'^2} - \frac{1}{2} * \rho * A * Cd * x'\sqrt{x'^2 * y'^2} \tag{5}$$

$$\rho * g * Vol + \frac{1}{2} * \rho * A * Cl * x'\sqrt{x'^2 * y'^2} + \frac{1}{2} * \rho * A * Cd * y'\sqrt{x'^2 * y'^2} - m * g \tag{6}$$

Applying Newton Seconds Law and [4] simplifying equations 5 and 6, the two second order ODEs are obtained, given by equations 7 and 8.

$$x'' = \frac{1}{2m} * \rho * A\sqrt{x'^2 * y'^2} * (y' * Cl - x' * Cd) \tag{7}$$

$$y'' = \frac{\rho * g * Vol}{m} - g + \frac{1}{2m} * \rho * A\sqrt{x'^2 * y'^2} * (x' * Cl + y' * Cd) \tag{8}$$

To numerically solve the second order ODE equations, they are first transformed into four first order ODEs by considering the state vector **Z** (9) and then deriving the state variables. Equations 10, 11, 12 and 13 show these first order ODE equations.

$$z = \begin{pmatrix} x \\ y \\ x' \\ y' \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} \tag{9}$$

$$z_1' = x' = z_3 \tag{10}$$

$$z_2' = y' = z_4 \tag{11}$$

$$z_3 = x'' = \frac{1}{2m} * \rho * A\sqrt{{z_3}^2 * {z_4}^2} * (z_4 * Cl - z_3 * Cd) \tag{12}$$

$$z_4 = y'' = \frac{\rho * g * Vol}{m} - g + \frac{1}{2m} * \rho * A\sqrt{{z_3}^2 * {z_4}^2} * (z_3 * Cl + z_4 * Cd) \tag{13}$$

To output the trajectory of the glider, the four first order differential equations were added to a function called stateDeriv to define the respective displacements and velocities in each plane; the constants within these equations were also defined here. For this model, the glider required a triangular path with direction changes every **10** minutes which was simulated and added within the stateDeriv function. This function allows the user to input previous displacements, velocities, and a displacement volume to model a trajectory path.

The resultant vertical force acting on the glider may not be zero in all conditions, depending on the water density. The bladders maximum displacement may not be able to be utilised and then the glider would not perform as expected. The initial bladder volume (described as **offset**), to achieve neutral buoyancy was consequently calculated. The offset could then be used as an origin for displacement fluctuation, ensuring the glider would return to the surface between each of its dives.

The Runge Kutta-Method calculates four derivatives within each time step, which are then weighted, combined, and averaged. This significantly improves accuracy over the Euler method, which alternatively assumes the initial derivative as a constant over each time step. Therefore, the Runge Kutta-method was implemented with inputs of time, step size, previous derivatives, offset displacement and change in displacement from the bladder.

The trajectory of the glider could now be output, using the initial glider conditions. To enable the glider to compute the necessary bladder displacement to intersect a target coordinate, a final function was added, which implemented the shooting method. The shooting method allows for BVP problems to be solved by iteratively reducing the previous error values, starting with two initial guesses. The equations below describe this method.

$$Z_1 = Initial\ guess\ 1 \qquad \varepsilon_1 = Z_1 - Z_D$$

$$Z_2 = Initial\ guess\ 2 \qquad \varepsilon_2 = Z_2 - Z_D$$

$$Z_D = Desired\ Value$$

$$Z_{\mathrm{n+1}} = Z_2 - \varepsilon_2 \left(\frac{Z_2 - Z_1}{\varepsilon_2 - \varepsilon_1}\right) \tag{14}$$

Within the glider’s application, the initial guesses refer to the displacement variation, and the errors compare the vertical displacement to the desired intersection at the correct horizontal displacement. The error in displacement is likely never zero, as this could require infinite iterations of the shooting method, therefore a suitable error of **5mm** was selected. This error value was selected as the glider’s diameter is **50mm**, so would allow for the point to still be intersected. If this error is unattainable (the input coordinates could not be intersected) within 20 iterations of the shooting method, the loop is exited to ensure efficient simulation run time. If the errors were less than specified, the route is plotted and required displacement is output. The maximum allowable displacement within the bladder could be surpassed. Therefore, if the displacement value calculated is greater than the **60ml**, while considering the offset, an error message will appear. As the offset was a constant, the glider would only function within one water density. An initial iteration of the shooting method was utilised to allow for the glider to automatically calculate the offset required to achieve neutral buoyancy, allowing for functionality in a range of water densities.

To ensure the best route is selected, the shooting method is run for a range of initial guesses and the results are saved. Although most of these modelled trajectories should now be able to intersect any achievable target point, the most energy efficient route should be selected. To maximise this efficiency, the route with the least total water displaced by the buoyancy engine is chosen by comparing all routes that intersected the specified coordinates.

5.0 Results and Discussion

While developing the simulation, it was found that changing the timestep had a significant effect on simulation runtime and accuracy. This was investigated to ensure further simulations produce the best possible output for an appropriate simulation time. Maintaining all other simulation variables, the results of using three different time steps is shown below in **Table 1**. This shows that a **0.1s** timestep is best as it creates the most accurate result while completing the computation relatively quickly.

Time Step (s)	**Simulation Time (s)**	**Accuracy (m)**
1	2.414	0.0523
0.1	9.760	0.00132
0.01	93.94	0.00136

Table 1 Simulation Results with Varying Timesteps

Using the selected time step from above, a simulation was run to model the behaviour of the glider. Initial time, displacement and velocity were all zero and the simulation was run for **6000s**. The target point was at **(400, -20)** with the floor set to **-40m**. **Figure 3** shows the plot output by this simulation. It shows the glider successfully returns to the surface after each dive, meaning the offset was correctly calculated. Importantly, the glider appears to path through the target point whilst not colliding with the floor.

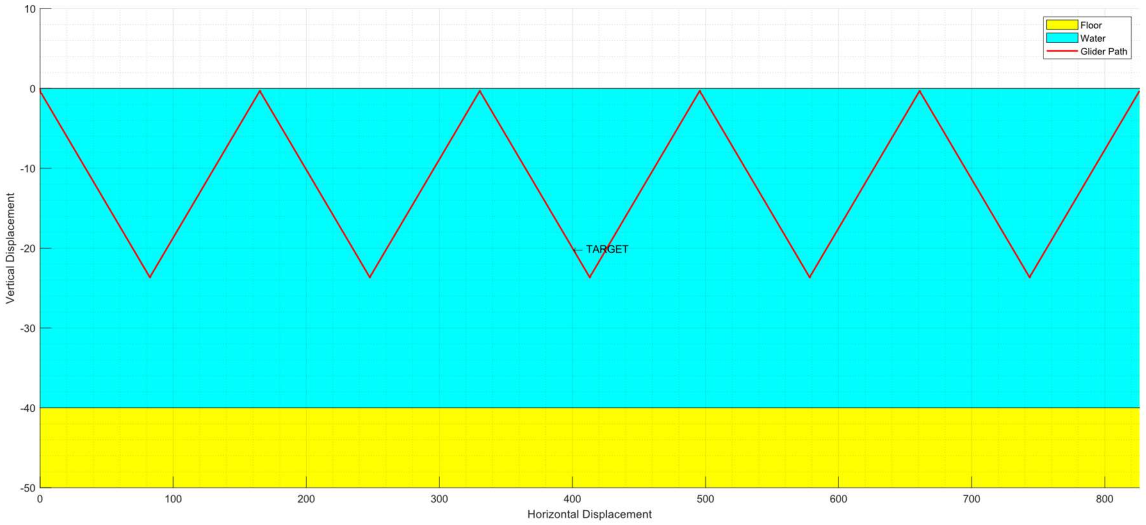
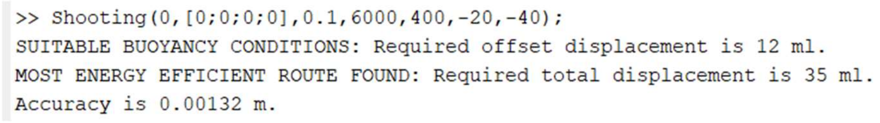


Figure 3 Glider Trajectory with Target (400, -20)

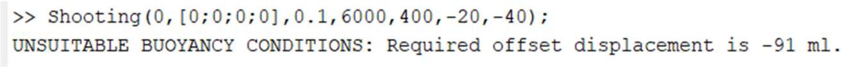
Figure 4 provides superior numerical detail and supports the above, showing the route was accurate to **0.00132m**. Another key detail shown here is displacement which is well within the glider's capabilities. These dual outputs make the results of the model very clear and ensures no misunderstanding is possible which, in a real-world situation, would result in the loss of the glider and any undelivered data.



```
>> Shooting(0,[0;0;0;0],0.1,6000,400,-20,-40);
SUITABLE BUOYANCY CONDITIONS: Required offset displacement is 12 ml.
MOST ENERGY EFFICIENT ROUTE FOUND: Required total displacement is 35 ml.
Accuracy is 0.00132 m.
```

Figure 4 Command Window Output for Successful Simulation

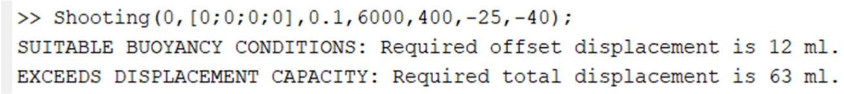
To increase code robustness and usability, non-ideal situations needed to be considered. For example, if the glider is placed in salt water where water density is around **1024kg/m^3** [6] the restricted displacement of the buoyancy engine would prevent the glider from becoming neutrally buoyant. The output of this simulation is seen in **Figure 5** below. The model will try to calculate an initial offset but, in this case, it stops after this step and warns the user.



```
>> Shooting(0,[0;0;0;0],0.1,6000,400,-20,-40);
UNSUITABLE BUOYANCY CONDITIONS: Required offset displacement is -91 ml.
```

Figure 5 Command Window Output for Unsuitable Conditions

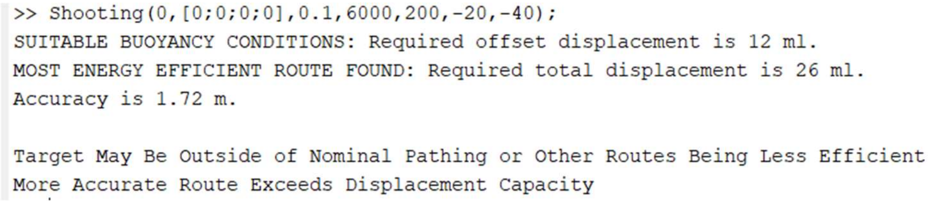
The next situation considered is when no route can be found to the point which is within the displacement limits. This would occur when the user has selected a target that is outside of the glider's maximum dive depth. If, unlike in **Figure 6**, the model did not warn the user, the glider would fail to reach its target depth and if the target had been set for the glider to dive under an obstacle, there would be a collision.



```
>> Shooting(0,[0;0;0;0],0.1,6000,400,-25,-40);
SUITABLE BUOYANCY CONDITIONS: Required offset displacement is 12 ml.
EXCEEDS DISPLACEMENT CAPACITY: Required total displacement is 63 ml.
```

Figure 6 Command Window Output for Displacement Exceeded

The model makes an initial route selection solely based on battery efficiency, which means it may select an inaccurate route. If the accuracy is greater than **1m** then the model will produce an output like that shown in **Figure 7**. This output will alert the user of the less-than-ideal accuracy of the route. If a viable, more accurate route can be found then it will be offered as a secondary solution to allow the user to select between the two routes. However, if no superior route can be found, as shown in **Figure 7**, then the user will still be alerted to make a choice. If accuracy is not of high importance for the mission, then this route would still be viable, and this step allows human risk analysis to be performed here.

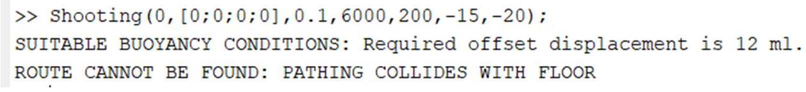


```
>> Shooting(0,[0;0;0;0],0.1,6000,200,-20,-40);
SUITABLE BUOYANCY CONDITIONS: Required offset displacement is 12 ml.
MOST ENERGY EFFICIENT ROUTE FOUND: Required total displacement is 26 ml.
Accuracy is 1.72 m.

Target May Be Outside of Nominal Pathing or Other Routes Being Less Efficient
More Accurate Route Exceeds Displacement Capacity
```

Figure 7 Command Window Output for Low Accuracy Route

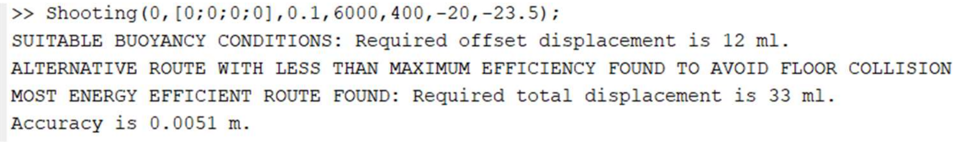
Operating in a lake environment can involve passing within proximity of the lakebed which is why the 'Floor' input was added to the model. If the glider path is calculated to touch or travel below the floor level, then the route is not displayed, and the user is warned as shown in **Figure 8**. If the glider was to strike the lakebed it would likely become damaged, or stuck, and be lost.



```
>> Shooting(0,[0;0;0;0],0.1,6000,200,-15,-20);
SUITABLE BUOYANCY CONDITIONS: Required offset displacement is 12 ml.
ROUTE CANNOT BE FOUND: PATHING COLLIDES WITH FLOOR
```

Figure 8 Command Window Output for Floor Collision

As the model initially selects the most battery efficient route, if a collision is expected, alternative routes are considered to avoid this. A superior route is not guaranteed but if one is found then the user will see the message shown in **Figure 9**. This allows the user to decide whether they want to carry out the mission with the secondary route or move to a different target.



```
>> Shooting(0,[0;0;0;0],0.1,6000,400,-20,-23.5);
SUITABLE BUOYANCY CONDITIONS: Required offset displacement is 12 ml.
ALTERNATIVE ROUTE WITH LESS THAN MAXIMUM EFFICIENCY FOUND TO AVOID FLOOR COLLISION
MOST ENERGY EFFICIENT ROUTE FOUND: Required total displacement is 33 ml.
Accuracy is 0.0051 m.
```

Figure 9 Command Window Output for Floor Collison Avoidance

6.0 Conclusions

As previously mentioned, underwater gliders are very effective vehicles for ocean research. Modelling for a micro glider in a lake environment increases the importance of some considerations that are currently applied for ocean gliders. The model produced allows for a range of inputs to be set, allowing maximum flexibility for the user. Coupling this with code robustness ensures that the model is very versatile.

7.0 References

1. Roberts, G. and Sutton, R., 2006. *Advances in Unmanned Marine Vehicles*. 69th ed. Stevenage: IET, pp.407-431.

2. J. Sherman, R. E. Davis, W. B. Owens, and J. Valdes, "The autonomous underwater glider "Spray"," in IEEE Journal of Oceanic Engineering, vol. 26, no. 4, pp. 437-446, Oct. 2001, doi: 10.1109/48.972076.

3. Sapna Sharma et al, A global database of lake surface temperatures collected by in situ and satellite methods from 1985-2009. Sci Data. 2015 Mar 17; 2:150008. doi: 10.1038/sdata.2015.8. PMID: 25977814; PMCID: PMC4423389.

4. Gundlach, J., Schlamminger, S., Spitzer, C., Choi, K., Woodahl, B., Coy, J. and Fischbach, E., 2007. Laboratory Test of Newton's Second Law for Small Accelerations. *Physical Review Letters*, 98(15).

5. Schaerer, Christian E. "The Shooting Method for the Solution of Ordinary Differential Equations: A Control-theoretical Perspective." International Journal of Systems Science 32.8 (2001): 1047-054. Web.

6. Sun, H., Feistel, R., Koch, M. and Markoe, A., 2008. New equations for density, entropy, heat capacity, and potential temperature of a saline thermal fluid. *Deep Sea Research Part I: Oceanographic Research Papers*, 55(10), pp.1304-1310.

7. W. Zihao, L. Ye, W. Aobo and W. Xiaobing, "Flying wing underwater glider: Design, analysis, and performance prediction," 2015 International Conference on Control, Automation and Robotics, 2015, pp. 74-77, doi: 10.1109/ICCAR.2015.7166005.

8. x. Moebs, W., J. Ling, S. and Sanny, J., 2016. University Physics Volume 1. Houston, Texas: OpenStax, pp.Section 14.4.

9. Moebs, W., J. Ling, S. and Sanny, J., 2016. University Physics Volume 1. Houston, Texas: OpenStax, pp.Section 5.4.

Appendix

Forces equations:

-Lift and drag equation [7]

$$F_L = \frac{1}{2}\rho V^2 C_L A$$

$$F_d = \frac{1}{2}\rho V^2 C_d A$$

Where:

V = Velocity

C_d and C_L = Coefficient of drag and lift

A = Area

p = Density

-Buoyancy equation [8]

$$F_b = \rho g Vol$$

Where:

Vol = Volume

g = gravity

p = Density

-Weight equation [9]

$$W = mg$$

m= Mass

g = Gravity