Position control for BlueROV2

Tudor Alinei-Poiană

January 2025

Contents

1 Introduction

This paper presents a strategy for controlling the position of a BlueROV2 (heavy configuration), along with a concise overview of the modeling approach. The document is organized as follows:

- Section 2 provides a brief explanation of the modeling procedure for the BlueROV2, highlighting challenges associated with mathematically characterizing such systems.
- Section 3 outlines the Ziegler-Nichols auto-tuning procedure applied to this system.
- Section 4 introduces the Fractional Order Ziegler-Nichols auto-tuning method.
- Section 5 presents the simulation results.
- Section 6 summarizes the findings and show possible future work directions.

2 BlueROV2 Heavy - model

The system used in this study is the BlueROV2 Heavy, presented in Figure 1. While the motion of a robot can be described using only a 3D rigid body motion model, for much accurate and realistic purposes, a dynamic model is considered because it can take into account all the forces and disturbances that act on the robot. The dynamic model for a 6Dof (Degrees of freedom), underwater ROV (Remotely operated vehicle) can be described using the Euler-Lagrange mathematical framework as presented in (1):

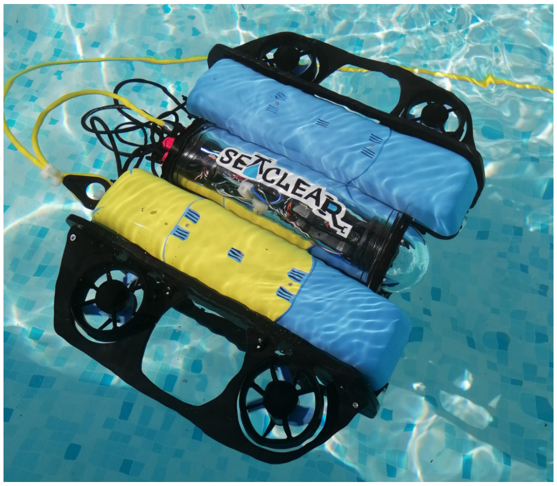


Figure 1: BlueROV2 Heavy Configuration

$$\dot{\eta} = J(\eta)v \tag{1}$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \tag{2}$$

Equation (1) describes the ROV's motion relative to the chosen reference coordinate system (kinematics), while (2) analyses the forces and moments induced to the ROV (kinetics).

2.1 Kinematic model

The states of interest for the system are:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; \Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}; v = \begin{bmatrix} u \\ v \\ w \end{bmatrix}; \omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3}$$

Considering the notation described in (3) we can say that the position and velocity of the ROV is described by:

$$\eta = \begin{bmatrix} p \\ \Theta \end{bmatrix} \text{ and } \tau = \begin{bmatrix} f \\ m \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ K \\ M \\ N \end{bmatrix} \tag{4}$$

Since Newtonian mechanics are presented in the body frame (denoted by the b) and the position of the ROV is described in a global reference frame, the forces that act on the robot will be described in the body frame, while the position of the submersible will be described in the global reference frame (usually the convention used being NED - n).
First, the transformation between the linear velocities described in body frame and NED (considering the roll-pitch-yaw convention for the Euler angles) is:

$$R_b^n = R_z R_y R_x \tag{5}$$

where

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \end{bmatrix} R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta & \\ 0 & 1 & 0 & \\ -\sin\theta & 0 & \cos\theta & 0 \end{bmatrix} R_z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Hence the linear velocities in NED is:

$$v^n = R_b^n v^b \tag{7}$$

Similarly, the transformation of angular velocities is given by:

$$\dot{\Theta} = T_\Theta \omega^b \tag{8}$$

where

$$T_\Theta = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{9}$$

Considering the equations described above we get:

$$\dot{\eta} = J(\eta)v \iff \begin{bmatrix} \dot{p} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} R_b^n & 0 \\ 0 & T_\Theta \end{bmatrix} \begin{bmatrix} v^b \\ \omega^b \end{bmatrix} \tag{10}$$

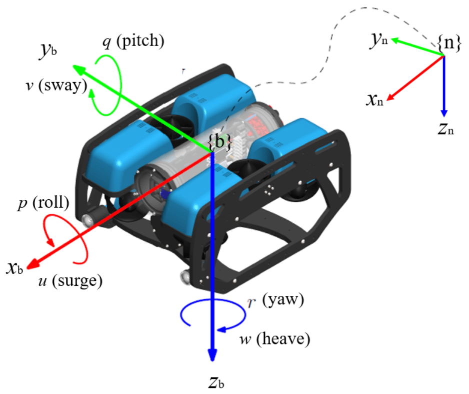


Figure 2: Transformation between body frame and NED frame

2.2 Kinetic model

The detailed kinetic model for an underwater ROV is:

$$M_{RB}\dot{v} + C_{RB}(v)\dot{v} + M_A\dot{v_r} + C_A(v_r)\dot{v} + D(v_w)v_w + g(\eta) = \tau \tag{11}$$

where: M_{RB} and M_A are the rigid body and added mass matrices, C_{RB} is the rigid body Coriolis and centripetal matrix induced by M_{RB} due to rotation of the local frame around NED frame. C_A is the added mass Coriolis and centripetal matrix induced by M_A due to the same rotation movement. $v_w = v - v_c$ is the relative velocity between the robot's velocity and waves.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_x & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_y & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{12}$$

Considering the origin of the body-frame in the geometric center of the ROV, we obtain symmetry in the XZ plane, thereby, this matrix can be simplified such that:

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & 0 \\ 0 & m & 0 & -mz_g & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -mz_g & 0 & I_x & 0 & 0 \\ mz_g & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \tag{13}$$

where $[x_g, y_g, z_g]$ is the position of the center of gravity relative to the geometrical center of the vehicle and is equal to $[0, 0, z_g]$.

Using the skew-symmetric cross-product operation on M_{RB} we obtain the Coriolis and centripetal matrix $C_{RB}(v)$:

$$C_{RB}(v) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & 0 \\ 0 & 0 & 0 & -mw & 0 & 0 \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & I_z r & -I_y q \\ -mw & 0 & -mu & -I_z r & 0 & I_x p \\ mv & -mu & 0 & I_y q & -I_x p & 0 \end{bmatrix} \tag{14}$$

The added mass matrix and added mass Coriolis and centripetal matrices are presented bellow

$$M_A = -\begin{bmatrix} X_{\dot u} & X_{\dot v} & X_{\dot w} & X_{\dot p} & X_{\dot q} & X_{\dot r} \\ Y_{\dot u} & Y_{\dot v} & Y_{\dot w} & Y_{\dot p} & Y_{\dot q} & Y_{\dot r} \\ Z_{\dot u} & Z_{\dot v} & Z_{\dot w} & Z_{\dot p} & Z_{\dot q} & Z_{\dot r} \\ K_{\dot u} & K_{\dot v} & K_{\dot w} & K_{\dot p} & K_{\dot q} & K_{\dot r} \\ M_{\dot u} & M_{\dot v} & M_{\dot w} & M_{\dot p} & M_{\dot q} & M_{\dot r} \\ N_{\dot u} & N_{\dot v} & N_{\dot w} & N_{\dot p} & N_{\dot q} & N_{\dot r} \end{bmatrix} \tag{15}$$

which can be simplified to:

$$M_A = -\begin{bmatrix} X_{\dot u} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot v} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\dot w} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{\dot p} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{\dot q} & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{\dot r} \end{bmatrix} \tag{16}$$

and

$$C_A(v) = \begin{bmatrix} 0 & 0 & 0 & 0 & z_{\dot w} w & 0 \\ 0 & 0 & 0 & -Z_{\dot w} w & 0 & -X_{\dot u} u \\ 0 & 0 & 0 & -Y_{\dot v} v & X_{\overset{:}{u}} u & 0 \\ 0 & -Z_{\dot w} w & Y_{\dot v} v & 0 & -N_{\dot r} r & M_{\overset{:}{G}} q \\ Z_{\dot w} w & 0 & -X_{\dot u} u & N_{\dot M} r & 0 & -K_{\dot p} p \\ -Y_{\dot v} v & X_{\dot u} u & 0 & -M_{\dot q} q & K_{\dot p} p & 0 \end{bmatrix} \tag{17}$$

The two matrices model the inertia of the fluid that moves with the body. Essentially, when the body accelerates, it has to move some fluid along with it, which effectively adds to its mass and moment of inertia. Because M_A and C_A matrices can be difficult to compute/identify and imprecise modeling of those matrices can lead to very poor performance in designing a controller for the ROV, they are omitted in the simulations. The added mass values depend heavily on the geometry of the body and its interaction with the surrounding fluid. Using incorrect or overly approximate values for M_A may lead to instability or poor performance in simulations, especially in high-speed or dynamic maneuvers. In low speed and small system, like in this case, it is acceptable to

neglect these terms.
There are four major sources causing hydrodynamic damping for a marine craft, including potential damping, wave drift damping, skin friction and damping due to vortex shedding. The effects of potential damping and wave drift damping are neglected for underwater vehicles. Subsequently, the ROV damping $D(v)$ can be approximated with a linear damping term caused by skin friction and a quadratic damping term.

$$D(v) = -\begin{bmatrix} X_u + X_{u|u|}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v|}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w|}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p + K_{p|p|}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q + M_{q|q|}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r + N_{r|r|}|r| \end{bmatrix} \tag{18}$$

In hydrostatics, the forces and moments acting on the ROV due to the gravitational and buoyancy forces are called restoring forces. Given that m is the mass of the vehicle, g is the acceleration of gravity, ρ is the water density and δ is the volume of fluid displaced by the ROV, the weight of the body W and buoyancy force B are determined by:

$$g(\eta) = \begin{bmatrix} (W-B)\sin\theta \\ -(W-B)\cos\theta\sin\phi \\ -(W-B)\cos\theta\cos\phi \\ z_g W\cos\theta\sin\phi \\ z_g W\sin\theta \\ 0 \end{bmatrix} \tag{19}$$

where:

$$W = mg \tag{20}$$

$$B = \rho g \delta \tag{21}$$

2.3 Thruster model and control allocation

In this subsection, we will define the model which describes the control force induced by the thrusters The BlueROV2 heavy has 8 thrusters which generate a total of 8 forces $F = [F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8]^T$ and have the same ammount of inputs $u = [u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8]^T$. Using the thruster technical sheet, Figures **??**, we can assume leaariTy between the input applied to each thruster and its output force such that:

$$F = Ku \tag{22}$$

where $K = diag(K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8)$ is the thrust coefficient that has all the diagonal elements equal (we are using identical thrusters).
Given the force vector $f = [F_x, F_y, F_z]$ and moment arms $r = [I_x, I_y, I_z]$, the

forces in a 6Dof can be determined by:

$$\tau = \begin{bmatrix} f \\ r \times f \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z l_y - F_y l_z \\ F_x l_z - F_z l_x \\ F_y l_x - F_x l_y \end{bmatrix} \tag{23}$$

Hence, considering BlueROV2 Heavy with 8 thrusters, the generalised forces and moments in 6 Dof due to 8 thrusters in terms of control inputs u can be then modelled as:

$$\tau = T(\alpha)F = T(\alpha)Ku \tag{24}$$

where $T(\alpha)$ is the thrust configuration matrix and α is the thrust rotation angle.

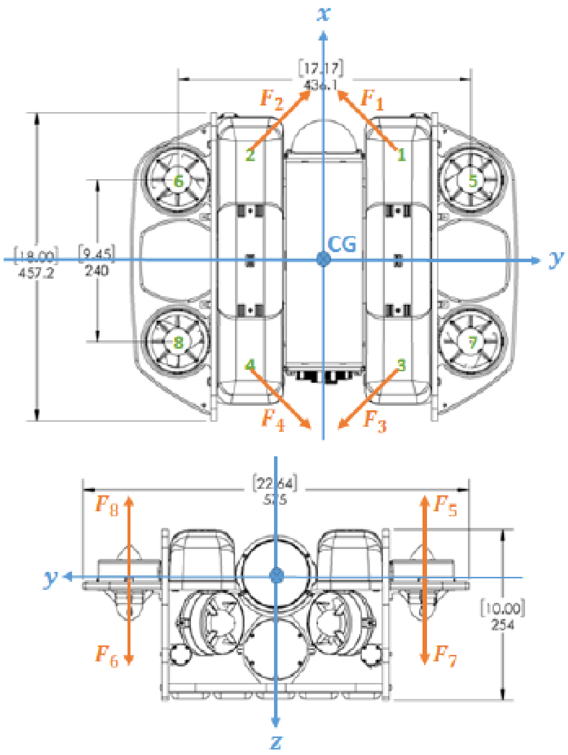


Figure 3: Thrust configuration (BlueRobotics 2018b)

The linear forces can be computed by projecting the force vector of each thruster on the ROV body reference frame, while for the moment arms we can use the technical specifications provided in the BlueROV2 technical sheet.

$$\tau = T(\alpha)F = \begin{bmatrix} 0.707 & 0.707 & -0.707 & -0.707 & 0 & 0 & 0 & 0 \\ -0.707 & 0.707 & -0.707 & 0.707 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 \\ 0.06 & -0.06 & 0.06 & -0.06 & -0.218 & -0.218 & 0.218 & 0.218 \\ 0.06 & 0.06 & -0.06 & -0.06 & 0.120 & -0.120 & 0.120 & -0.120 \\ -0.1888 & 0.1888 & 0.1888 & -0.1888 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \end{bmatrix} \tag{25}$$

Control allocation computes the control input signal u to apply to the thrusters such that we obtain the overall control forces τ. We can obtain this relation by left-multiplying (24) with $K^{-1}T^{-1}$. Because T is not a square matrix, we can use the pseudo-inverse matrix (denoted by the $*$ superscript) in order to achieve the same result. Thus, we get:

$$u = K_{-1}T^{*}\tau \tag{26}$$

The overall model used in simulation has the following simulink schematic:

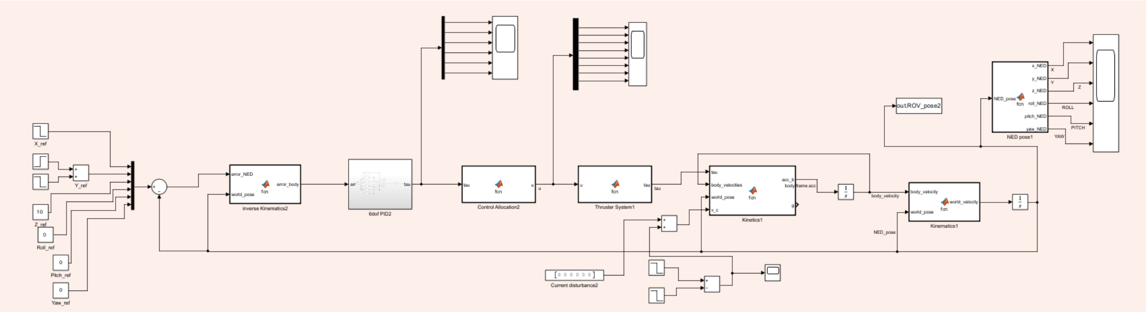


Figure 4: Simulink scheme

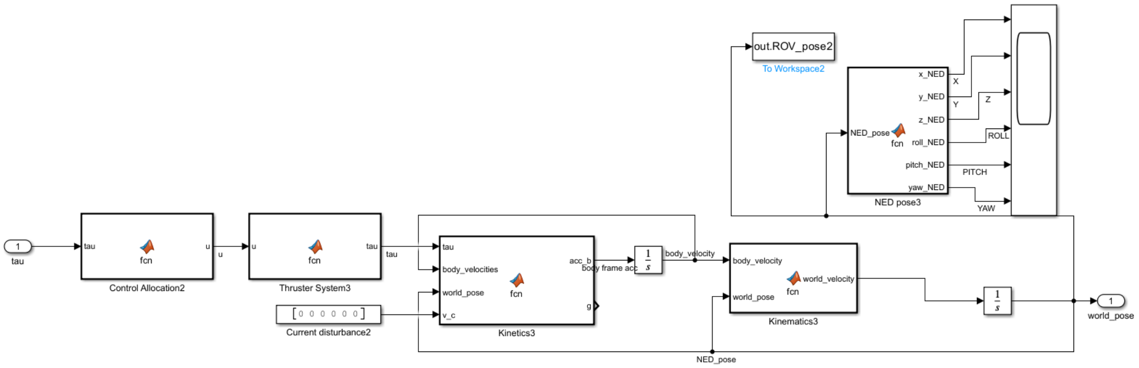


Figure 5: Simulink scheme

In order to control the the position of the ROV, we will consider that the system is decoupled (or the influence of the other states to the current controlled state is minimal) and we will tune 6 PIDs in order to control the position $[x^n, y^n, z^n]$ and the orientation $[\phi, \theta, \psi]$.

3 Ziegler-Nichols control method

Because the Ziegler-Nichols autotuning method is designed for linear systems and our system is highly nonlinear, we will design the controllers around a predefined working point $[0, 0, 10, 0, 0, 0]$ and we will perform a relay test for each of the states. The relay test command value will be kept low in order to avoid reaching out of the linearity zone, as well as the on-off switching value. The parameters of the controllers are presented Table 1.

	Surge	Heave	Sway	Roll	Pitch	Yaw
K_p	11.6366	0.9057	0.3833	0.1028	1.1471	0.1274
K_i	0.1869	0.1231	0.0255	0.0080	0.8657	0.0111
K_d	1.8069	4.4367	3.8433	0.8775	1.0122	0.9761

Table 1: PID parameters

4 Fractional Order Ziegler-Nichols

The same procedure was applied as in the previous section, in order to tune the PIDs. For the FOZN method, a Matlab script was developed in order to compute the fractional order controller parameters. The script receives the order of integration λ, the desired ration between the integrator time constant and derivative time constant, the critical gain and the period of oscillation corresponding to the computed critical gain. Bellow is described the code that computes the parameters of a FO-PID controller using this method:

```
function [kp, Ti, Td] = FOZN_PID(lambda, r, kc, Tc)

R = r^lambda;
C = cos(lambda*pi/2);
S = sin(lambda*pi/2);
ba_ratio = 0.4667;

X = roots([0.467*C-S, 0.467*R, R*(0.467*C+S)]);
X = X(find(X > 0));

alpha = 0.6 / (1 + C*(X/R + 1/X));
beta = X / ((2*pi)^lambda);
gamma = 1 / R;

kp = alpha * kc;
Ti = beta * Tc^lambda;
Td = gamma * Ti;

end
```

Listing 1: FOZN PID Controller Function

The parameters of the controllers are presented Table 2.

	Surge	Heave	Sway	Roll	Pitch	Yaw
K_p	1.5115	0.7577	1.0616	0.0860	0.9595	0.1066
T_i	9.0231	8.0876	16.0251	10.1005	4.0745	9.6726
T_d	4.7399	4.2485	3.4736	5.3058	2.1404	5.0811
λ	0.43	0.45	0.95	0.4	0.52	0.4

Table 2: PID parameters

5 Results

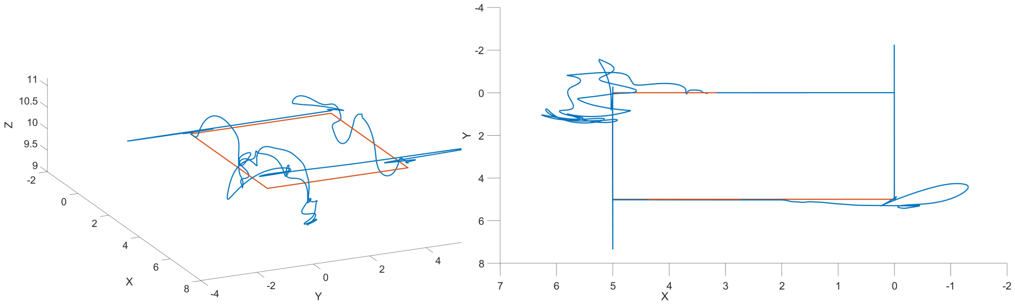
For each control method, we considered two simulation scenarios:

- The stabilization and zero steady state error, verified by applying a sequence of step inputs which form a square trajectory;

- The tracking scenario, In which sine and cosine waves are used as reference signals for the X and Y coordinates, respectively, describing a circular trajectory for the ROV.

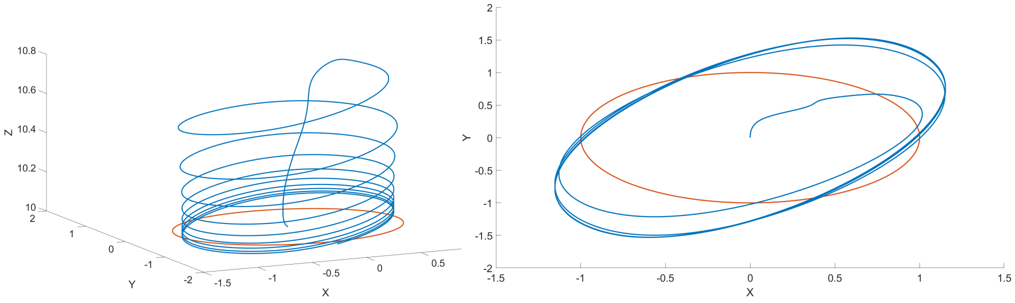
The controllers were also tested under conditions of command saturation. However, the integer-order Ziegler-Nichols PIDs struggled to handle this nonlinearity, resulting in numerical issues that caused instability during the simulation. In contrast, the FO-PIDs did not experience saturation in the command signals, as the values generated by the controllers remained within the desired limits, provided the system operated within the linearity zone. The control performance was also tested in the presence of disturbances (waves with a certain speed). Regardless of the scenario, the FOZN-PIDs showed better control performance than the regular ZN-PIDs as shown in **??**:

5.1 Integer order ZN results



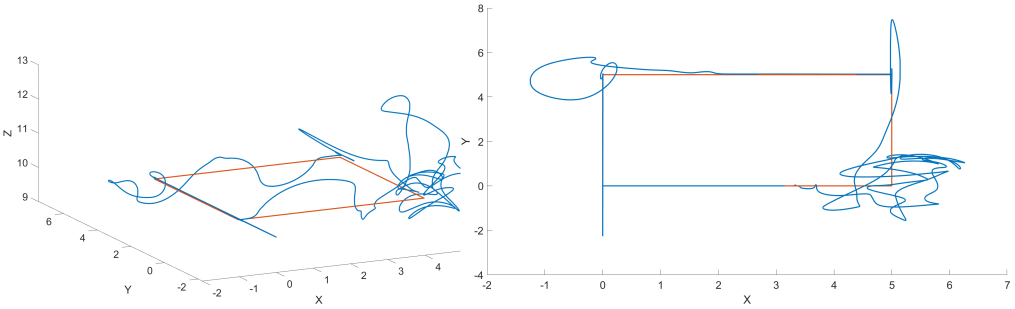
(a) Square trajectory - 3D view (b) Square trajectory - 3D view

Figure 6: Square trajectory with no disturbance



(a) Circular trajectory - 3D view (b) Circular trajectory - 3D view

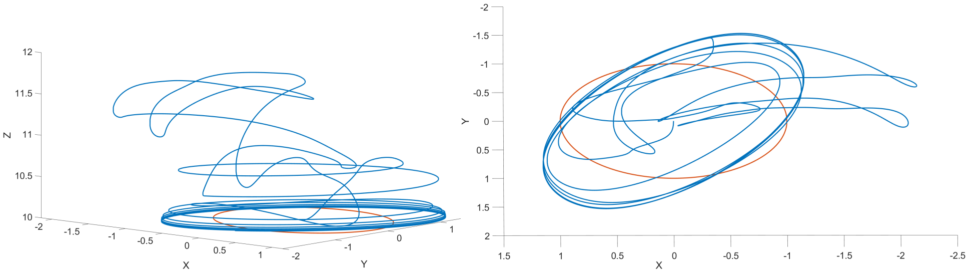
Figure 7: Circular trajectory with no disturbance



(a) Square trajectory - 3D view

(b) Square trajectory - 3D view

Figure 8: Square trajectory with disturbance

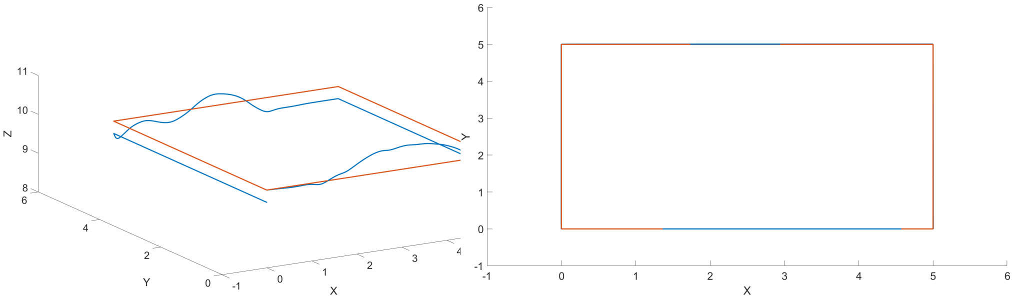


(a) Circular trajectory - 3D view

(b) Circular trajectory - 3D view

Figure 9: Circular trajectory with disturbance

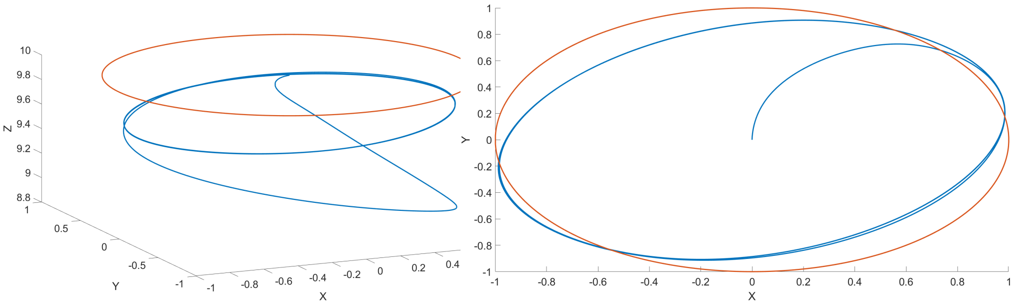
5.2 Fractional order ZN results



(a) Square trajectory - 3D view

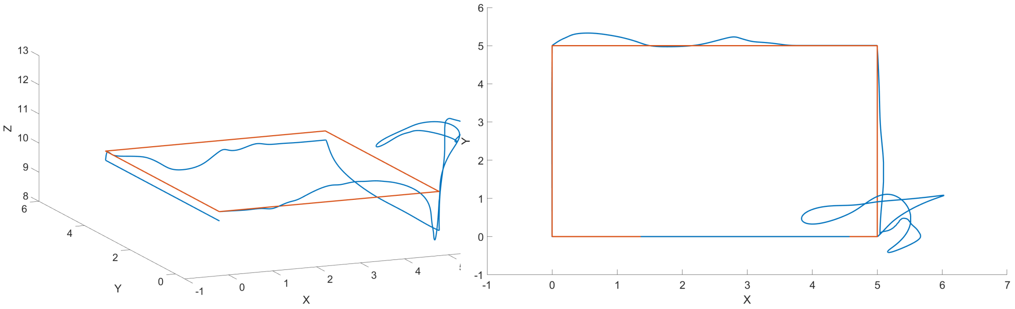
(b) Square trajectory - 3D view

Figure 10: Square trajectory with no disturbance



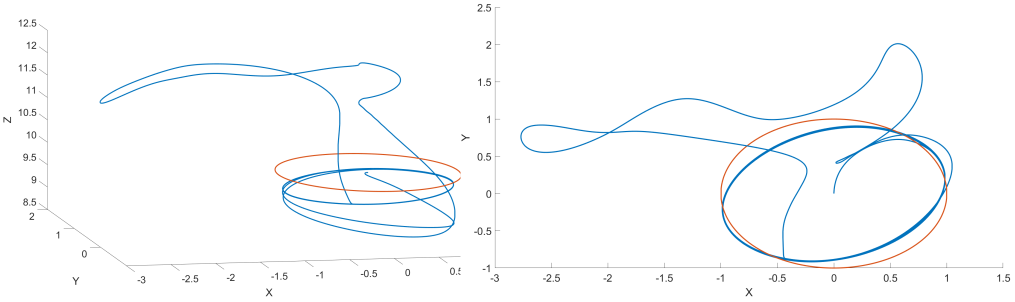
(a) Circular trajectory - 3D view (b) Circular trajectory - 3D view

Figure 11: Circular trajectory with no disturbance



(a) Square trajectory - 3D view (b) Square trajectory - 3D view

Figure 12: Square trajectory with disturbance



(a) Circular trajectory - 3D view (b) Circular trajectory - 3D view

Figure 13: Circular trajectory with disturbance

6 Conclusions and Future Work

The fractional-order Ziegler-Nichols autotuner demonstrated superior performance compared to its integer-order counterpart, especially in terms of disturbance rejection, tracking accuracy, and reference following. This advantage made it particularly well-suited for the complex dynamics of controlling an underwater ROV position.

However, since both controllers are based on linear control algorithms, their performance deteriorates significantly as the system moves further from the linearity zone. In this application, the departure from the linearity zone can be explained as follows: as the desired position of the ROV moves further away from the reference point $[0, 0, 10, 0, 0, 0]$, the controllers increase the command to maintain the performance guarantees, resulting in larger reference forces. This, in turn, leads to higher accelerations, which—even over a short period—cause significant displacements in both orientation and translation. The controllers will attempt to correct these displacements with increasingly aggressive commands, potentially destabilizing or damaging the system.

This limitation can be mitigated, in future works, by integrating feedback linearization with these classical approaches. A potential drawback of this method is that discrepancies between the model and the actual system may lead to suboptimal control performance, therefore, it is crucial to focus on enhancing the accuracy of the system model.

References

[1] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control.* 04 2021.