IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 25, 2020, accepted October 26, 2020, date of publication November 10, 2020,
date of current version November 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037122

Implementation of Nonlinear Adaptive U-Model Control Synthesis Using a Robot Operating System for an Unmanned Underwater Vehicle

NUR AFANDE ALI HUSSAIN[1], (Member, IEEE),
SYED SAAD AZHAR ALI[1], (Senior Member, IEEE), PERE RIDAO[2], (Member, IEEE),
PATRYK CIESLAK[2], AND UBAID M. AL-SAGGAF[3], (Member, IEEE)

[1]Center for Intelligent Signal and Imaging Research (CISIR), Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia
[2]Girona Underwater Vision and Robotics, University of Girona, 17004 Girona, Spain
[3]Center of Excellence in Intelligent Engineering Systems, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Syed Saad Azhar Ali (saad.azhar@utp.edu.my)

This research was supported by the Yayasan Universiti Teknologi Petronas-Fundamental Research Grant-Petroleum Resource Fund (YUTP-FRG) funded by PRF under the grant 0015LC0-235, Universiti Teknologi PETRONAS, Malaysia and King Abdulaziz University, Jeddah, Saudi Arabia under the grant 015ME0-066.

ABSTRACT This paper presents the development of unmanned marine robotic control modelling and control synthesis using a coupled multivariable underactuated nonlinear adaptive U-model approach. The proposed controller was developed using thru an open source robot operating system (ROS) platform. The new adaptive coupled U-model based internal model control (IMC) node was successfully developed and tested. The proposed controller demonstrated the simplicity of the control synthesis process and the implementation of the mathematical algorithm in real-time. The controller was compared with the proven existing GIRONA 500 UUV for real-time performance. The ROS environment provides fast and reliable controller design and development compared to conventional software architecture. Simulation and real-time experiment were conducted using ROS via the GIRONA 500 UUV platform and compared with a PID mission controller. A new ROS node of nonlinear adaptive U-model based IMC was developed using ROS. The results showed good control signal convergence and tracking performance between the plant or system model with the proposed method.

INDEX TERMS Adaptive control, nonlinear, UUV, underactuated, ROS.

I. INTRODUCTION

Most of the autonomous unmanned underwater vehicle (UUV) control systems are multivariable. The control framework methodology can be varied based on the application. There are no precise control methodology strategies that can be implemented in UUV applications. There are a few methodologies for the UUV control framework plan in the literature which attempt to demonstrate the practicality of the control algorithm. Most of the platforms are multiple-input multiple output (MIMO) due to the installation of multiple sensors and transducers onboard. Thus, controller design and development shall include the real-time implementation approach. However, a complex mathematical approach may lead to higher computation demand and energy needs thus making it difficult to be implemented in real-time [1]. With different designs and shapes, UUV are highly nonlinear, underactuated or with a limited control actuator, highly coupled in motion and vulnerable to external disturbances. The underactuated system needs to overcome the nonlinearity of the dynamics plus the disturbances and uncertainties arising from the marine environment [2]–[5]. Unfortunately, there is no such precise approach for nonlinear control and modelling frameworks in the presence of external disturbances especially in real-time implementation. In significant cases, the control framework outline strategies for nonlinear control systems vary from one framework to another. A few methodologies are described for nonlinear control framework plans in the literature which attempt to represent nonlinear control systems. It is desirable to have a control structure that is capable of overcoming external disturbances without involving a complex mathematical and modelling approach

The associate editor coordinating the review of this manuscript and approving it for publication was Min Wang.

that can be demonstrated not only through simulation but in real-time implementation as well. A reference-based model approach such an internal model control (IMC) structure can be implanted for a control strategy without any predefined precise mathematical modelling, plus it includes a disturbance rejection capability. The underactuated trajectory control problem of an UUV in the presence of environmental disturbances was proposed by [6]. This scheme implemented the barrier Lyapunov function for control law combined with a backstepping algorithm, and a radial basis function neural network (RBFNN) with dynamic surface control for solving the effects of external disturbances. The proposed method was compared with the quadratic Lyapunov function and produced better tracking results via simulation. Another underactuated trajectory control approach for an AUV under perturbations was achieved using adaptive sliding mode control [7]. The sliding mode adaptive control was compared to the generalised super-twisting algorithm (GSTA) sliding mode via real-time implementation using a Leonard ROV. The adaptive sliding mode method converged onto the desired input better than the GSTA approach with a reduced root-mean-square error value. Another control approach for the ROV platform used a double-loop sliding mode control for improvement of chatter reduction and to lower the root mean square error value [8]. Those authors introduced a novel switching term to replace the conventional switching term for sliding mode control through simulation without real-time implementation. Several other works employed the sliding mode control approach for an UUV such as in [9], [10] through simulation and real-time implementation. An adaptive fuzzy sliding mode controller was proposed to estimate the nonlinearity of MIMO underwater vehicles in [11] through a simulation approach. This method was the combination of two different nonlinear controllers aimed to counter the dynamics and chatter effect from the sliding mode control and also from external disturbances. The fuzzy logic control approximates the couple dynamics of the UUV. However, this method needs to be integrated with the sliding mode controller for stability analysis and the systematic control design method. The controller design was more complex and had a higher computational demand. Fuzzy logic control could reduce the chatter effect caused by the discontinuity of the sliding mode control system excited by unmodelled dynamics. In [4], a trajectory 3-D tracking control system was addressed for an UUV with a prescribed performance under model uncertainties and external disturbances through simulation and real-time implementation. Control tracking with a specified performance converged to the desired values with minimal error. However, only specified performances were presented for torpedo-like MIMO vehicles and unicycle-like MIMO vehicles.

It is desirable to have a control structure that is capable of overcoming external disturbances without involving a complex mathematical and modelling approach. Other work was done by [12] using two degrees of freedom IMC-PID with logarithmic approximations for the load frequency controller of a power system. The inverse component made use of a low pass filter to reject any load disturbances. The control structure required another feedback controller to counteract the effects of befuddling or mismatching. The results showed good performance in terms of error analysis, settling time and disturbance rejection. Based on the literature, a variety of approaches are available when designing a nonlinear controller for an unmanned marine robotics platform, especially for the underactuated model and these approaches can be highly robust against external disturbances.

As mentioned in the literature, the nonlinear modelling approach is a vital solution to overcome the nonlinearity, variation of the dynamics and hydrodynamics, underactuated, and external disturbances in a marine environment. Several strategies have been implemented to overcome these problems by implementing hybrid control approaches such as fuzzy logic control combined with sliding mode control [11], [13]–[15]. Other approaches augment the controller law by incorporation with the nonlinear neural network method and new novel approaches [16], [17]. These approaches will ensure the nonlinear effects and the disturbances can be reduced, thus improving the controller performance. A U-model has been proposed by [18] for a nonlinear dynamical control design for the Hammerstein model, a laboratory-scale level system and a continuous stir reactor. The control law was formulated as the current control term $u(t)$ which would be suitable for a nonlinear control application to reduce mathematical complexity. The U-model is an adaptive control-orientated model and it is based on a control input signal. It is more general compared to other estimation approaches and exhibits a polynomial structure applicable to the control term. This makes it suitable for real-time applications. Our previous works in [19] demonstrate the effectiveness of U-model modelling and the control approach in unmanned marine robotics applications through simulation.

This paper implements a nonlinear multivariable adaptive control synthesis using the U-model for nonlinear UUV applications in the 3D ROS visualisation tool GUI for the robotic platform simulation in a targeted environment such as an underwater scenario with sensor simulation using the GIRONA 500 UUV platform as in Fig. 1. The ROS simulation environment enables the controller to be analysed and verified before a real-time experiment can be conducted in the underwater environment. The controller was compared with the proven existing GIRONA 500 UUV controller for the validation process.

II. APPROACH AND METHODS

A. NONLINEAR INTERNAL MODEL CONTROL APPROACH

An Internal Model Control (IMC) is selected as a nonlinear modelling control framework. Fig. 2 shows an online adaptive IMC basic principle. IMC approach popular in process control applications and effective with linear plants [20]. It also can be used in nonlinear modelling and control applications such in [20]–[25]. It has good disturbance rejection capabilities and reduced the UUV controller design and development

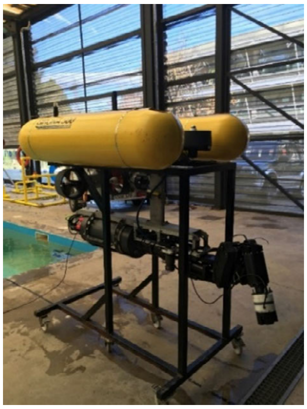


FIGURE 1. GIRONA 500 UUV.

complexity due to the uncertainty in the underwater environment.

The IMC control structure consists of a reference input $R(t)$, nonlinear system G_S, a nonlinear model of G_M, the controller as the inverse of G_M in the presence of unknown external disturbance $D(t)$ and output of the nonlinear system $Y(t)$. The output $Y(t)$ is compared with the model output $Y_M(t)$ to calculate the error $E(t)$. The closed loop system of the IMC control scheme can be expressed by [26]:

$$Y(t) = \frac{(R(t) - D(t))\, G_M^{-1} G_S}{1 + (G_S - G_M)\, G_M^{-1}} + D(t) \tag{1}$$

Rearranging Eq. (1), we get,

$$Y(t) = \frac{G_s Y_M^{-1} R(t) + (1 - Y_M^{-1} Y_M) D(t)}{1 + (G_s - Y_M) Y_M^{-1}} \tag{2}$$

Eq. (2) shows that according to IMC scheme, if the system can be truly modelled and the model can be inverted, it is possible to reject the disturbances, and output tracking is achieved. Therefore, the IMC tracking problem transforms to identifying a true model and its inverse. Recently, U-model has been used for certain process control, robotics and other nonlinear SISO and MIMO applications. U-model represents simplified NARMAX nonlinear polynomial model that can be inverse according to the IMC scheme. Therefore, this paper focuses on MIMO U-model based methodology for underwater marine applications.

B. COUPLED UNDERACTUATED MIMO NONLINEAR ADAPTIVE U-MODEL CONTROL SYNTHESIS APPROACH

For coupled and underactuated system, we establish a MIMO U-model by considering a multivariable nonlinear coupled system that can be represented by n-input $\times p$-output multivariable NARMAX model as:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix} = \begin{pmatrix} f_{y_1}(y_1, \ldots, y_p, u_1, \ldots, u_n, d) \\ f_{y_2}(y_1, \ldots, y_p, u_1, \ldots, u_n, d) \\ \vdots \\ f_{y_p}(y_1, \ldots, y_p, u_1, \ldots, u_n, d) \end{pmatrix} \tag{3}$$

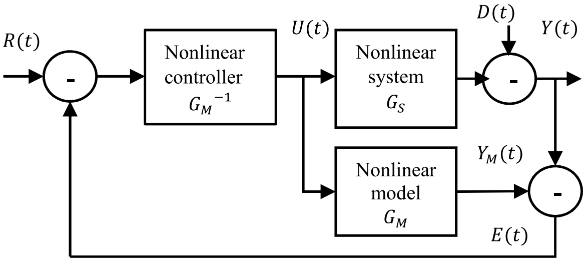


FIGURE 2. General IMC structure [19].

where the outputs, $y_1(t), y_2(t), \cdots, y_p(t)$ are expressed as functions of inputs, $u_1(t), u_2(t), \cdots, u_n(t)$ and the modelling mismatches. The multivariable U-model expresses the outputs in terms of weighted, time-delayed polynomials in the control input $u_1(t-1), u_2(t-1), \cdots, u_n(t-1)$ as:

$$\begin{aligned} y_1(t) = {} & b_{011} + b_{111} u_1(t-1) + b_{211} u_1(t-1)^2 \\ & + \ldots + b_{N11} u_1(t-1)^N \\ & + b_{021} + b_{121} u_2(t-1) + b_{221} u_2(t-1)^2 \\ & + \ldots + b_{N21} u_2(t-1)^N + \ldots \\ & + b_{0n1} + b_{1n1} u_n(t-1) + b_{2n1} u_n(t-1)^2 \\ & + \ldots + b_{Nn1} u_n(t-1)^N \end{aligned} \tag{4}$$

$$\begin{aligned} y_2(t) = {} & b_{012} + b_{112} u_1(t-1) + b_{212} u_1(t-1)^2 \\ & + \ldots + b_{N12} u_1(t-1)^N \\ & + b_{022} + b_{122} u_2(t-1) + b_{222} u_2(t-1)^2 \\ & + \ldots + b_{N22} u_2(t-1)^N + \ldots \\ & + b_{0n2} + b_{1n2} u_n(t-1) + b_{2n2} u_n(t-1)^2 \\ & + \ldots + b_{Nn2} u_n(t-1)^N \end{aligned} \tag{5}$$

Similarly, for the p^{th} output $y_p(t)$ will be:

$$\begin{aligned} y_p(t) = {} & b_{01p} + b_{11p} u_1(t-1) + b_{21p} u_1(t-1)^2 \\ & + \ldots + b_{N1p} u_1(t-1)^N \\ & + b_{02p} + b_{12p} u_2(t-1) + b_{22p} u_2(t-1)^2 \\ & + \ldots + b_{N2p} u_2(t-1)^N + \ldots \\ & + b_{0np} + b_{1np} u_n(t-1) + b_{2np} u_n(t-1)^2 \\ & + \ldots + b_{Nn2} u_n(t-1)^N \end{aligned} \tag{6}$$

The equations can be represented in vector form as:

$$y_p(t) = \left(b_{01p} b_{11p} b_{21p} \ldots b_{N1p}\right) \begin{pmatrix} 1 \\ u_1(t-1) \\ u_1(t-1)^2 \\ \vdots \\ u_1(t-1)^N \end{pmatrix} + \left(b_{02p} b_{12p} b_{22p} \ldots b_{N2p}\right) \begin{pmatrix} 1 \\ u_2(t-1) \\ u_2(t-1)^2 \\ \vdots \\ u_2(t-1)^N \end{pmatrix}$$

$$+\ldots+\left(b_{0np}b_{1np}b_{2n1}\ldots b_{Nnp}\right)\begin{pmatrix}1\\ u_n(t-1)\\ u_n(t-1)^2\\ \vdots\\ u_n(t-1)^N\end{pmatrix} \tag{7}$$

All the inputs and outputs are coupled through the parameters, b_{jkl}. The subscripts *j, k* and *l,* represent the j^{th} power of control signal, k^{th} row (model output), and l^{th} column (model input). The parameters b_{jkl} are identified iteratively using an adaptive algorithm. In particular the first coefficient for each input b_{0kl} can be adaptively trained to model the past values of the outputs [19]. Defining the coefficient vectors $\boldsymbol{b}_{jk}$ and the input vectors $\boldsymbol{u}_l$, the output equations can be written in closed form as

$$y_1=\boldsymbol{b}_{11}\boldsymbol{u}_1+\boldsymbol{b}_{21}\boldsymbol{u}_2+\ldots+\boldsymbol{b}_{n1}\boldsymbol{u}_n \tag{8}$$

$$y_2=\boldsymbol{b}_{12}\boldsymbol{u}_1+\boldsymbol{b}_{22}\boldsymbol{u}_2+\ldots+\boldsymbol{b}_{n2}\boldsymbol{u}_n \tag{9}$$

$$\vdots$$

$$y_p=\boldsymbol{b}_{1p}\boldsymbol{u}_1+\boldsymbol{b}_{2p}\boldsymbol{u}_2+\ldots+\boldsymbol{b}_{np}\boldsymbol{u}_n \tag{10}$$

The vectors, $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$, are comprised of powers of the inputs $u_1(t-1), u_2(t-1), \cdots, u_n(t-1)$ as:

$$\boldsymbol{u}_1=\begin{pmatrix}1\\ u_1(t-1)\\ u_1(t-1)^2\\ \vdots\\ u_1(t-1)^N\end{pmatrix},\quad \boldsymbol{u}_2=\begin{pmatrix}1\\ u_2(t-1)\\ u_2(t-1)^2\\ \vdots\\ u_2(t-1)^N\end{pmatrix},\ldots,$$

$$\boldsymbol{u}_n=\begin{pmatrix}1\\ u_n(t-1)\\ u_n(t-1)^2\\ \vdots\\ u_n(t-1)^N\end{pmatrix} \tag{11}$$

In matrix form

$$\begin{pmatrix}y_1\\ y_2\\ \vdots\\ y_p\end{pmatrix}=\begin{pmatrix}\boldsymbol{b}_{11} & \boldsymbol{b}_{21} & \ldots & \boldsymbol{b}_{n1}\\ \boldsymbol{b}_{12} & \boldsymbol{b}_{22} & \ldots & \boldsymbol{b}_{n2}\\ \ldots & \ldots & \ddots & \vdots\\ \boldsymbol{b}_{1p} & \boldsymbol{b}_{2p} & \ldots & \boldsymbol{b}_{np}\end{pmatrix} \times\begin{pmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \ldots & \boldsymbol{u}_n\end{pmatrix} \tag{12}$$

$$Y=BU \tag{13}$$

where:

$$B=\begin{pmatrix}\boldsymbol{b}_{11} & \boldsymbol{b}_{21} & \ldots & \boldsymbol{b}_{n1}\\ \boldsymbol{b}_{12} & \boldsymbol{b}_{22} & \ldots & \boldsymbol{b}_{n2}\\ \ldots & \ldots & \ddots & \vdots\\ \boldsymbol{b}_{1p} & \boldsymbol{b}_{2p} & \ldots & \boldsymbol{b}_{np}\end{pmatrix}\quad \text{and}$$

$$U=\begin{pmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \ldots & \boldsymbol{u}_n\end{pmatrix} \tag{14}$$

Equation (13) is a simplified of a multivariable nonlinear coupled U-model representation. In order to design the controller, we need to solve Equation (13). Since the system matrix B is a non-square matrix that includes the interaction among inputs and the outputs the Jacobian matrix requires the Newton-Raphson method. Therefore, the challenge is to avoid the singularity while inverting the non-square Jacobian matrix. We can avoid the singularity by using the following techniques available in the literature [27]:

(a) Employing pseudoinverse,
(b) Utilising the converse of the Jacobian framework from the past moment,
(c) Adding a small number to the Jacobian lattice to stay away from the singularity.

In order to acquire higher modelling accuracy with less error correlation, neural networks are integrated with the U-model methodology. Neural networks are a proven approach to the linear and nonlinear modelling and estimation process. A radial basis function neural network (RBFNN) consists of a single layered feedforward network and offers faster learning time compared to the multi-layered feedforward neural network (MFNN). RBFNN is capable of approximating the linear and nonlinearities in the systems. This algorithm has also been used in multi-applications such as control systems, signal processing, medical, renewable energy and machine learning such as in [28]–[37]. The RBFNN is incorporated with the U-model by computing the term $B_0(t)$ while $B_1(t), B_2(t), \ldots B_M(t)$ by the NLMS algorithm.

$$B_0(t)=Y(t)=\hat{w}_1\varphi(u(t-1))+\hat{w}_2\varphi(u(t-1)) +\ldots+\hat{w}_n\varphi(u(t-1)) \tag{15}$$

where $\hat{w}_n$ and φ are the weights with n number of neurons and basis function. The input of the *i*th hidden neuron is:

$$\varphi_i(\|u(t)-c_i\|) \tag{16}$$

where c_i is the centre of the *i*th hidden layer node while the weight vector for each neuron is:

$$W(t)=[\hat{w}_1\hat{w}_2\ldots\hat{w}_n] \tag{17}$$

Equation (15) can be simplified as:

$$B_0(t)=W\varphi(t) \tag{18}$$

The radial basis function (RBF) is chosen as an activation function due to better learning speed [28], [30], [31]. The activation function can be expressed as:

$$\phi(t)=e(-\frac{u(t-1)-c_i{}^2}{\beta^2})\quad \text{for } i=1,2\ldots n \tag{19}$$

where n is the number of hidden layer neurons, c_i is the centre of the ith hidden layer $(c_1, c_2, c_3..c_n)$ node and β is the width of the activation function. As a result the weights B_0 of the neural network and the rest of the parameters B_j ($j=1, 2$, N) are updated using NLMS as:

$$W(t+1)=W(t)+\mu(t)\,error(t)\phi(t)^T \tag{20}$$

$$B_j(t+1)=B_j(t)+\mu(t)\,error(t)U^j(t-1)^T \tag{21}$$

where $\mu(t)$ is the NLMS learning rate $(0<\mu(t)<1)$. These time varying parameters $B_i(t)$ and RBFNN weights

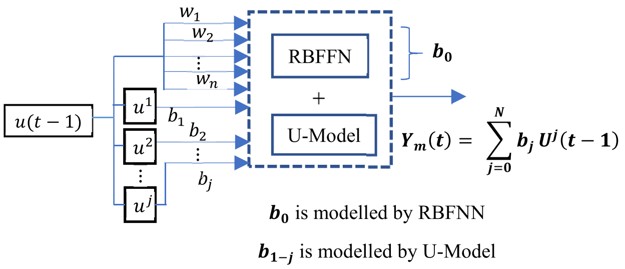

w_1
w_2
w_n
RBFFN
+
U-Model
b_0
$u(t-1)$
u^1
u^2
u^j
b_1
b_2
b_j
$Y_m(t) = \sum_{j=0}^{N} b_j U^j(t-1)$
b_0 is modelled by RBFNN
b_{1-j} is modelled by U-Model

FIGURE 3. U-model based IMC with RBFNN.

$W_i(t)$ are updated online via a NLMS filter as mentioned in Equations (20) and (21) simultaneously based on the identification tracking error. Thus, any changes in parameters in the dynamic system will be taken care off by these adaptive weights and parameter values. The representation of the adaptive mechanism with five neurons of the RBF activation function and third order of U-model polynomial equation is presented in Fig. 3.

The parameter $B_0(t)$ will compensate all the uncertainties, nonlinearities, and lagged parameters. Thus, the controller synthesis will only involve current control signal $u(t-1)$. This algorithm will reduce total computational demand during UUV mission, thus reducing total power consumption.

Based on Fig. 3, the U-model Y_m, incorporated with RBFNN for the SISO system can be expressed as:

$$\begin{aligned} Y_m = {} & \hat{w}_1\varphi_1(u(t-1)) + \cdots + \hat{w}_n\varphi_n(u(t-1)) \\ & + b_1(t)u(t-1) + b_2(t)u^2(t-1) \\ & + \cdots + b_n(t)u^n(t-1) \end{aligned} \tag{22}$$

C. ROBOT OPERATING SYSTEM (ROS) GIRONA 500 UUV

GIRONA 500 UUV platform was developed by Underwater Robotic Laboratory, University of Girona Spain in 2012 for underwater survey and intervention. The unique design of GIRONA 500 UUV with capability to accommodate a large volume of different instruments and equipment such underwater manipulator [38]. Fig. 1 show the design and characteristic of GIRONA 500 UUV platform. The main structure of this platform is Aluminium 6082-T6 alloy frame T-shaped pillars and torpedo-shaped pressure cylinder for electronic compartment. The overall size is (1 m in height x 1 m in width x 1.5 m in length) and weight up to 200 kg depending on the payload being attached. The platform can be equipped with three to eight underwater thrusters depending on the underwater intervention mission for under-actuated or fully actuated control autonomy. The pressure cylinder are covered with thermoformed ABS plastic skin whose streamline shape is based on the Myring hull profile equation [39] for drag reduction. The maximum operating depth for underwater intervention is up to 500 m. This platform implements open source ROS for system integration and control.

The floatation modules, consist of epoxy composite foam with a density 400 kg/m^3, are place in top part of the platform to make the vehicle neutrally buoyant. This is to make sure the platform will afloat when critical failure during underwater intervention mission. Heavier equipment is place in the lower hull to create good performance stability in roll and pitch that influence the separation between centre of buoyancy (CB) and centre of gravity (CG) make it suitable for bathymetric and imaging survey. The power system consists of Li-ion battery packs with combined capacity of 2.2 kWh of energy. RS-232 interface connection is implemented to manage and monitor the state of the system at any time such temperature, voltage condition, and time to full charge/discharge. The battery system is located at the bottom part of the vehicle. A 614.4 kHz phased array of doppler velocity log (DVL), attitude and heading reference system (AHRS) added by fibre optic gyro (FOG) sensors are used for dead-reckoning navigation. Absolute position in the surface is determined by GPS system while ultra-short baseline (USBL) for underwater. Sound velocity sensor (SVS) is used for measuring sound speed profile for optimizing data accuracy for underwater acoustic devices (DVL, UBSL, and Side Scan Sonar). Different configuration of thrusters can be implements onto the vehicle for greater autonomy and long-range mission endurance.

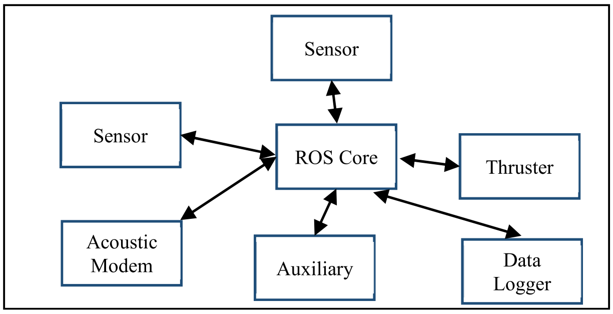


FIGURE 4. ROS software architecture.

The coupled underactuated multivariable nonlinear adaptive U-model based IMC controller is designed and developed in the ROS architecture [40] for rapid implementation. The UUV application involves multiple sensors and transducers in order to operate in a harsh marine environment, thus making the system configuration and system integration process difficult and complex. The ROS framework uses the "publish-subscribe" method by passing the message around the sensors and transducers compared with the conventional method that is highly time consuming. The ROS framework simplifies the communication and integration process via the ROS core system, thus making it faster to implement.

Fig. 4 shows the ROS software architecture. The conventional software architecture shown in Fig. 5 is mainly for one-to-one communication and needs to be reconfigured again in order to change one module while the ROS based communication architecture uses "publish-subscribe" via the ROS core system [41].

The ROS operates using an Ubuntu Linux open source operating system and can be learnt through *http://wiki.ros.org/*. A ROS system is made up of many

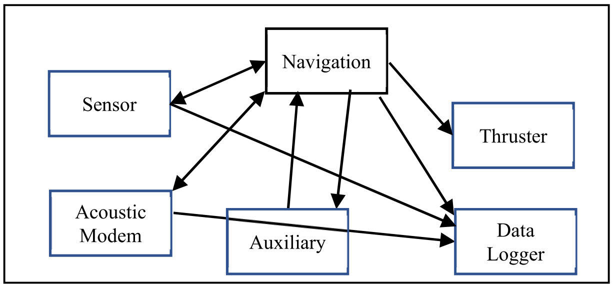


FIGURE 5. Convention software architecture.

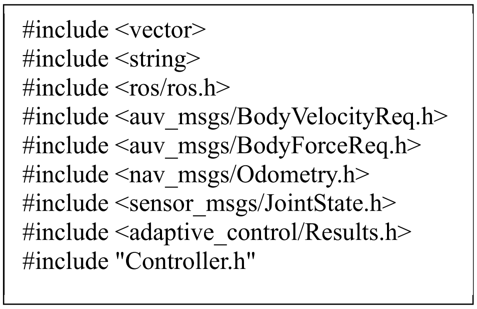
different program call nodes or topics that run simultaneously and communicate with each other by passing messages. The basic programming languages for ROS applications are Python and C++ for simulation and real-time implementation. A new node or topic is designed for the adaptive nonlinear U-model based IMC in order to subscribe to all the relevant topics and to publish a U-model controller node for the GIRONA 500 UUV platform. In this new node, topic *pose_ekf_slam/odometry*, *cola2_navigation/nav_sts*, *cola2_control/body_velocity_req* represent the navigation data, an extended Kalman Filter (EKF) with simultaneous localisation and mapping (SLAM) for underwater 6DOF localisation and mapping application [42], [43] and the velocity input for the underwater thrusters.

Catkin is set of tools that the ROS uses to generate executable programs, libraries, scripts, and interfaces that other code can use. Catkin comprises a set of *CMake* macros and custom Python scripts. *CMake* is commonly used for open source build systems. To build a new node, a dedicated directory or workspace must be created through Catkin. A workspace consists of directories for ROS code to be stored and we can have multiple ROS workspaces but only one workspace can be run at any one time. A Catkin workspace can be created and initialised through a command-line as:

```
cirs@cirs~ubuntu : ~$ mkdir-p~?catkin_ws/src
cirs@cirs~ubuntu : ~$ cd~/catkin_ws/src
cirs@cirs~ubuntu : ~$ catkin_init_workspace
```

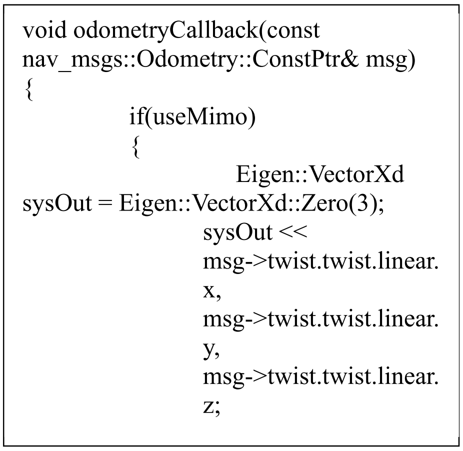
The 6 DOF vehicle position in the reference frame is achieved through an extended Kalman Filter (EKF) filter and *multistep_excitation*: reference input signal generation for controller responses, validation, and performance. This adaptive nonlinear U-model based IMC controller node publishes messages to the *cola2_control/body_velocity_req* and *cola2_control/body_force_req*. However, only *body_velocity_req* messages will be used for the controller reference. Fig. 6 and Fig. 7 show the publish messages algorithm to send messages from the adaptive nonlinear U-model based IMC node to the *body_velocity_req* node.

The subscribe messages form *pose_ekf_slam/odometry* node consists of the velocity data in surge, swag, and heave or x, y, and z directions. All these messages are used for the sensor update or current velocity and position during simulation and real-time testing for GIRONA 500 UUV. In term of the algorithm, the Eigen C++ library that is suitable for linear algebra, matrices, vectors, and related applications is implemented. This is suitable for the adaptive nonlinear U-model based IMC that consists of polynomial equations. The controller computation also involves the Eigen C++ library. The published messages from the adaptive nonlinear U-model based IMC controller node are the velocity data in x, y, and z directions through U-model control synthesis. This data is the controller update values for the GIRONA 500 UUV platform. Fig. 8 shows the messages for subscribe and publish for the adaptive nonlinear U-model based IMC in the ROS GIRONA 500 UUV architecture during simulation and real-time implementation.



```
#include <vector>
#include <string>
#include <ros/ros.h>
#include <auv_msgs/BodyVelocityReq.h>
#include <auv_msgs/BodyForceReq.h>
#include <nav_msgs/Odometry.h>
#include <sensor_msgs/JointState.h>
#include <adaptive_control/Results.h>
#include "Controller.h"
```

FIGURE 6. Controller node header program.



```
void odometryCallback(const
nav_msgs::Odometry::ConstPtr& msg)
{
            if(useMimo)
            {
                        Eigen::VectorXd
sysOut = Eigen::VectorXd::Zero(3);
                    sysOut <<
                    msg->twist.twist.linear.
                    x,
                    msg->twist.twist.linear.
                    y,
                    msg->twist.twist.linear.
                    z;
```

FIGURE 7. Subscribe *pose_ekf_slam/odometry* node for U-model node.

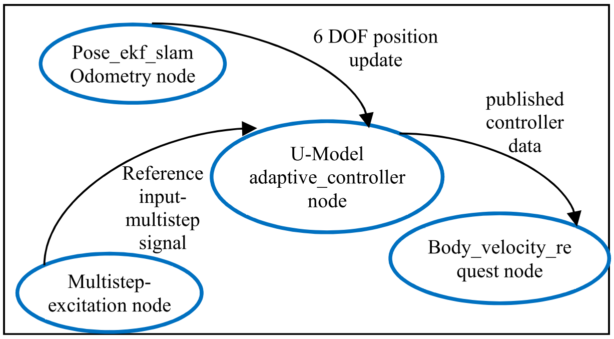

Pose_ekf_slam Odometry node
6 DOF position update
published controller data
U-Model adaptive_controller node
Reference input-multistep signal
Body_velocity_request node
Multistep-excitation node

FIGURE 8. Subscribe and publish messages from U-model adaptive controller node.

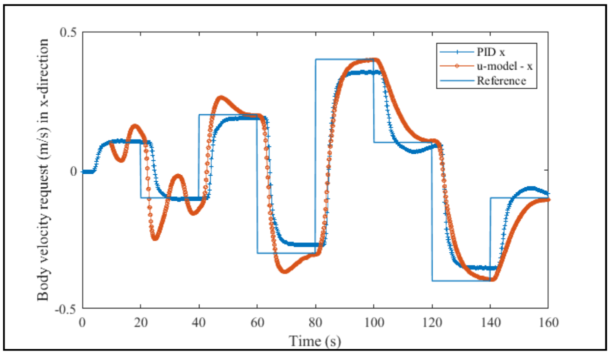


FIGURE 9. Comparison of control tracking in the surge (x-direction).

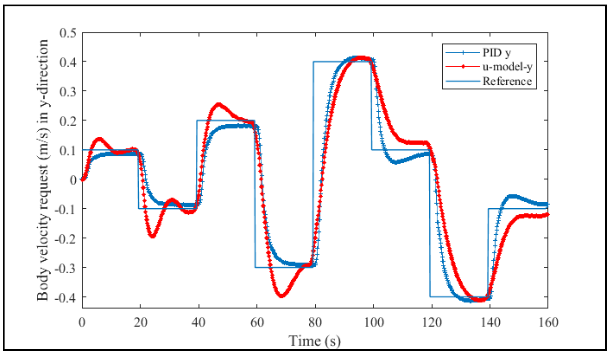


FIGURE 10. Comparison of control tracking in the sway (y-direction).

Rvis is a 3D ROS visualisation tool GUI for robotic platform simulation in the targeted environment such as an underwater scenario with sensor simulation [44]–[46]. Rvis can plot and display many types of data streams through ROS nodes such as cameras, underwater acoustic transducers, a manipulator arm, and other related sensors. All these elements such as 3D rendering of the view and data from sensors can be accessed simultaneously. Using the launch command in the terminal, the Rvis terminal and GIRONA 500 UUV can be used through a teleoperated keyboard key or pre-defined manoeuvring control. A multistep excitation node controller reference will be used for the adaptive nonlinear U-model-based IMC controller node in the simulations and real-time experiment as a pre-defined manoeuvring control.

III. SIMULATION RESULTS

The simulation results of the GIRONA 500 UUV implementing an adaptive nonlinear IMC-based U-model algorithm and a comparison with the existing GIRONA 500 UUV workable mission PID controller are shown in Fig. 9 to Fig. 12. The 3rd order system of the MIMO U-model was set up with initial values for the parameters β_1, β_2, and $\beta_3 = [0.1, 0.1, 0.1]$ with a learning rate equal to 0.05 and small gain parameter $q = 0.01$.

Fig. 9 shows the comparison of tracking control of body velocity request in the surge or x-direction. Adaptive nonlinear IMC based U-model and PID control approaches converged to the desired reference value. The U-model control approach converged with signal oscillation at the beginning of the simulation without a proper tuning process compared to the PID control method. The GIRONA 500 AUV PID control parameters were already tested and proven during the simulation and real-time missions. However, the U-model control approach may be improved with proper tuning gain selection. The adaptive mechanism in the U-model adapted to changing dynamics, although with random values (0 to 1) for the learning rate, initial adaptive parameter values, and q value. Thus, the U-model control approach was capable of synthesising control signals for the unknown control plants. Similar trends have been observed in Fig. 10 for comparison of control tracking in the sway (y-direction), in which both controllers converged to the desired body velocity request reference value. The U-model control approach produced oscillation in the control signal compared to the PID control method. Again, the same learning rate, initial adaptive parameter values, and q value were implemented in the simulation compared to the PID control method using different controller gains. Lastly, Fig. 11 shows the comparison of control tracking in the heave (z-direction). Both controllers converged to the desired reference value which the U-model control approach converged closest to the reference value compared to the PID control approach. The GIRONA 500 UUV PID control parameters were already tested and proven during simulation and real-time missions. However, the U-model control approach may be improved with proper tuning gain

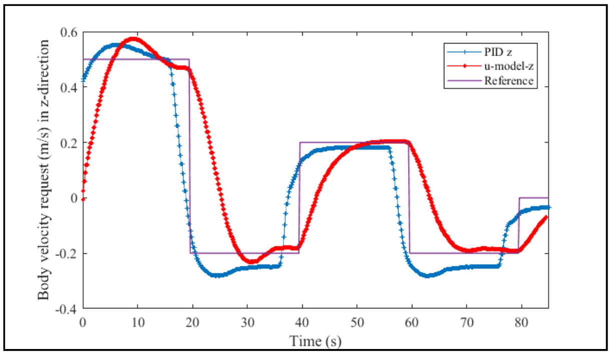
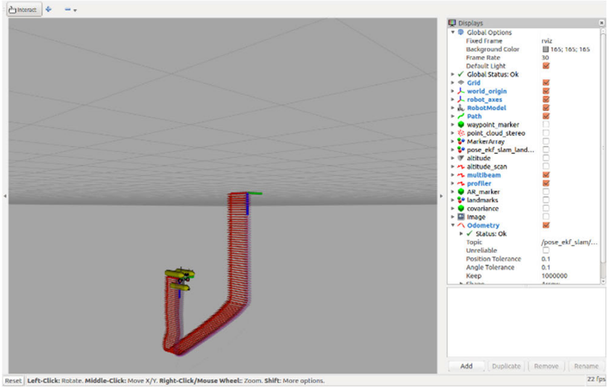


FIGURE 11. Comparison of control tracking in the heave (z-direction).


Interact
Displays
Add
Duplicate
Remove
Rename
22 fps
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.

```
Global Options
  Fixed Frame            rviz
  Background Color       165; 165; 165
  Frame Rate             30
  Default Light          ☑
Global Status: Ok
Grid                     ☑
world_origin             ☑
robot_axes               ☑
RobotModel               ☑
Path                     ☑
waypoint_marker          □
point_cloud_stereo       □
MarkerArray              □
pose_ekf_slam_land...    □
altitude                 □
altitude_scan            □
multibeam                ☑
profiler                 ☑
AR_marker                □
landmarks                □
covariance               □
Image                    □
Odometry                 ☑
  Status: Ok
  Topic                  /pose_ekf_slam/...
  Unreliable             □
  Position Tolerance     0.1
  Angle Tolerance        0.1
  Keep                   1000000
```

FIGURE 12. 3D Rvis GIRONA UUV environment.

FIGURE 13. Girona Underwater Vision and Robotics pool facilities.

selection. The adaptive mechanism in the U-model adapted to changing dynamics, although with random values (0 to 1) for the learning rate, initial adaptive parameter values, and q value.

Thus, the U-model control approach was found capable of synthesising control signals for unknown control plants. Fig. 12 shows the Rvis GUI interface for GIRONA 500 UUV with way point marker during underwater manoeuvring simulations. In the Rvis GUI display, we can select several nodes to be included during simulation such as robot model, path, odometry, multibeam, world_origin, and profiler.

IV. REAL-TIME EXPERIMENT RESULTS

The experiment was conducted in order to validate the adaptive nonlinear IMC-based U-model algorithm from the ROS simulation environment into a real-time workable controller. The experiment was conducted at the pool testing facilities at the University of Girona, Spain as in Fig. 13. The size of the pool was 16 m length x 8 m width. Due to this underwater terrain, the experiment was conducted in decoupled manoeuvres due to the limited underwater space and safety of the underwater vehicle. The data was transmitted online via a communication cable or umbilical cord from the GIRONA 500 UUV. The advantage of this approach was the controller, and the algorithm could be tuned online without having to retrieve the underwater vehicle to the surface. All the reference and odometry data was saved as *.bag record* format and exported to the MATLAB Simulink for analysis.

Several real-time experiments were conducted in the pool facilities for the control validation as in Fig. 14. The predefined velocity reference consisted of multistep excitation of different amplitudes in the x, y, and z directions with a period of 10 s for each amplitude for the real-time validation due to limited underwater manoeuvring space. $R_1(t)$ as the x-axis, $R_2(t)$ is the y-axis and $R_3(t)$ for z-axis where $R_1(t)$ and $R_2(t)$ are $[0.1\ -0.1\ 0.2\ -0.3\ 0.4\ 0.1\ -0.4\ -0.1]\ \text{ms}^{-1}$ and $R_3(t)$ is $[0.05\ -0.05\ 0.1\ -0.2\ 0.1\ -0.2\ 0]\ \text{ms}^{-1}$. All real-time data (*.rosbag*) was exported to MATLAB Simulink 2018b version due to integration capability with the ROS environment using *U_Model=rosbag('u_model.bag')* in the command window, where *u_model.bag* is the rosbag file from the ROS environment. Next, the ROS topic odometry was selected for further analysis by entering *odometryU_Model = select(U_Model,'Topic','/pose_ekf_slam/odometry').* Then, the rosbag file was converted to a time series data by entering *odometry_tsU_Model= timeseries(odometryU_Model)* in the command window. Lastly, the time series data was analysed using the MATLAB plotting tool.

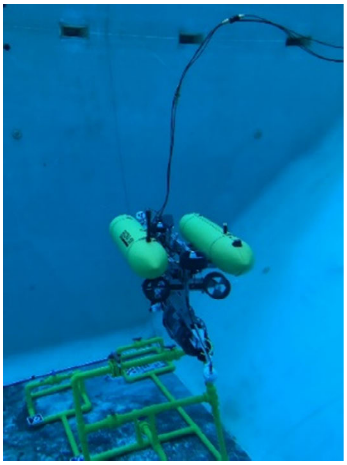


FIGURE 14. GIRONA 500 UUV during real-time experiments.

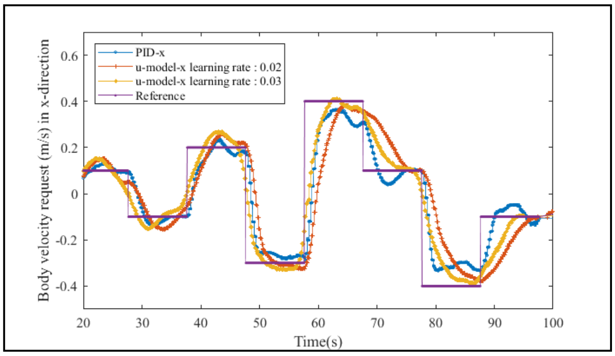


FIGURE 15. Real-time comparison of control tracking in the surge (x-direction).

In Fig. 15, two U-model control methods were compared with the PID control method on the GIRONA 500 UUV platform. All the control methods converged to the desired multi-amplitude reference body velocity request signal. The U-model with a learning rate of 0.03 converged with better performance compared with the U-model with a learning rate of 0.02 and the PID control approach. The U-model with a learning rate of 0.03 converged closely with the reference value for most of the reference amplitudes compared to the existing PID controller.

Fig. 16 shows a comparison of control tracking in the sway (y-direction). Again, the U-model approach used a similar learning rate value of 0.02 and 0.03 to compare with the PID control approach. All the controller approaches converged to the desired body velocity request value. The vehicle movement was not smooth compared to the surge (x-direction) testing due to the limited number of underwater thrusters

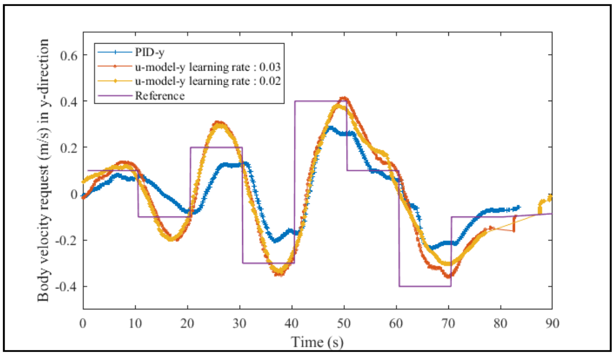


FIGURE 16. Real-time comparison of control tracking in the sway (y-direction).

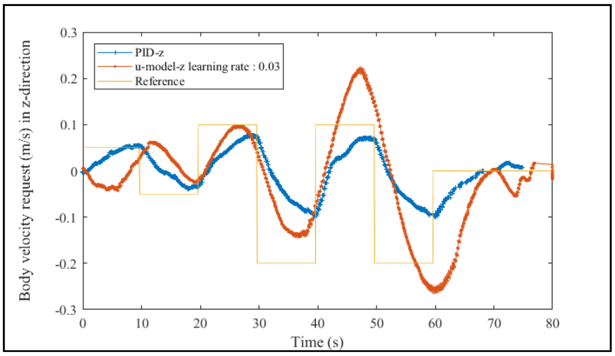


FIGURE 17. Real-time comparison of control tracking in the heave (z-direction).

and representing underactuated control manoeuvring. Both the U-model control approaches converged with better performance compare to the PID control approach although with a small overshoot during the first of three cycles of the multi-amplitude signal.

Fig. 17 shows the real-time experiment comparison of control tracking in the heave (z-direction). Both control approaches tried to converge to the desired body velocity request reference signal due to the shorter period of 10 seconds compared to the simulation in the Rvis 3D visualisation tool that used a 20 second time frame period for each of the reference amplitudes. Due to fast-changing dynamics, both controllers needed enough time to converge as in the Rvis 3D visualisation. However, both controllers converged at the end of the experiment due to the extra time of the reference body velocity request signal. The reference body velocity request value could be reduced to a smaller amount to enable the controller to converge to the desired reference. This could be seen in the experiment, where the vehicle velocity movement in the heave direction was more than the surge and sway directions. This did not reflect a real underwater mission because, in the heave direction, the GIRONA 500 UUV was moving at a slower speed due to safety reasons. The underwater thrusters for the heave direction usually implement a higher torque on the motor than the horizontal thrusters for the surge and sway directions due to the buoyancy force against the vehicle, thus affecting the vehicle in the heave direction.

V. CONCLUSION

The implementation of an adaptive nonlinear IMC based U-model in a robot operating system (ROS) was undertaken in preparation for real-time experiments. The development of the new controller adaptive nonlinear IMC based U-model node in a C++ algorithm was executed successfully in the Rvis 3D environment and compared to the existing mission controller. The nonlinear IMC based U-model approach only required a suitable initial condition range from 0 to 1 for all the parameters, weights, and appropriate learning rate values for better stability and convergence speed. This nonlinear control approach can minimise the time duration of the controller design by implementing an adaptive learning rate method for the unknown control system compared with the PID control approach. The formulation and implementation of the adaptive nonlinear U-model based IMC model are much simpler than other complex mathematical modelling and control approaches. The results showed all the control tracking in surge, sway, and heave directions converged to the desired reference values. The real time experiment showed good results with all the controllers converging to the desired values except in the heave direction for which the controller required an extra period of steady state reference value instead of just 10 seconds. However, the results were considered acceptable and satisfactory because the controller was able to follow the reference values.

ACKNOWLEDGMENT

The experiment was conducted at the Computer Vision and Robotics pool facilities of University of Girona, Spain, using the GIRONA 500 UUV platform under a research attachment programme funded by the Public Service Department (PSD) Malaysia.

REFERENCES

[1] C. Shen, B. Buckham, and Y. Shi, "Modified C/GMRES algorithm for fast nonlinear model predictive tracking control of AUVs," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1896–1904, Sep. 2017, doi: 10.1109/TCST.2016.2628803.

[2] Y. Chen, R. Zhang, X. Zhao, and J. Gao, "Adaptive fuzzy inverse trajectory tracking control of underactuated underwater vehicle with uncertainties," *Ocean Eng.*, vol. 121, pp. 123–133, Jul. 2016, doi: 10.1016/j.oceaneng.2016.05.034.

[3] O. Korkmaz, S. K. Ider, and M. K. Ozgoren, "Control of an underactuated underwater vehicle manipulator system in the presence of parametric uncertainty and disturbance," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 578–584, doi: 10.1109/ACC.2013.6579899.

[4] C. P. Bechlioulis, G. C. Karras, S. Heshmati-Alamdari, and K. J. Kyriakopoulos, "Trajectory tracking with prescribed performance for underactuated underwater vehicles under model uncertainties and external disturbances," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 429–440, Mar. 2016, doi: 10.1109/TCST.2016.2555247.

[5] S. Mohan, J. Kim, and Y. Kim, "A null space control of an underactuated underwater vehicle-manipulator system under ocean currents," in *Proc. Oceans Yeosu*, May 2012, pp. 1–5, doi: 10.1109/OCEANS-Yeosu.2012.6263367.

[6] Z. Zheng, L. Ruan, and M. Zhu, "Output-constrained tracking control of an underactuated autonomous underwater vehicle with uncertainties," *Ocean Eng.*, vol. 175, pp. 241–250, Mar. 2019, doi: 10.1016/j.oceaneng.2019.02.023.

[7] J. Guerrero, J. Torres, V. Creuze, and A. Chemori, "Trajectory tracking for autonomous underwater vehicle: An adaptive approach," *Ocean Eng.*, vol. 172, pp. 511–522, Jan. 2019, doi: 10.1016/j.oceaneng.2018.12.027.

[8] B. Huang and Q. Yang, "Double-loop sliding mode controller with a novel switching term for the trajectory tracking of work-class ROVs," *Ocean Eng.*, vol. 178, pp. 80–94, Apr. 2019, doi: 10.1016/j.oceaneng.2019.02.043.

[9] I.-L.-G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl, "Trajectory tracking for an articulated intervention AUV using a super-twisting algorithm in 6 DOF," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 311–316, 2018, doi: 10.1016/j.ifacol.2018.09.506.

[10] J. Guerrero, E. Antonio, A. Manzanilla, J. Torres, and R. Lozano, "Autonomous underwater vehicle robust path tracking: Auto-adjustable gain high order sliding mode controller," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 161–166, 2018, doi: 10.1016/j.ifacol.2018.07.272.

[11] A. Balasuriya and L. Cong, "Adaptive fuzzy sliding mode controller for underwater vehicles," in *Proc. 4th Int. Conf. Control Autom. ICCA Final Program Book Abstr. ICCA*, Jun. 2003, pp. 917–921, doi: 10.1109/ICCA.2003.1595156.

[12] J. Singh, K. Chattterjee, and C. B. Vishwakarma, "Two degree of freedom internal model control-PID design for LFC of power systems via logarithmic approximations," *ISA Trans.*, vol. 72, pp. 185–196, Jan. 2018, doi: 10.1016/j.isatra.2017.12.002.

[13] F. Baklouti, S. Aloui, and A. Chaari, "Adaptive fuzzy sliding mode tracking control of uncertain underactuated nonlinear systems," *J. Control Sci. Eng.*, vol. 2016, p. 6, Jan. 2016.

[14] G. V. Lakhekar, L. M. Waghmare, and P. S. Londhe, "Enhanced dynamic fuzzy sliding mode controller for autonomous underwater vehicles," in *Proc. IEEE Underwater Technol. (UT)*, Feb. 2015, pp. 1–7, doi: 10.1109/UT.2015.7108283.

[15] Z. H. Ismail, M. B. M. Mokhar, V. W. E. Putranti, and M. W. Dunnigan, "A robust dynamic region-based control scheme for an autonomous underwater vehicle," *Ocean Eng.*, vol. 111, pp. 155–165, Jan. 2016, doi: 10.1016/j.oceaneng.2015.10.052.

[16] Y. Wang, K. Zhu, B. Chen, and M. Jin, "Model-free continuous nonsingular fast terminal sliding mode control for cable-driven manipulators," *ISA Trans.*, vol. 98, pp. 483–495, Mar. 2020, doi: 10.1016/j.isatra.2019.08.046.

[17] Y. Wang, L. Liu, D. Wang, F. Ju, and B. Chen, "Time-delay control using a novel nonlinear adaptive law for accurate trajectory tracking of cable-driven robots," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5234–5243, Aug. 2020, doi: 10.1109/TII.2019.2951741.

[18] Q. M. Zhu and L. Z. Guo, "A pole placement controller for non-linear dynamic plants," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 216, no. 6, pp. 467–476, Sep. 2002, doi: 10.1177/095965180221600603.

[19] N. A. A. Hussain, S. S. A. Ali, M. Ovinis, M. R. Arshad, and U. M. Al-Saggaf, "Underactuated coupled nonlinear adaptive control synthesis using U-model for multivariable unmanned marine robotics," *IEEE Access*, vol. 8, pp. 1851–1865, 2020, doi: 10.1109/ACCESS.2019.2961700.

[20] Y. Bouzid, H. Siguerdidjane, and Y. Bestaoui, "Nonlinear internal model control applied to VTOL multi-rotors UAV," *Mechatronics*, vol. 47, pp. 49–66, Nov. 2017, doi: 10.1016/j.mechatronics.2017.08.002.

[21] M. Hannachi and D. Soudani, "Internal model control of multivariable discrete-time systems," in *Proc. 7th Int. Conf. Model., Identificat. Control (ICMIC)*, Dec. 2015, pp. 1–6, doi: 10.1109/ICMIC.2015.7409384.

[22] L. Angel, J. Viola, and M. Paez, "Speed control of a motor-generator system using internal model control techniques," in *Proc. IEEE 3rd Colombian Conf. Autom. Control (CCAC)*, Oct. 2017, pp. 1–6, doi: 10.1109/CCAC.2017.8276453.

[23] A. K. Yadav and P. Gaur, "Intelligent modified internal model control for speed control of nonlinear uncertain heavy duty vehicles," *ISA Trans.*, vol. 56, pp. 288–298, May 2015, doi: 10.1016/j.isatra.2014.12.001.

[24] Y. Bouzid, H. Siguerdidjane, and Y. Bestaoui, "Improved 3D trajectory tracking by nonlinear internal model-feedback linearization control strategy for autonomous systems," *IFAC-PapersOnLine*, vol. 49, no. 9, pp. 13–18, 2016, doi: 10.1016/j.ifacol.2016.07.480.

[25] S. S. A. Ali, F. M. Al-Sunni, M. Shafiq, and J. M. Bakhashwain, "U-model based learning feedforward control of MIMO nonlinear systems," *Electr. Eng.*, vol. 91, no. 8, pp. 405–415, Apr. 2010, doi: 10.1007/s00202-010-0149-3.

[26] M. Shafiq and N. R. Butt, "U-model based adaptive IMC for nonlinear dynamic plants," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, Sep. 2005, p. 5, doi: 10.1109/ETFA.2005.1612627.

[27] A. Syed, S. Azhar, F. M. Al-Sunni, M. Shafiq, J. M. Bakhashwain, and S. S. Azhar, "MIMO U-model based control?: Real-time tracking control and feedback analysis via small gain theorem," *WSEAS Trans. Circuits Syst.*, vol. 7, no. 7, pp. 610–619, 2008.

[28] R. ul Amin, L. Aijun, L. Hongshi, and L. Jiaxing, "An adaptive sliding mode control based on radial basis function network for attitude tracking control of four rotor hover system," in *Proc. IEEE Chin. Guid., Navigat. Control Conf. (CGNCC)*, Aug. 2016, pp. 580–585, doi: 10.1109/CGNCC.2016.7828850.

[29] D. Lam and D. Wunsch, "Unsupervised feature learning classification with radial basis function extreme learning machine using graphic processors," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 224–231, Jan. 2017, doi: 10.1109/TCYB.2015.2511149.

[30] T. Mutaz and A. Ahmad, "Solar radiation prediction using radial basis function models," in *Proc. Int. Conf. Develop. E-Syst. Eng. (DeSE)*, Dec. 2015, pp. 77–82, doi: 10.1109/DeSE.2015.55.

[31] M. G. Ruano, E. Hajimani, and A. E. Ruano, "A radial basis function classifier for the automatic diagnosis of cerebral vascular accidents," in *Proc. Global Med. Eng. Phys. Exchanges/Pan Amer. Health Care Exchanges (GMEPE/PAHCE)*, Apr. 2016, pp. 1–4, doi: 10.1109/GMEPE-PAHCE.2016.7504656.

[32] C. Lee, "Gender classification using M-estimator based radial basis function neural network," in *Proc. Int. Conf. Signal Process. Multimedia Appl. (SIGMAP)*, Vienna, Austria, 2014, pp. 302–306.

[33] R. A. Husain, A. S. Zayed, W. M. Ahmed, and H. S. Elhaji, "Image segmentation with improved watershed algorithm using radial bases function neural networks," in *Proc. 16th Int. Conf. Sci. Techn. Autom. Control Comput. Eng. (STA)*, Dec. 2015, pp. 357–362, doi: 10.1109/STA.2015.7505175.

[34] N. Vafamand,"Adaptive robust Neural-Network-Based backstepping control of tethered satellites with additive stochastic noise," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3922–3930, Oct. 2020, doi: 10.1109/taes.2020.2985276.

[35] C. Xi and J. Dong, "Adaptive neural network-based control of uncertain nonlinear systems with time-varying full-state constraints and input constraint," *Neurocomputing*, vol. 357, pp. 108–115, Sep. 2019, doi: 10.1016/j.neucom.2019.04.060.

[36] B. Homayoun, M. M. Arefi, N. Vafamand, and S. Yin, "Neuro-adaptive command filter control of stochastic time-delayed nonstrict-feedback systems with unknown input saturation," *J. Franklin Inst.*, vol. 357, no. 12, pp. 7456–7482, Aug. 2020, doi: 10.1016/j.jfranklin.2020.04.042.

[37] Q. Hou, J. Dong, and C. Xi, "Adaptive neural network tracking control for switched uncertain non-linear systems with actuator failures and time-varying delays," *IET Control Theory Appl.*, vol. 13, no. 12, pp. 1929–1939, Aug. 2019, doi: 10.1049/iet-cta.2018.6188.

[38] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From survey to intervention," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb. 2012, doi: 10.1109/TMECH.2011.2174065.

[39] D. F. Myring, "A theoretical study of body drag in subcritical axisymmetric flow," *Aeronaut. Quart.*, vol. 27, no. 3, pp. 186–194, 1976, doi: 10.1017/S000192590000768X.

[40] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS, First*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[41] T. Seta, A. Okamoto, S. Inaba, and M. Sasano, "A new ROS-based operation software for an autonomous underwater vehicle HOBALIN and its test in real sea," in *Proc. OCEANS MTS/IEEE Kobe Techno-Oceans (OTO)*, May 2018, pp. 1–6, doi: 10.1109/OCEANSKOBE.2018.8559223.

[42] F. Hidalgo and T. Braunl, "Review of underwater SLAM techniques," in *Proc. 6th Int. Conf. Autom., Robot. Appl. (ICARA)*, Feb. 2015, pp. 306–311, doi: 10.1109/ICARA.2015.7081165.

[43] E. Galceran, R. Campos, N. Palomeras, M. Carreras, and P. Ridao, "Coverage path planning with realtime replanning for inspection of 3D underwater structures," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Hong Kong, 2014, pp. 6586–6591, doi: 10.1109/ICRA.2014.6907831.

[44] R. Mishra and A. Javed, "ROS based service robot platform," in *Proc. 4th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2018, pp. 55–59, doi: 10.1109/ICCAR.2018.8384644.

[45] Y. Gao, S. Li, X. Wang, and Q. Chen, "A patrol mobile robot for power transformer substations based on ROS," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 456–460, doi: 10.1109/CCDC.2018.8407176.

[46] S. S. Maram, T. Vishnoi, and S. Pandey, "Neural network and ROS based threat detection and patrolling assistance," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Paradigms (ICACCP)*, Feb. 2019, pp. 1–5, doi: 10.1109/ICACCP.2019.8883008.

NUR AFANDE ALI HUSSAIN (Member, IEEE) received the B.E. (electrical) degree (Hons.) from Universiti Teknologi Malaysia (UTM), in 2003, the M.Sc. degree of electrical & electronic engineering from Universiti Sains Malaysia (USM), in 2014, and the Ph.D. degree in electrical & electronic engineering with Universiti Teknologi PETRONAS (UTP), Malaysia, in 2020, with a focus on adaptive non-linear MIMO control system for marine robotics. He was a Former Visiting Researcher with Girona Underwater Vision and Robotics, University of Girona, Spain. His research interest includes unmanned control system engineering.

SYED SAAD AZHAR ALI (Senior Member, IEEE) received the B.S. degree from the NED University of Engineering & Technology, Pakistan, in 1999, and the M.S. degree and the Ph.D. degree in electrical engineering from the King Fahd University of Petroleum & Minerals, Saudi Arabia, in 2001 and 2007, respectively. He joined the King Fahd University of Petroleum & Minerals, in 2000. He has served for Air University and Iqra University, Pakistan, prior to joining the Center for Intelligent Signal & Imaging Research, Universiti Teknologi PETRONAS (UTP), Malaysia, as an Associate Professor with areas of research in nonlinear control, underwater robotics, underwater image enhancement, and signal processing.

PERE RIDAO (Member, IEEE) received the Ph.D. degree in computer engineering from the University of Girona, Spain, in 2001. He is currently the Director of the Computer Vision and Robotics Research Institute (VICOROB), the Head of the Underwater Robotics Research Center (CIRS), and an Associate Professor with the Department of Computer Engineering, University of Girona. Since 1997, he has participated in 19 research projects (ten European and nine National), he is the author of more than 100 publications. His research interests include designing and developing autonomous underwater vehicles for 3D mapping and intervention. He is also the Chair of the IFAC's Technical Committee on Marine Systems and a Co-Founder of IQUA Robotics S.L. UdG Spin-off company devoted to the manufacturing and commercialization of AUV technologies.

PATRYK CIESLAK received the M.Sc. degree in engineering and the Ph.D. degree from the Department of Robotics and Mechatronics, Faculty of Mechanical Engineering and Robotics, AGH University of Science and Technology, Krakow, Poland. He is currently a Postdoctoral Research Fellow with the Underwater Robotics Research Center (CIRS), University of Girona, Spain, and successfully developed and implemented a control system for an autonomous underwater vehicle-manipulator system (AUVMS) for the GIRONA 500 UUV and 4-DoF manipulator in a ROS based architecture. His research interests include control system design, especially for mobile, and underactuated and underwater robots.

UBAID M. AL-SAGGAF (Member, IEEE) received the B.Sc. degree (Hons.) in electrical engineering and the B.Sc. degree (Hons.) in mathematics from the King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, in 1980, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, USA, in 1983 and 1986, respectively. He was appointed as an Assistant Professor with the King Fahd University of Petroleum & Minerals, in 1986, and then an Associate Professor in 1991. He was on leave from 1992 to 2010 and worked for the Ministry of Defence and Aviation as an Executive Technical Advisor for the Research and Development Department. He joined the Department of Electrical and Computer Engineering, King Abdulaziz University, in 2010, where he is currently a Professor. He is also the Director of the Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University, and the Innovation and Prototyping Center (IPC), King Abdulaziz University.