

Article

State Estimation of an Underwater Markov Chain Maneuvering Target Using Intelligent Computing

Wasiq Ali ^{1,2} , Yaan Li ¹ , Muhammad Asif Zahoor Raja ³ , Wasim Ullah Khan ^{4,*} and Yigang He ^{4,*}

- ¹ School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China; wasiqali@mail.nwpu.edu.cn (W.A.); liyaan@nwpu.edu.cn (Y.L.)
- ² Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan
- ³ Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Yunlin 64002, Taiwan; rajamaz@yuntech.edu.tw
- ⁴ School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China
- * Correspondence: kwasim814@whu.edu.cn (W.U.K.); yghe1221@whu.edu.cn (Y.H.)

Abstract: In this study, an application of deep learning-based neural computing is proposed for efficient real-time state estimation of the Markov chain underwater maneuvering object. The designed intelligent strategy is exploiting the strength of nonlinear autoregressive with an exogenous input (NARX) network model, which has the capability for estimating the dynamics of the systems that follow the discrete-time Markov chain. Nonlinear Bayesian filtering techniques are often applied for underwater maneuvering state estimation applications by following state-space methodology. The robustness and precision of NARX neural network are efficiently investigated for accurate state prediction of the passive Markov chain highly maneuvering underwater target. A continuous coordinated turning trajectory of an underwater maneuvering object is modeled for analyzing the performance of the neural computing paradigm. State estimation modeling is developed in the context of bearings only tracking technology in which the efficiency of the NARX neural network is investigated for ideal and complex ocean environments. Real-time position and velocity of maneuvering object are computed for five different cases by varying standard deviations of white Gaussian measured noise. Sufficient Monte Carlo simulation results validate the competence of NARX neural computing over conventional generalized pseudo-Bayesian filtering algorithms like an interacting multiple model extended Kalman filter and an interacting multiple model unscented Kalman filter.

Keywords: neural computing; state estimation; Markov chain; turning trajectory; bearings only tracking; maneuvering object



Citation: Ali, W.; Li, Y.; Raja, M.A.Z.; Khan, W.U.; He, Y. State Estimation of an Underwater Markov Chain Maneuvering Target Using Intelligent Computing. *Entropy* **2021**, *23*, 1124. <https://doi.org/10.3390/e23091124>

Academic Editors: Friedhelm Schwenker and Stelios Bekiros

Received: 11 May 2021

Accepted: 17 June 2021

Published: 29 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Real-time state estimation of maneuvering object has captured a lot of interest from the research community recently [1]. The phenomena of state prediction found its extensive significance in many practical civilian and defense applications like sonar, radar, indoor tracking, surveillance, unmanned aerial vehicle, smart transportation network, drone navigation, wireless radio network, air traffic control, and bioinformatics [2,3]. Despite its wide range of usage in numerous engineering problems, efficient state prediction still has plenty of room for improvement and various challenges to deal with [4]. Ambiguity in designing the complex maneuvering in state vector of the object and incorrectly measured bearings are the main factors which create complications in state approximation of a maneuvering vehicle [5]. The real-time kinetics of maneuvering targets are often designed by dynamic and measurement models through numerical expression. These mathematical equations of maneuvering targets describe the behavior of motion parameters like speed, velocity, position, course, turning rate and explain how they switch from their present

points in real-time [6]. In the case of maneuvering state estimation, a single dynamic model is unable to describe all random maneuvering behaviors of underwater passive vehicles. In the single model, true movement and state vector of the target can be mismatched, and this largely affects the overall performance of the state estimation mechanism [7]. The state prediction models which compute the numerical expressions of target dynamics at each point of trajectory are considered as global and best models [8].

In literature, a lot of valuable research efforts have been done for achieving efficient state prediction of underwater maneuvering objects [9]. A single model Kalman filtering algorithm appeared as the initial attempt in this domain; however, its prediction accuracy is often largely affected by random and complex maneuverings of the underwater object [10]. After that, a conventional full statistic model was proposed by researchers for improving the performance of state approximation in which second-order terms of acceleration are approaching the first-order Markov chain [11]. Although the accuracy of state estimation phenomena improved by a filtering technique based on this model for small maneuverings; meanwhile, the accuracy of this method diverges at constant velocity and high maneuverings of underwater objects. Additionally, the maneuvering process needs more prior statistical information about the target than what usually exists. In the last decade, decision-based algorithms also attracted a lot of attention from the research community which seems better for detecting high maneuverings of the underwater vehicle and handling them efficiently [12]. The variable dimension filter and input estimation algorithms are major examples of this methodology [13–15].

The dynamics of the underwater maneuvering target can be efficiently designed with generalized pseudo-Bayesian (GPB) based interacting multiple model (IMM) algorithms, which model the maneuverings of target by using a set of sub-filters [16,17]. However, transition probability information and assumed dynamic models are already known in IMM state estimation techniques [18]. Practically, in GPB algorithms, model switching probabilities are hard to achieve, and it is also difficult to break the dynamics of the target into different multi motion models [19]. Furthermore, computational cost significantly rises with increasing the number of motion models which seriously affects the real-time performance of the overall state estimation scheme. Nevertheless, IMM methodologies have some mutual drawbacks like unnecessary mathematical computations in the case of maneuvering objects and divergence of accuracy because of over motion modeling at maneuvering instants [20]. On the other hand, the maneuvering state estimation problem can be efficiently designed with time series by following state-space modeling [21]. In all the discussed approaches, an independent and identical white Gaussian process noise sequence is considered in the state model of maneuvering objects [22].

Smart computing techniques based on NARX neural networks appeared as an emerging field of study in the last two decades for almost every field of engineering [23,24]. In artificial intelligence computing methodologies, the strength of optimization algorithms and robustness of neural networks are exploited for solving different complex real-life problems [25,26]. Neural intelligence computing schemes are finding their extensive applications in several branches of practical sciences like astrophysics [27], plasma physics [28], atomic physics [29], thermodynamics [30], fluid dynamics [31], electric motors [32], rotating electrical machine [33], electromagnetic [34], meteorology [35], nonlinear optics [36], mathematical equations [37], bioinformatics [38], and nanotechnology [39]. These soft computing schemes performed exceptionally well in the above-mentioned applications and took place of all conventional methodologies [40], while, in correspondence with other traditional techniques, neural networks have produced much better efficiency and convergence rate in all related studies.

The issue of nonlinearity and ambiguity is a major challenge and hard to handle in several real-life practical systems, whose dynamics are time-varying and also depend upon their current state. The NARX neural network paradigm shows a satisfactory convergence rate and can be beneficial for the above-mentioned systems [41]. In literature, time-series data based dynamic inputs are effectively handled with this neural intelligence computing

technique [42–44]. Application of a neural network for approximating time series is an unsteady approach, where initial knowledge about time series data is not necessary. Maneuvering state prediction of an underwater turning object is typically a nonlinear study in which motion parameters of the target are frequently varied with respect to time.

A NARX based neural intelligence scheme is mathematically designed in our proposed study for efficiently estimating the real-time state of an underwater maneuvering target in different ocean atmospheres. The time series framework of complex passive bearings from a two-dimensional maneuvering object is comprehensively examined in this work for approximating motion parameters. Different underwater cases for turning and high maneuvering objects are assumed to develop a neurocomputing scheme as a novel application for accurate state prediction in the proposed work. The robustness, strength, and precision of neural intelligence computing are equated with our recently recorded work [16], in which a similar state approximation model is investigated with GPB based algorithms like an interacting multiple model extended Kalman filter (IMMEKF) and an interacting multiple model unscented Kalman filter (IMMUKF). Probabilistic variations of measured noise in the Gaussian environment are used to design ideal and cluttered undersea scenarios to compare the performance of the proposed scheme with conventional filtering methodologies. Position, velocity, and turn rate based state estimation is conducted in this work by exploiting the potency of NARX neurocomputing strategy. A detailed and briefed graphical abstract of the proposed study is shown in Figure 1. The salient features of the conducted study are outlined as:

- (1) The strength and potency of the NARX based neurocomputing paradigm are extensively investigated for accurate state approximation of a maneuvering underwater Markov chain target.
- (2) State estimates, position error, velocity error, turn rate estimates, error histogram, and regression analysis of NARX are computed for passive turning targets and compared with conventional nonlinear and multiple model variants of the Kalman filter like IMMEKF and IMMUKF.
- (3) Motion features of the kinetic target in the highly maneuvering trajectory are designed by utilizing the firmness of the well-known Wiener process velocity (WPV) model and the coordinated turn (CT) model.
- (4) The standard deviation of measurement noise is selected as an evaluation criterion, and its numerical values are varied in simulations for analyzing the trend of given techniques.
- (5) All the given algorithms are compared on the basis of a minimum mean square error (MSE), which is chosen as the performance matrix and simulation outcomes show that the accuracy of the NARX based neural network is far better from multiple model Kalman filters for estimating the real-time state of an underwater maneuvering object.

The rest of the study is organized in the following pattern. Section 2 provides bearings only tracking (BOT) based multiple model maneuvering state estimation system design in two-dimensional Cartesian coordinates. Comprehensive mathematical modeling of continuously turning object is also developed in this section. The structure and working principle of the NARX neural intelligence network are explained in Section 3 with the process of training, testing, and validation. Simulation outcomes and discussion in the context of minimum MSE of given methods are described in Section 4. The last section of the proposed work reveals the major contributions and defines further goals for future research.

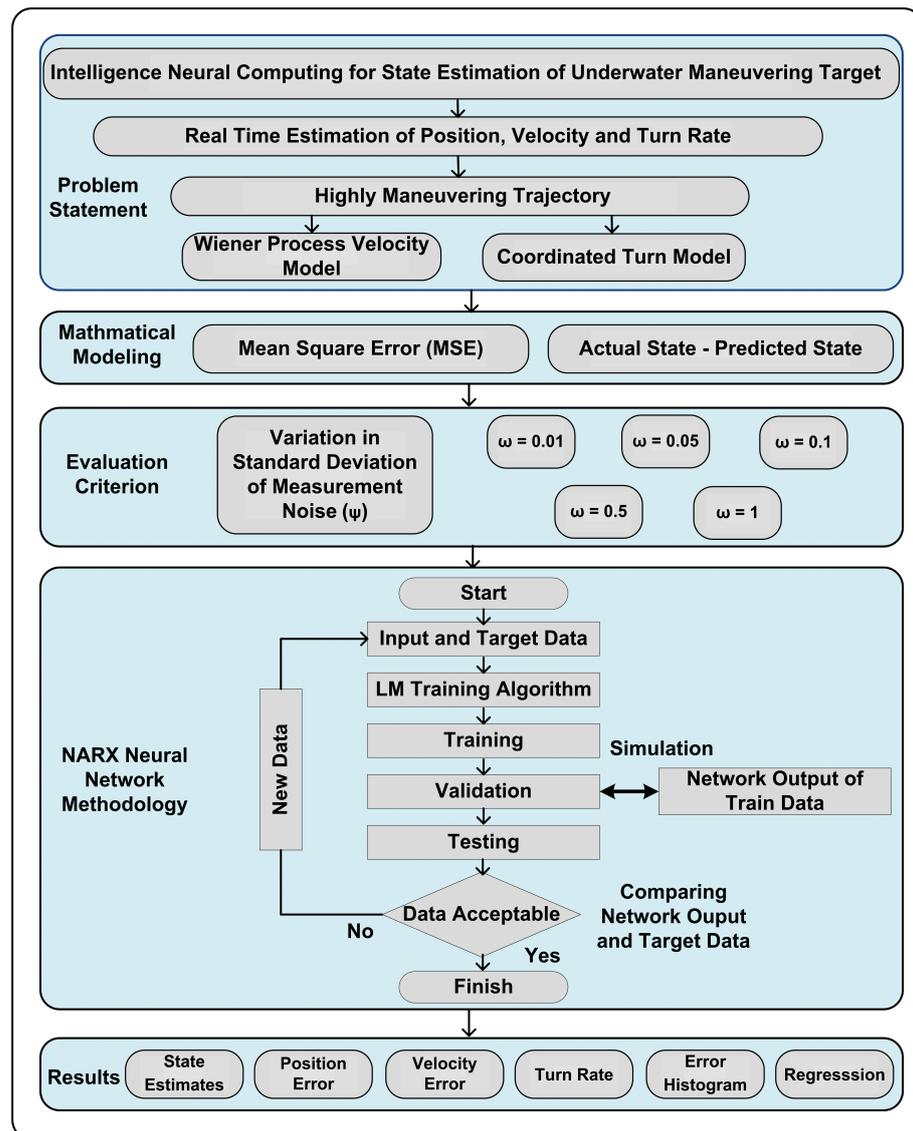


Figure 1. Overall flow chart of the proposed scheme.

2. Markov Chain Maneuvering State Estimation System Model

A rectangular coordinates' two-dimensional state prediction system modeling of the Markov chain maneuvering target is mathematically designed in this section of the study. The complex ocean medium is assumed in this scheme for precise real-time state prediction of a continuous turning object by deploying state space-based BOT phenomena. Passive noisy bearings from the maneuvering object at each time instant are collected from eight acoustic sensors, and these sensors are regarded as observer platforms. The sensors are deployed along a line to involve the idea of Uniform Linear Array (ULA) and to obtain maximum benefits of ULA in maneuvering target tracking. This nonlinear complex data collected at observers is called the bearing of the moving object, and it is largely based on the position of each sensor. The arrangement of observers is done in such a manner that there is an equal distance between each observer. A continuous turning trajectory of the underwater target is assumed in this proposed Markov chain maneuvering state estimation system, which is being estimated by an intelligence neural computing scheme. The movements of maneuvering object and estimation architecture are illustrated in Figure 2.

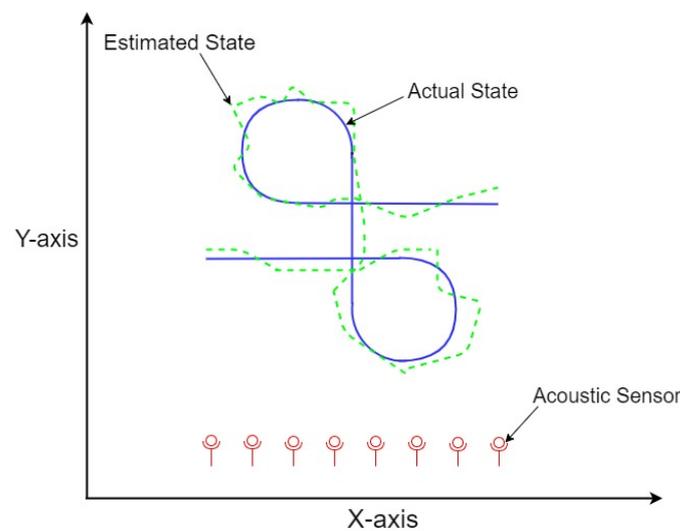


Figure 2. Maneuvering state estimation framework.

The modeling parameters of many practical systems fluctuate with respect to time. One particular system model is unable to define these varying system parameters. In real-time state estimation problems, there is always the probability that modeling parameters might change during the prediction process. These types of systems are referred to as Markov chain or multiple model systems. In these scenarios, the selection of a single system model may lead to divergence of the overall scheme. Consequently, it is necessary to design a generic state model of a moving object which can have the properties of different system models. Here, in this study, WPV and CT models are applied for defining the dynamics of the underwater maneuvering target.

2.1. Wiener Process Velocity (WPV) Model

The instantaneous state of the maneuvering object at time step t with position (x_t, y_t) and velocity (\dot{x}_t, \dot{y}_t) in two-dimensional Cartesian coordinates is shown in state vector \mathbf{Y}_t^a as:

$$\mathbf{Y}_t^a = [x_t^a \quad y_t^a \quad \dot{x}_t^a \quad \dot{y}_t^a]^T. \tag{1}$$

In the meantime, the state vector at the receiving station in Cartesian coordinates can be defined as:

$$\mathbf{Y}_t^b = [x_t^b \quad y_t^b \quad \dot{x}_t^b \quad \dot{y}_t^b]^T. \tag{2}$$

The respective state vector of the maneuvering object and base station is given as:

$$\mathbf{Y}_t = \mathbf{Y}_t^a - \mathbf{Y}_t^b = [x_t \quad y_t \quad \dot{x}_t \quad \dot{y}_t]^T. \tag{3}$$

Dynamics of maneuvering object are designed according to the discrete-time WPV model in the domain of state-space phenomena. According to this model, the state equation can be modeled as:

$$\mathbf{Y}_{t+1} = \mathbf{A}_t \mathbf{Y}_t + \tilde{\zeta}_t^P. \tag{4}$$

In the above given state equation, the state transition matrix is represented with \mathbf{A}_t having dimensions of $i \times i$. In the state-space methodology, the state transition matrix describes the feedback of the dynamic model, while process noise $\tilde{\zeta}_t^P$ is assumed in Gaussian distribution with zero mean. The distribution of state transition matrix \mathbf{A}_t with sampling space Δt is assumed as follows:

$$\mathbf{A}_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

and this sampling space Δt is defined as:

$$\Delta t = [(t + 1) - t]. \quad (6)$$

The state space model explained in Equation (4) must be written in discrete time for precise prediction of motion parameters with the NARX scheme. The reason for choosing a discrete-time kinetic model is that this modeling can better evaluate the performance of the system for time instant t , which is a multiple of sampling space Δt . The updated form of state equation in discrete time with its appropriate parameters is given as:

$$\mathbf{Y}_{t+1} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_t} \underbrace{\begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix}}_{\mathbf{Y}_t} + \zeta_t^{\mathbf{P}}. \quad (7)$$

The Gaussian distributed process noise $\zeta_t^{\mathbf{P}}$ can be modeled with covariance \mathbf{B}_t as:

$$\zeta_t^{\mathbf{P}} \approx \mathcal{N}(0, \mathbf{B}_t). \quad (8)$$

For achieving relatively slow turns of the object's trajectory, the value of process variance is taken 0.05 in simulations.

2.2. Coordinated Turn (CT) Model

A specific dynamic model that is commonly used for defining the motion parameters of the maneuvering target, which is frequently moving in a turning track is known as the CT model. In this dynamic modeling, the state vector holds a supplementary parameter termed as the turn rate. For the CT model, the state vector of the maneuvering object in Cartesian coordinates for its position, velocity and turn rate can be written as:

$$\mathbf{Y}_t^a = [x_t^a \ y_t^a \ \dot{x}_t^a \ \dot{y}_t^a \ \varphi_t^a]^T. \quad (9)$$

In addition, the state vector at the observer platform can be designed as:

$$\mathbf{Y}_t^b = [x_t^b \ y_t^b \ \dot{x}_t^b \ \dot{y}_t^b \ \varphi_t^b]^T. \quad (10)$$

Meanwhile, a corresponding state vector among the maneuvering object and observer platform is given as:

$$\mathbf{Y}_t = \mathbf{Y}_t^a - \mathbf{Y}_t^b = [x_t \ y_t \ \dot{x}_t \ \dot{y}_t \ \varphi_t]^T. \quad (11)$$

The state equation in the CT model is defined as:

$$\mathbf{Y}_{t+1} = \mathbf{A}_t \mathbf{Y}_t + \zeta_t^{\mathbf{P}}. \quad (12)$$

The above state equation in the CT model is almost identical as in the WPV model given in (4) with an extra parameter, which actually explains the transfer function of Gaussian distributed process noise. Likewise, the WPV model, the discrete form of the state equation for the CT model is illustrated below as:

$$\mathbf{Y}_{t+1} = \underbrace{\begin{bmatrix} 1 & 0 & \frac{\sin(\varphi_t \Delta t)}{\varphi_t} & \frac{\cos(\varphi_t \Delta t) - 1}{\varphi_t} & 0 \\ 0 & 1 & \frac{1 - \cos(\varphi_t \Delta t)}{\varphi_t} & \frac{\sin(\varphi_t \Delta t)}{\varphi_t} & 0 \\ 0 & 0 & \cos(\varphi_t \Delta t) & -\sin(\varphi_t \Delta t) & 0 \\ 0 & 0 & \sin(\varphi_t \Delta t) & \cos(\varphi_t \Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_t} \mathbf{Y}_t + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\zeta_t^{\mathbf{P}}}. \quad (13)$$

In this dynamic modeling, white Gaussian noise is also assumed with zero mean, and its covariance is similar as that given in Equation (8):

$$\zeta_t^P \approx N(0, \mathbf{B}_t). \quad (14)$$

The mathematical computations of the CT model are given in matrix form while its behavior is nonlinear. Thus, the CT model can be designed with five mathematical expressions as:

$$x_{t+1} = x_t + \frac{\sin(\varphi_t \Delta t)}{\varphi_t} \dot{x}_t + \frac{\cos(\varphi_t \Delta t) - \Delta t}{\varphi_t} \dot{y}_t, \quad (15)$$

$$y_{t+1} = y_t + \frac{1 - \cos(\varphi_t \Delta t)}{\varphi_t} \dot{x}_t + \frac{\sin(\varphi_t \Delta t)}{\varphi_t} \dot{y}_t, \quad (16)$$

$$\dot{x}_{t+1} = \cos(\varphi_t \Delta t) \dot{x}_t - \sin(\varphi_t \Delta t) \dot{y}_t, \quad (17)$$

$$\dot{y}_{t+1} = \sin(\varphi_t \Delta t) \dot{x}_t + \cos(\varphi_t \Delta t) \dot{y}_t, \quad (18)$$

$$\varphi_{t+1} = \varphi_t + \zeta_t^P. \quad (19)$$

The variance of Gaussian process noise ζ_t^P for turn rate φ_t is set to $Y_\varphi = 0.15$ in simulations for assuming relatively high maneuverings.

2.3. Measurement Model

In this state estimation modeling, both WPV and CT models have the same measurement model, which is also designed with the principle of state-space phenomena. The mathematical expression of the measurement model at time step t is given as:

$$\mathbf{X}_{t+1} = \mathbf{H}(\mathbf{Y}_{t+1}, \zeta_{t+1}^M). \quad (20)$$

Real-time passive maneuverings from the turning object at time step t are integrated in a $\mathbf{H}(\cdot)$ matrix, which is also known as measurement function. It contains complex measurements by following the methodology of the point-slope tangent relationship. In the above measurement model, independent Gaussian distributed measurement noise is denoted with ζ at time step t . Passive measurements at sensors are obtained with the real position of a dynamic target and orientation of sensors by using the following relationship as:

$$\mathbf{H}(\mathbf{Y}_{t+1}) = \underbrace{\text{arctangent} \left[\frac{y_t - \Lambda_y^e}{x_t - \Lambda_x^e} \right]}_{\text{bearings}}. \quad (21)$$

In two-dimensional Cartesian coordinates, the real-time position of maneuvering target is represented with (x_t, y_t) while localization of sensors e is denoted with $(\Lambda_x^e, \Lambda_y^e)$ in the above measurement function. Measurement model \mathbf{X} given in (20) can be written in an updated form for acoustic sensors e at time step t as:

$$\mathbf{X}_t^e = \text{arctangent} \left[\frac{y_t - \Lambda_y^e}{x_t - \Lambda_x^e} \right] + \zeta_t^{eM}. \quad (22)$$

Measurement noise ζ_t^{eM} in the above relationship has zero mean, while its covariance \mathbf{T} can be computed as:

$$\zeta_t^{eM} \approx N(0, \mathbf{T}_t), \quad (23)$$

$$\mathbf{T}_t = \text{diag}(\omega_\chi^2). \quad (24)$$

In Equation (24), the standard deviation of measurement noise is denoted with ω_χ , which plays a vital role in analyzing the accuracy of state estimation techniques in the maneuvering target tracking framework. Variation of the standard deviation of measurement noise defines the behavior of the undersea atmosphere. In our study, different values of the

standard deviation of measurement noise are chosen for investigating the convergence and robustness of the neural intelligence scheme and GPB algorithms. The coordinated turning trajectory of the underwater object is developed by following these steps.

- (1) Initial location of maneuvering object is the origin $(0, 0)$, having a constant velocity of 1 towards x -axis e.g., $(\dot{x}_t, \dot{y}_t) = (1, 0)$.
- (2) A right turn is taken by target after 4 s of starting point with turn rate $\varphi_t = -1$.
- (3) The target ends turning right at the total time of 9 s and moves straight with a constant velocity of 1 for the next 2 s.
- (4) At the total time of 11 s, the underwater target maneuver for the left turn with turning parameter $\varphi_t = 1$.
- (5) At the total time of 16 s, the target ends turning on the left side and moves straight with a similar velocity for 4 s.

3. Intelligent Neural Computing

In this section, we designed an NARX neural intelligence mathematical model for the efficient state prediction of the kinetic target in an underwater medium. In neural computing, synchronization between supplementary external information and designed time series cannot be ignored. In the problems of parameter estimation, the behavior of the applied method is associated with noise depraved bearing. Hence, the information of measurement noise or preceding data are usually incorporated for time series modeling to find the satisfactory accuracy.

3.1. Nonlinear Autoregressive with the Exogenous Input (NARX) Neural Scheme

For the mathematical modeling of NARX neural intelligent computing, supervised learning of neurons is used for finding the next information in time series by precisely integrating preceding values. In time-series data, the external input and second output are responsible for estimating future values in the designed structure of the NARX neural computing. Therefore, input, output, delay, and hidden layers are combined for designing a multilayer architecture of the NARX model. The time series of state equation $\mathbf{Y}(t)$ is estimated for n preceding points of actual state \mathbf{Y} with external input series of measurements $\mathbf{X}(t)$ for a maneuvering object using the delay factor of m . For estimating required time series, hidden nodes of NARX neural scheme are designed as:

$$\mathbf{Y}(t) = k \begin{pmatrix} \mathbf{X}(t-1), \mathbf{X}(t-2), \dots, \mathbf{X}(t-m) \\ \mathbf{Y}(t-1), \mathbf{Y}(t-2), \dots, \mathbf{Y}(t-n) \end{pmatrix} + \boldsymbol{\eta}(t). \quad (25)$$

In the above designed nonlinear model of NARX neural intelligence learning, measurements $\mathbf{X}(t)$ is a j -dimensional additional time series while \mathbf{Y} is denoting output time series with dimensions of k . The error of neural structure is $\boldsymbol{\eta}(t)$, while output delays are represented with m and n , respectively:

$$q(t) = f_{\rho_1}(\mathbf{X}(t)w_{\rho_1} + \gamma_{\rho_1}), \quad (26)$$

$$\mathbf{Y}(t) = f_{\rho_2}(q(t)w_{\rho_2} + \gamma_{\rho_2}). \quad (27)$$

In the above equations, the hidden layer vector is denoted with $q(t)$ having dimensions of ρ . Weights between multilayers are represented by w_{ρ_1} and w_{ρ_2} correspondingly. However, thresholds of hidden layers and input layers are represented with γ_{ρ_1} and γ_{ρ_2} accordingly. The network function of hidden nodes is f_{ρ_1} , and the activation response of output nodes is shown by f_{ρ_2} . A multiple layer paradigm of NARX technique is depicted in Figure 3.

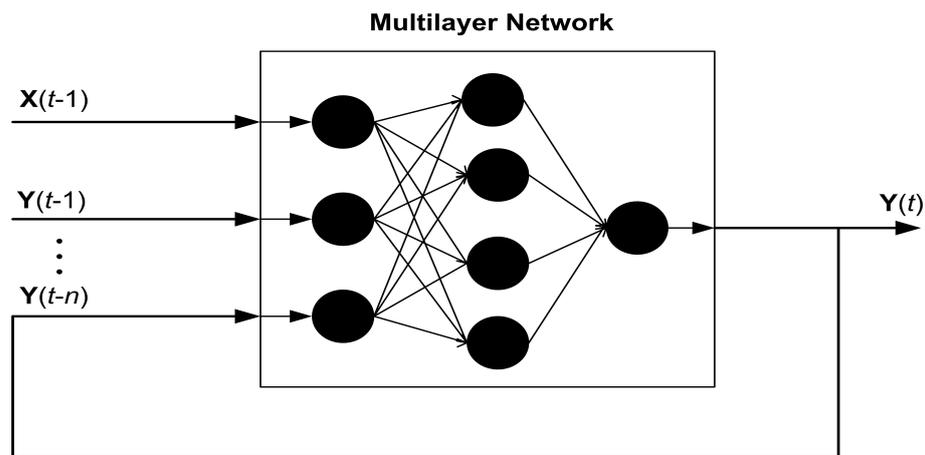


Figure 3. The multilayer structure of the NARX neural network.

3.2. Architecture of an NARX Neural Scheme

It is important to note that NARX is a recursive discrete-time neural network, which consists of a time series and can be designed as:

$$Y(t+1) = k \left(\begin{matrix} Y(t), Y(t-1), \dots, Y((t-c)+1) \\ X(t-(n+1)), \dots, X((t-c)-(n+1)) \end{matrix} \right). \tag{28}$$

The time delay parameter n is assumed null in the above equation. Based upon this assumption, the updated form of NARX model is shown as:

$$Y(t+1) = k \left(\begin{matrix} Y(t), Y(t-1), \dots, Y(t-c, +1) \\ X(t), X(t+1), \dots, X(t-c, +1) \end{matrix} \right). \tag{29}$$

The above function is expressed in vector form as:

$$Y(t+1) = k(Y(t), X(t)). \tag{30}$$

In the above vector form, input and output regress are represented with vectors $X(t)$ and $Y(t)$ correspondingly. For the efficient training of the NARX neural model, we applied a Levenberg–Marquardt (LM) based algorithm. The detailed design of NARX neural strategy is represented in Figure 4.

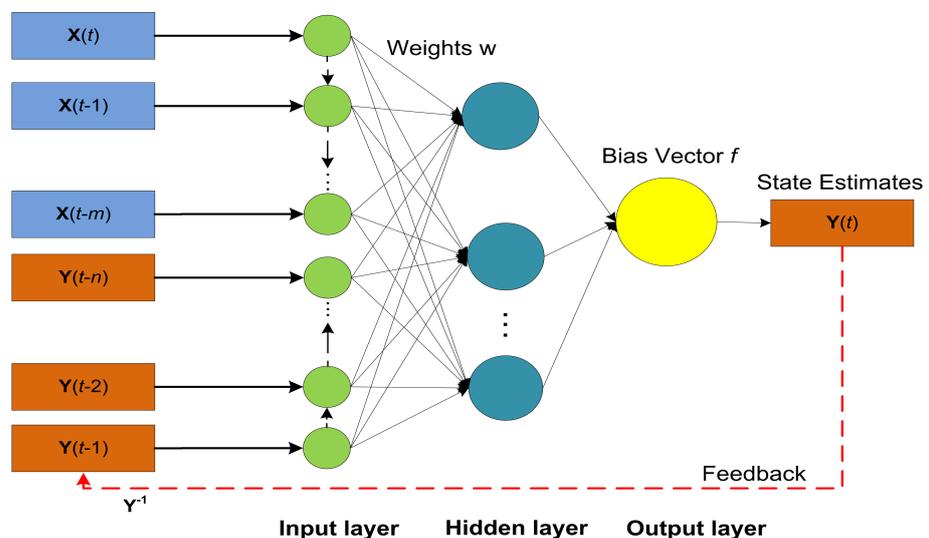


Figure 4. Architecture of the NARX neural network.

3.3. Levenberg–Marquardt (LM) Training Method

The LM method is a well-known training setup found in literature for designing an NARX neural network. It is also known as the fastest learning mechanism for the formulation of an algorithm. Its training efficiency is far better from other training algorithms because it computes the second-order derivatives without solving the Hessian matrix. The two scientists Donald Marquardt and Kenneth Levenberg introduced the concept of the LM training method. Minima of measurement function $\mathbf{J}(\mathbf{X})$ is calculated initially in the LM technique with the summation of squares for nonlinear functions as:

$$\mathbf{J}(\mathbf{X}) = \sum_{\alpha=1}^{\beta} \frac{1}{2} [j_{\alpha}(\mathbf{X})]^2. \quad (31)$$

The Hessian function \mathbf{P} and gradient parameter g are computed as:

$$\mathbf{P} = \mathbf{S}_t^T \mathbf{S}_t, \quad (32)$$

$$g = \mathbf{S}_t^T \boldsymbol{\eta}_t. \quad (33)$$

In the above Hessian matrix and gradient equation, \mathbf{S} is representing the Jacobian matrix. Here, we combined the first-order derivatives of neural network error in correspondence with weights and biases, whereas the overall error of the NARX neural model for all training samples is shown with $\boldsymbol{\eta}_t$. The Jacobian matrix is only used for necessary computations in the LM training scheme. However, the means of the squared errors are combined in the performance function. The search space for the LM training technique is calculated in the following order as:

$$(\mathbf{S}_t^T \mathbf{S}_t + \wp_t \mathbf{I}) \boldsymbol{\theta}_t = -\mathbf{S}_t^T \boldsymbol{j}_t. \quad (34)$$

In the given search space function, an identity matrix is represented with \mathbf{I} and positive scalars are denoted with \wp_t giving increment of $\boldsymbol{\theta}_t$. Therefore, the principle of updating weights w_t in the training mechanism is expressed as:

$$w_{t+1} = w_t - \frac{\mathbf{S}_t \boldsymbol{\eta}_t}{(\mathbf{S}_t^T \mathbf{S}_t + \wp_t \mathbf{I})}. \quad (35)$$

3.4. Performance Evaluation Criterion

At every time instant t , the least mean square position error between a true and approximated position for turning object is formulated as an evaluation criterion for neural intelligence computing methodology. In this study, it shows the precision and convergence for a neural intelligence technique. On the other hand, the MSE function of NARX, IMMEKF, and IMMUKF are formulated for each independent Monte Carlo simulation as:

$$\text{MSE}(t) = \frac{1}{N} \sum_{t=1}^N \left\| \mathbf{Y}_t^{\text{True}} - \mathbf{Y}_t^{\text{Est}} \right\|^2 \quad (36)$$

In the above MSE function, we denote the true state for turning objects as $\mathbf{Y}_t^{\text{True}}$, while an approximated state of the target with NARX and filtering techniques is represented by $\mathbf{Y}_t^{\text{Est}}$. However, the total number of data points is shown by N which are 200 in simulations, whereas $t = 1$ is the initial data point. These position errors are computed at each time step t for the turning trajectory.

4. Simulation Results and Discussion

In this section of the study, simulation results in the sense of real-time state approximates, position error, velocity error, turn rate estimates, error histogram, and regression are briefly explained for proposed NARX based neural computing.

The standard deviation of measured Gaussian noise is chosen as a performance parameter and numerically varied from 0.01 to 1 radian in simulations for designing five different cases. The highest value of measured noise $\omega = 1$ radian is showing a cluttered ocean environment, while minimum value $\omega = 0.01$ radian depicts the ideal atmosphere. In simulations, there are many mathematical variables in state estimation phenomena that need to be accurately adjusted for acquiring the required performance. Suitable values of state estimation parameters are given in Table 1.

Table 1. Appropriate setting of different state estimation variables.

Parameters	Appropriate Values
Starting state of the target	$Y_0 = [0 \ 0 \ 1 \ 0 \ 0]^T$
Localization function of sensors	$(\Lambda_x^e, \Lambda_y^e)$
Total sensors	$e = 8$
Distance between sensors	0.5
Values of Standard deviation of measured noise	$\omega = 0.01 \rightarrow 1$ radians
Process noise variance of WPV model	$b_1 = 0.05$
Process noise variance of CT model	$Y_\varphi = 0.15$
Sampling space	$\Delta t = 0.1$ s
Input and output delays	$m, n = 2$
Total sample points	200
Hidden neurons	100
NARX target steps	1000

Passive bearings combined at acoustic observers are denoted with $X(t)$ and the actual state of maneuvering target represented with $Y(t)$ are applied to NARX neural network as inputs for computing approximated state $Y(t)$. The neural network toolbox for NARX in the MATLAB software package is used for the conduction of the simulations. The toolbox structure of NARX Model is based on input, hidden, and output layer as shown in Figure 5.

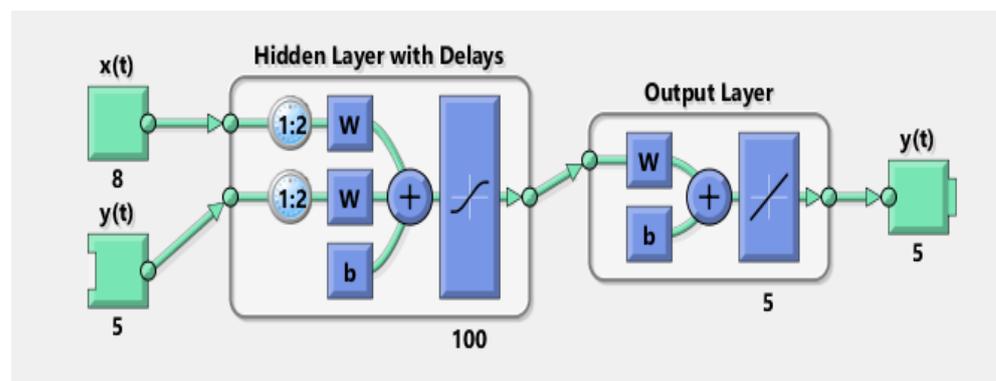


Figure 5. Architecture of the NARX neural network.

The actual state vector in Equation (11), which consists of five entities like x - y positions, x - y velocities, and the turning parameter is applied on one input while passive measurements from eight sensors are applied as another input to the NARX model for obtaining the estimated state vector of five elements. In simulations, 100 hidden neurons are used in the hidden layer while a sigmoid function is used for activation of these neurons. The process of training weights for NARX is done with the LM method by using backpropagation through time methodology. The epoch mode strategy is conducted in the training process of the neural network. In the designing of the NARX neural network, 70% of the data of the complete time series is used for the training process, while the remaining 30% is divided equally in the validation and testing procedure of the obtained results.

4.1. State Estimation Analysis of Markov Chain Maneuvering Target for Different Cases of the Standard Deviation of Measurement Noise

In this subsection, MATLAB simulations' results and their brief discussion are explained for real-time state prediction of the maneuvering object by applying IMMEKF, IMMUKF, and NARX neural intelligence computing. Accuracy and convergence based performance analysis of NARX is compared with IMM filters for five particular variations of the standard deviation of measured noise. In every case, state estimates, position error, velocity error, turn rate estimates, error histogram and regression analysis of GPB algorithms and NARX neural computing are presented from Figures 6–15. These five different cases are discussed below with their mathematical expressions and simulation outcomes.

4.1.1. Case 1: The Standard Deviation of Measurement Noise = 0.01 Radian

In the first case, the standard deviation of measured Gaussian noise ω is selected as a 0.01 radian that is depicting nearly an ideal and smooth underwater environment. Although covariance in this case is computed from the standard deviation of measurement noise as:

$$\mathbf{T}_t = \text{diag}(\omega_{\chi}^2), \quad (37)$$

while measurement noise at time instant t for e sensors is calculated from covariance as:

$$\zeta_t^{eM} \approx N(0, \mathbf{T}_t). \quad (38)$$

For each passive bearing at time step t for sensor e , above the computed measurement noise, is integrating with measurement model \mathbf{X} as:

$$\mathbf{X}_t^e = \text{arctangent} \left[\frac{y_t - \Lambda_y^e}{x_t - \Lambda_x^e} \right] + \zeta_t^{eM}. \quad (39)$$

The above designed measurement model \mathbf{X}_t^e is applied at one input of the NARX neural model as input time series data. Another input of the NARX model is the target's time series where the true state vector defined below is applied for obtaining an estimated state of underwater maneuvering target:

$$\mathbf{Y}_t = \mathbf{Y}_t^a - \mathbf{Y}_t^b = [x_t \quad y_t \quad \dot{x}_t \quad \dot{y}_t \quad \varphi_t]^T. \quad (40)$$

For this value of the standard deviation of measured noise in case 1, real-time state estimates, average true and estimated position and velocity error, true and estimated turn rate, error histogram, and regression analysis of real and predicted turning path are given as:

- In Figure 6a, a state estimation performance of NARX for a tracking turning trajectory of the maneuvering object is equated with GPB algorithms based on IMMEKF and IMMUKF, and it is quite vibrant that NARX is accurately tracing the actual turning path of the maneuvering target.
- Position error analysis in mean square sense is presented in Figure 6b in which NARX is showing the least error among other techniques.
- In Figure 6c, the error between true velocity and estimated velocity that is also presented in the context of MSE is shown. In this result, the NARX neural scheme is efficiently estimating true velocity compared to multiple model Kalman filters for all data points.
- Turn rate estimates of IMMEKF, IMMUKF, and NARX are illustrated in Figure 6d. Estimates of turning parameter are also validating the effectiveness of NARX over IMM filters for all data points in complete turning trajectory.
- Error histogram between target time series $\mathbf{Y}(t-1), \mathbf{Y}(t-1), \dots, \mathbf{Y}(t-n)$, and estimated output data \mathbf{Y}_t^{Est} are illustrated in Figure 6e. An error histogram consists of a set of error values that can be negative, and these define how much difference there is between estimated and target time series. In this analysis, the overall error of the

NARX neuro network is divided into 20 bins that are shown in vertical bars. At the start of the histogram, a bin is unique from others having an error of 0.00017 at the height of 350 instances indicating that several points in time series have errors in this range. The zero error bar also falls on this bin which defines the zero error of the neural network.

- Regression analysis of the NARX scheme for training, validation, and testing procedure is shown in Figure 6f. The time series data set in the designed neural network is distributed in training, validation, and testing with the proportion of 75%, 15%, and 15%, respectively. This analysis integrates the statistical parameters for showing the relationship between the output variable Y_t^{Est} and target variable $Y(t)$. Here, in regression results, the true target and estimated output overlay each other. A linear trend is observed in both values that show the strength of NARX neural computing.

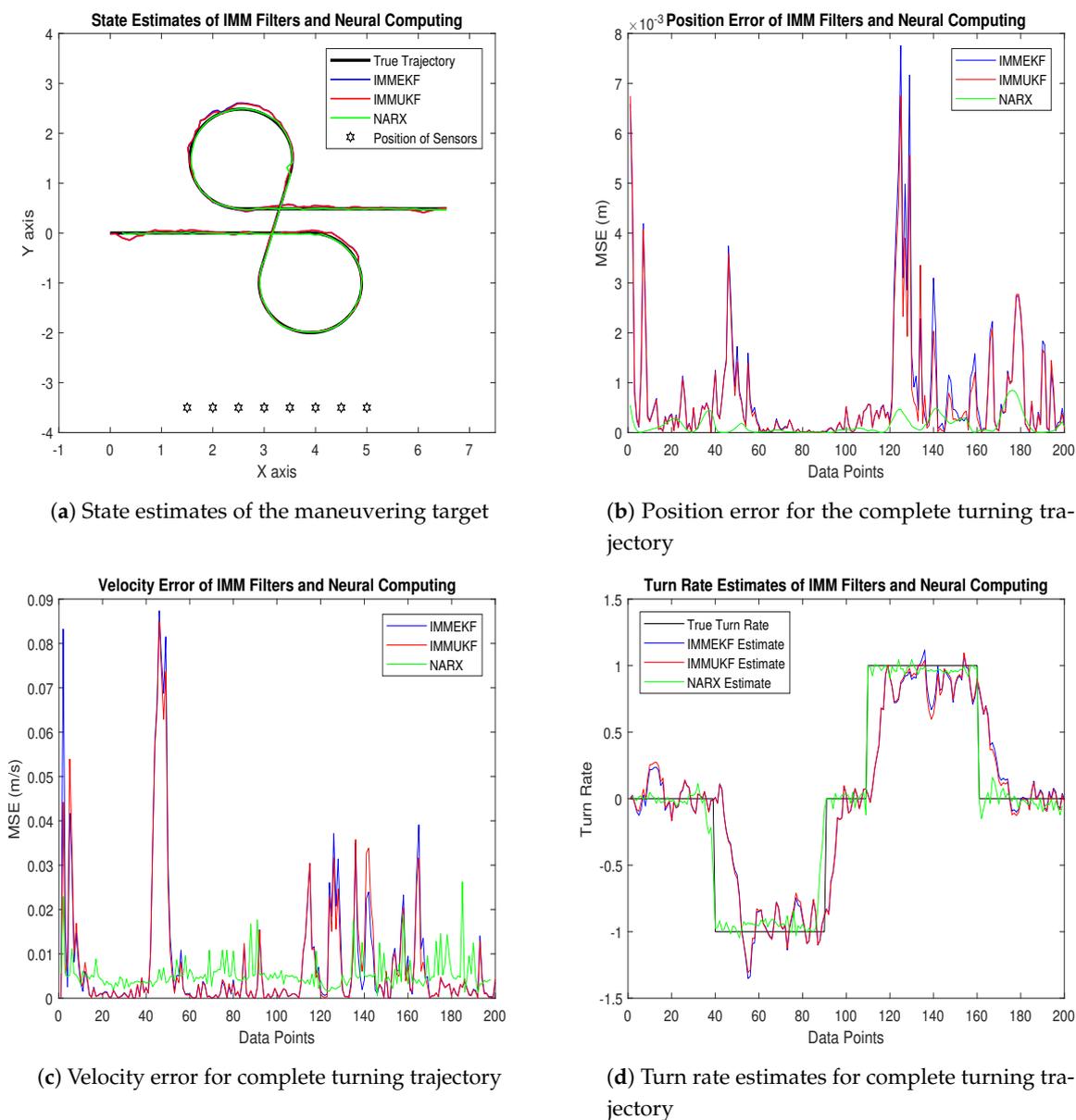
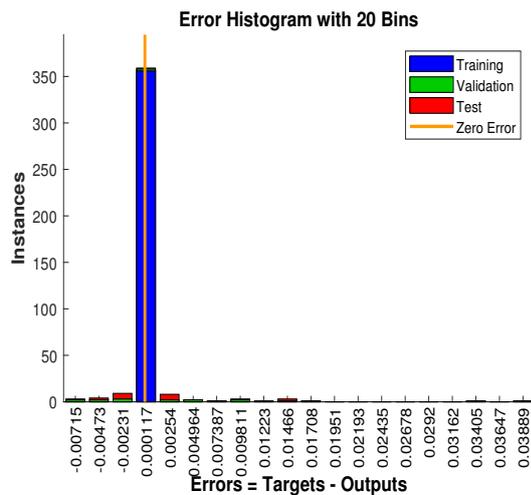
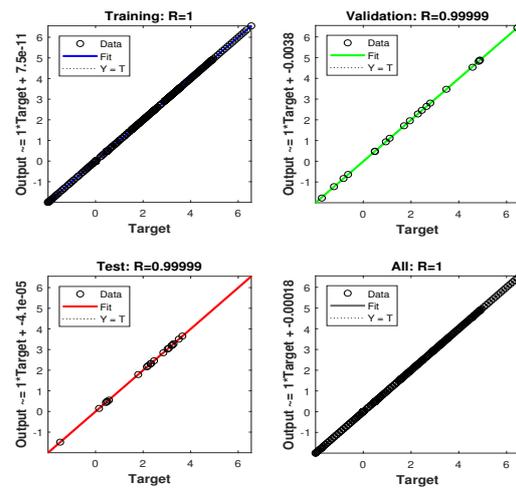


Figure 6. Cont.



(e) Error histogram between target data and output data



(f) Regression analysis of NARX

Figure 6. State estimation performance of IMMEKF, IMMUKF, and NARX for $\omega = 0.01$ radian.

We also compute position and velocity MSE between real and estimated position and velocity of the object. These position and velocity error results also satisfy the graphical simulations that the precision of NARX is better from IMM Kalman filters. Position and velocity errors calculated for IMMEKF, IMMUKF, and NARX are enlisted in Figure 7.

Position and Velocity MSE for $\omega = 0.01$ radian

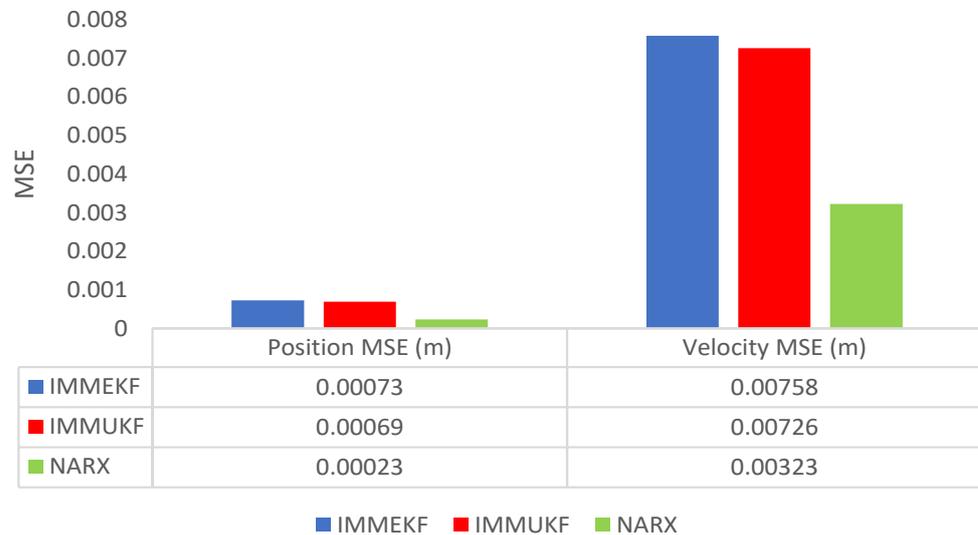


Figure 7. Average MSEs of predicting the position and velocity for IMMEKF, IMMUKF, and NARX in case 1.

4.1.2. Case 2: The Standard Deviation of Measurement Noise = 0.05 Radian

The standard deviation of measurement noise is chosen as $\omega = 0.05$ radian in case 2 for incorporating some amount of measurement noise in the estimation process. Meanwhile, covariance from this standard deviation of measurement noise is computed as:

$$T_t = \text{diag}(\omega_{\chi}^2), \tag{41}$$

Gaussian distributed measurement noise at time step t for sensor e is depending upon above calculated covariance as:

$$\zeta_t^e \approx N(0, T_t). \quad (42)$$

At each time instant t , measurement noise is including in the measurement model for sensor e as:

$$\mathbf{X}_t^e = \arctangent \left[\frac{y_t - \Lambda_y^e}{x_t - \Lambda_x^e} \right] + \zeta_t^e. \quad (43)$$

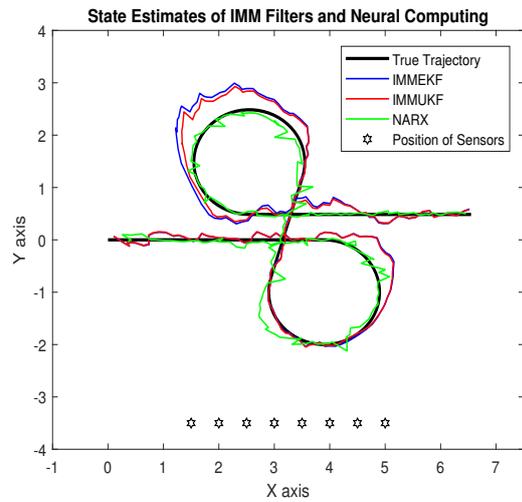
Likewise, this designed measurement model \mathbf{X}_t^e is applied to the neural network as an input time series. At other inputs of NARX, the following state vector is applied as a target time series:

$$\mathbf{Y}_t = \mathbf{Y}_t^a - \mathbf{Y}_t^b = [x_t \ y_t \ \dot{x}_t \ \dot{y}_t \ \varphi_t]^T. \quad (44)$$

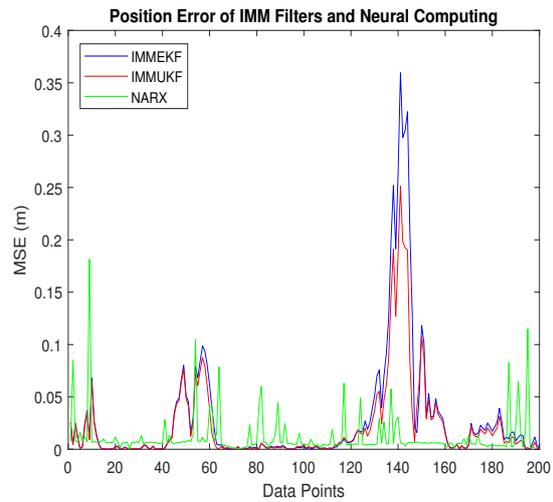
Real-time position of maneuvering object (x_t, y_t) , velocity (\dot{x}_t, \dot{y}_t) , and turn rate φ_t are actually the motion parameters which we want to estimate with NARX intelligence computing. Simulation results consisting of state estimates, position error, velocity error, turn rate estimates, error histogram, and regression analysis for this case are shown below:

- In Figure 8a, state estimates of all three techniques are compared for the coordinated turning trajectory of the maneuvering target. It is worth to note that, for this value of the standard deviation of measured noise, NARX is showing better state estimates and convergence than conventional nonlinear filtering methods.
- Average mean square position error among actual and predicted position of underwater maneuvering target is depicted in Figure 8b, which is representing the precision of NARX over IMMEKF and IMMUKF.
- In Figure 8c, true and estimated velocity of the maneuvering object estimated from IMM filters and NARX neural network is compared, and intelligence neural methodology is performing far better from filtering techniques.
- Figure 8d is showing turn rate estimates for this case of measurement noise in which again NARX is showing better estimates of turning parameter than existing filtering schemes.
- In Figure 8e, error histogram results are shown among target time series data set $\mathbf{Y}(t-1), \mathbf{Y}(t-1), \dots, \mathbf{Y}(t-n)$, and predicted output value \mathbf{Y}_t^{Est} of the object's state. In the middle part of histogram, a vertical bin consists of an error of 0.002741, and the length of this for the training dataset lies around 250 instances, while the validation and testing dataset lie between 200 and 250 instances. In this case, zero error lies beneath the vertical bar with center 0.002741.
- In Figure 8f, the regression of NARX neural intelligence computing is given for training, validation, and testing procedure. In regression analysis, the effectiveness of the NARX neural network is depicted by linear behavior and the head-to-head response of the true target and estimated output.

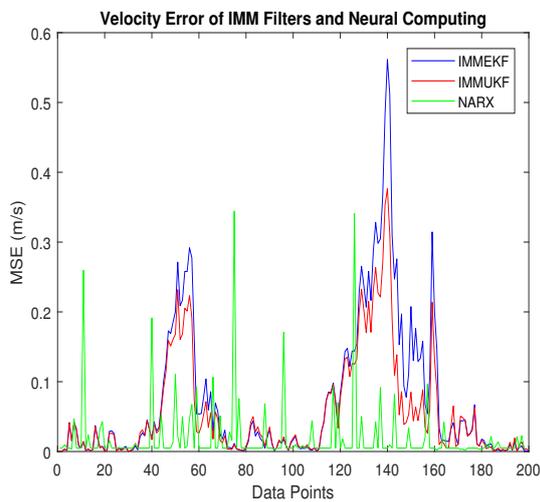
Position and velocity MSEs between actual and predicted position and velocity of turning target are computed for this variation of measurement noise. These position and velocity errors verify the previous results that estimation performance of NARX is higher than IMM Kalman filters, and it proves the strength of the neural methodology in state prediction of the maneuvering object. These position and velocity errors computed from IMMEKF, IMMUKF, and NARX are enlisted in Figure 9.



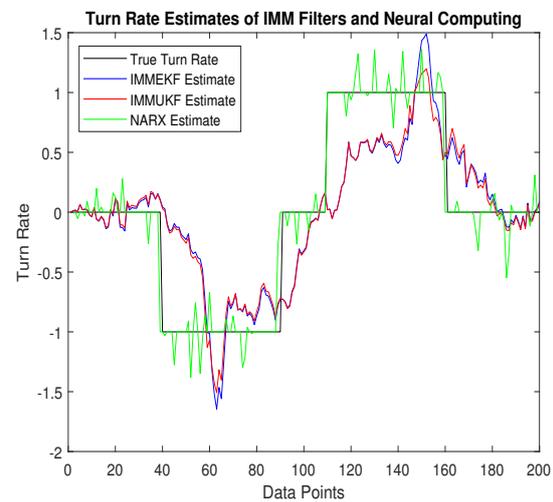
(a) State estimates of Maneuvering target



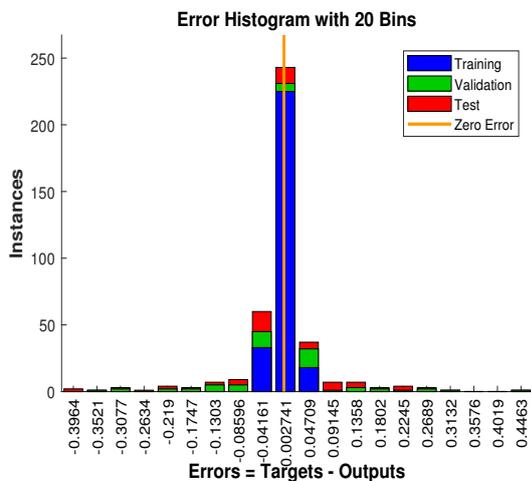
(b) Position error for complete turning trajectory



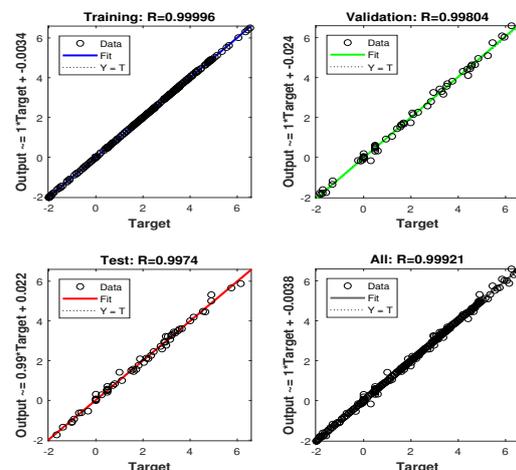
(c) Velocity error for complete turning trajectory



(d) Turn rate estimates for complete turning trajectory



(e) Error histogram between target data and output data



(f) Regression analysis of NARX

Figure 8. State estimation performance of IMMEKF, IMMUKF, and NARX for $\omega = 0.05$ radian.

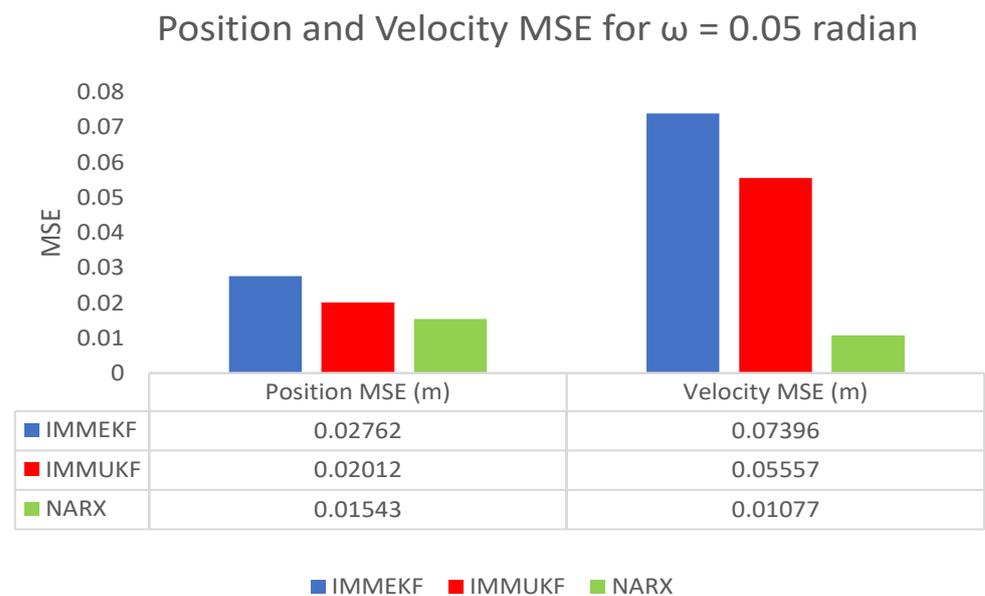


Figure 9. Average MSEs of predicting the position and velocity for IMMEKF, IMMUKF, and NARX in case 2.

4.1.3. Case 3: The Standard Deviation of Measurement Noise = 0.1 Radian

The standard deviation of measured Gaussian noise is increased to $\omega = 0.1$ radian in this case representing that there is enough noise added in the system. The covariance T at time instant t from 0.1 radian standard deviation of Gaussian measurement noise is calculated as:

$$T_t = \text{diag}(\omega_{\bar{x}}^2), \tag{45}$$

Measurement noise from the above developed covariance for e sensor at time step t is defined as:

$$\zeta_t^{eM} \approx N(0, T_t). \tag{46}$$

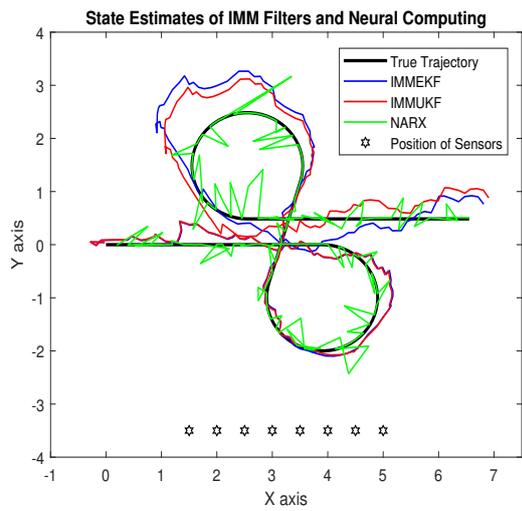
Now, this generated amount of Gaussian measured noise is incorporating with the system model as:

$$X_t^e = \text{arctangent} \left[\frac{y_t - \Lambda_y^e}{x_t - \Lambda_x^e} \right] + \zeta_t^{eM}. \tag{47}$$

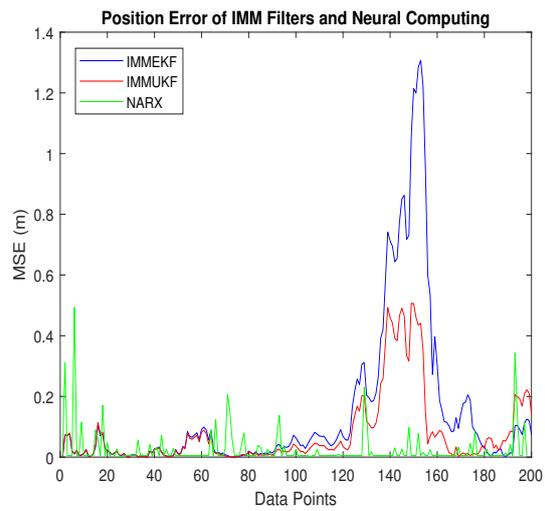
Measurement model X at time step t for sensor e is given in the above expression where passive bearings from all installed acoustic sensors are merging with white Gaussian distributed measurement noise. This measurement model expression is applied to the neural network model as an input dataset, while another input of the neural network consists of a given state vector of motion parameters:

$$Y_t = Y_t^a - Y_t^b = [x_t \quad y_t \quad \dot{x}_t \quad \dot{y}_t \quad \varphi_t]^T. \tag{48}$$

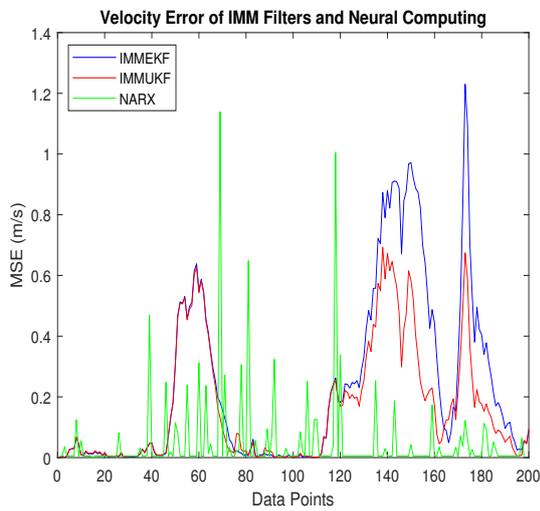
Simulation results for this variation of measurement noise in the shape of state estimates, least position error, velocity error, turning parameter estimates, error histogram, and regression analysis are shown below:



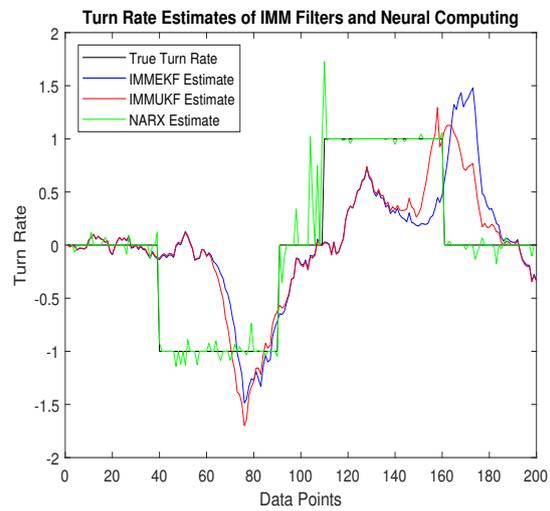
(a) State estimates of Maneuvering target



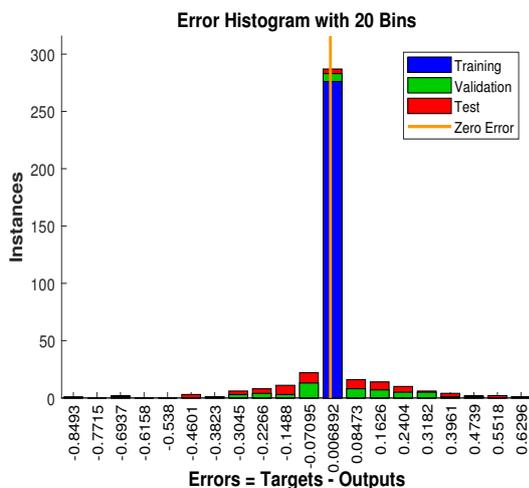
(b) Position error for complete turning trajectory



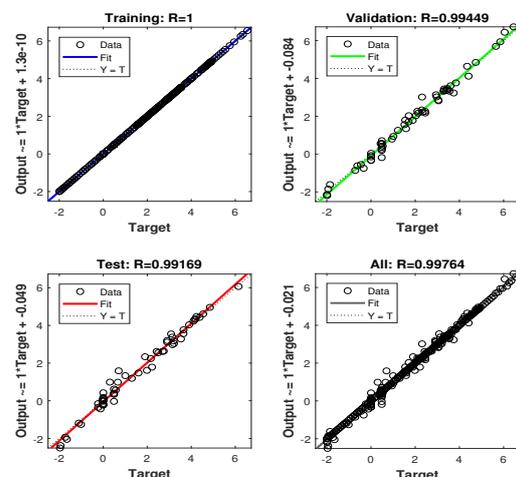
(c) Velocity error for complete turning trajectory



(d) Turn rate estimates for complete turning trajectory



(e) Error histogram between target data and output data



(f) Regression analysis of NARX

Figure 10. State estimation performance of IMMEKF, IMMUKF, and NARX for $\omega = 0.1$ radian.

- State prediction results in coordinated turn trajectory for all methods are given in Figure 10a, where the NARX based neural intelligence computing technique is showing better accuracy over conventional IMMEKF and IMMUKF. It can be seen that multimodel filters are experiencing more difficulty to estimate the state of the maneuvering object at turns of the trajectory than NARX, which is depicting the competency of the neural paradigm.
- Real-time positions errors of IMM filters and NARX are shown in Figure 10b in the context of the mean square. In this result, the accuracy of NARX is also better from other methods for all samples of turning trajectory.
- In Figure 10c, the velocity error of maneuvering object from all algorithms is shown in meter per second for all data points. The middle of the trajectory performance of NARX is not satisfactory, but, at the start and at the end of data points, NARX shows better performance than IMM filters.
- Turn rate estimates of all algorithms are presented in Figure 10d in which the estimation performance of neural computing is far better from Kalman filters.
- A comparative analysis among target time series $Y(t-1), Y(t-1), \dots, Y(t-n)$, and the estimated value Y_t^{Est} of target's dynamics in the form of the histogram is represented in Figure 10e. A vertical bar incorporating an error of 0.006892 is shown in the center of the histogram. The height of this vertical bin for the training process is near 300 instances, whereas validation and testing dataset appears between 250 and 300 instances. Zero error for this analysis lies under the vertical bin with a value of 0.006892.
- Regression phenomena in graphical form for the procedures of training, validation, and testing are explained in Figure 10f. In this graph, a minimum divergence between target and output value is noted, which is because of an increment in the standard deviation of measured noise.

MSEs of position and velocity in meters and m/sec correspondingly between actual and predicted position and velocity of the target are also computed in this case. These position and velocity errors also approve the graphical results that precision of the neural network is far better with respect to multimodel filters. These position and velocity errors computed from IMMEKF, IMMUKF, and NARX are represented in Figure 11.

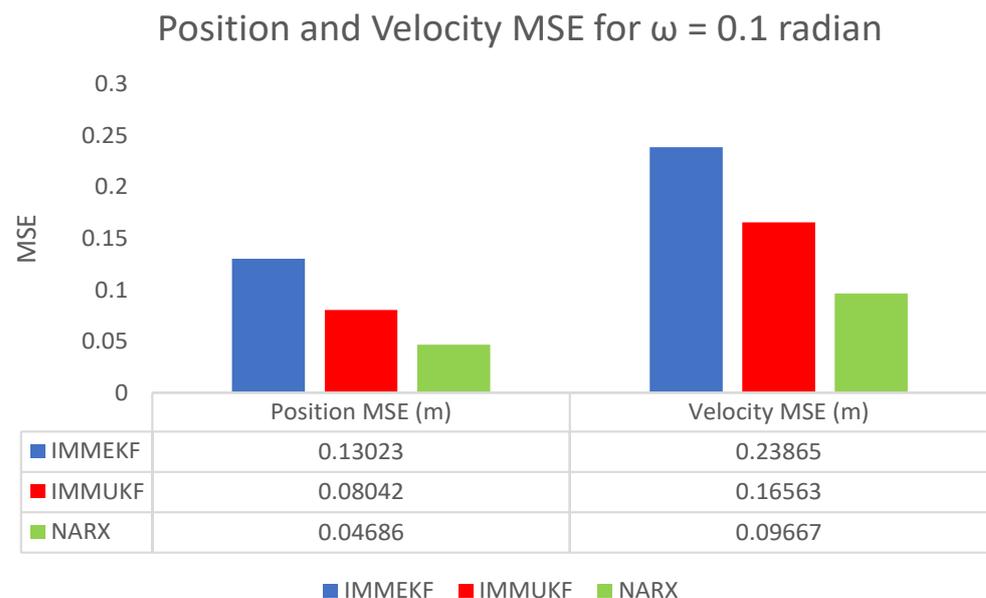


Figure 11. Average MSEs of predicting the position and velocity for IMMEKF, IMMUKF and NARX in case 3.

4.1.4. Case 4: The Standard Deviation of Measurement Noise = 0.5 Radian

In case 4, the numerical value of measured noise is increased to $\omega = 0.5$ radian, which is representing a sufficient quantity of Gaussian noise that is incorporated in the state estimation system. The mathematical expression of covariance designed with this measured noise is expressed as:

$$\mathbf{T}_t = \text{diag}(\omega_{\mathbf{x}}^2), \quad (49)$$

while measurement Gaussian noise is computed from covariance as:

$$\zeta_t^{eM} \approx N(0, \mathbf{T}_t). \quad (50)$$

Measured noise at times step t for e sensor is added in the measurement equation of the system as:

$$\mathbf{X}_t^e = \arctangent \left[\frac{y_t - \Lambda_y^e}{x_t - \Lambda_x^e} \right] + \zeta_t^{eM}. \quad (51)$$

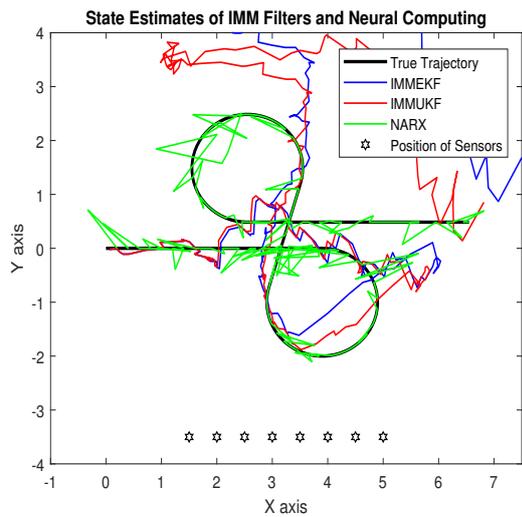
The complete measurement model equation involving passive bearings and measurement noise is given to one input of the NARX neural model as an input time series \mathbf{X}_t^e , while another input called target time series is the actual state vector is given below:

$$\mathbf{Y}_t = \mathbf{Y}_t^a - \mathbf{Y}_t^b = [x_t \ y_t \ \dot{x}_t \ \dot{y}_t \ \varphi_t]^T. \quad (52)$$

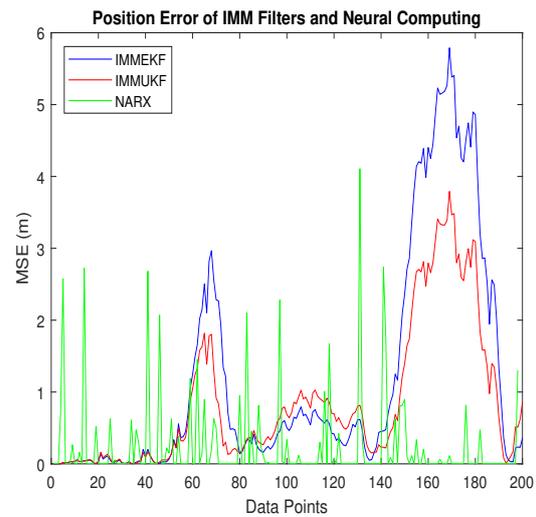
These real-time positions, velocities, and turn rates of kinetic passive maneuvering objects are developed for the absolute turning path of underwater targets and given in NARX neural design as target data for predicting these required motion parameters. In Figure 12, simulation results of state estimates, position error, velocity error, turn rate estimates, error histogram, and regression are shown as:

- In Figure 12a, state estimates of IMMEKF, IMMUKF, and NARX are presented in which the amount of measurement noise is high so all algorithms are experiencing difficulties to follow the real trajectory. However, even in the noisy atmosphere, it is seen that estimates of NARX are approaching a real trajectory more than the two other given algorithms.
- Average MSE among actual and predicted position of the maneuvering object is represented in Figure 12b, which is also showing that all algorithms have enough amount of error, but from a comparative point of view, NARX is estimating turning trajectory with less position error than IMMEKF and IMMUKF.
- In Figure 12c, velocity error from all methodologies is shown, which is also validating the strength of the neural network.
- Turn rate estimates in this case are presented in Figure 12d in which the estimation performance of neural computing is also better from IMM filtering methods.
- An error histogram among target values $\mathbf{Y}(t-1), \mathbf{Y}(t-1), \dots, \mathbf{Y}(t-n)$, and output estimated value \mathbf{Y}_t^{Est} of target's state is shown in Figure 12e. An error value -0.03387 in a vertical bin appears at the center of the histogram which has a height of close to 300 data points for the training dataset, while validation and testing datasets fall among 250 and 300 data sets. The zero error line on the error axis lies beneath the vertical with center -0.03387 in the given error histogram.
- Regression in this case is given in Figure 12f which is showing a sufficient amount of divergence between the true target and estimated output because of the increase in the standard deviation of measured noise.

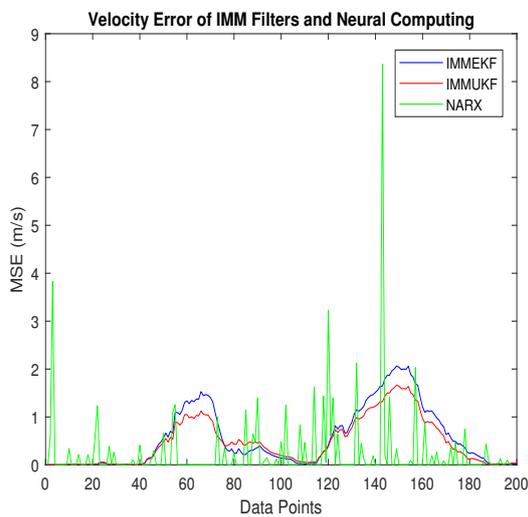
The position and velocity MSEs between actual and predicted position and velocity of the maneuvering target are also calculated in this case. These position and velocity error responses also verify previous results that NARX is showing better performance from Kalman filters. These position and velocity errors computed from IMMEKF, IMMUKF, and NARX are defined in Figure 13.



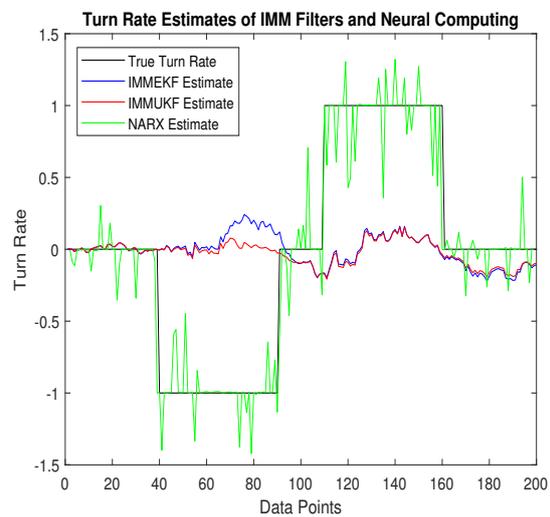
(a) State estimates of the maneuvering target



(b) Position error for complete turning trajectory



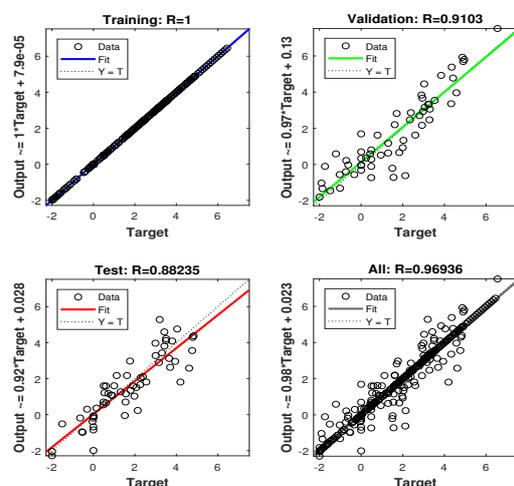
(c) Velocity error for complete turning trajectory



(d) Turn rate estimates for complete turning trajectory



(e) Error histogram between target data and output data



(f) Regression analysis of NARX

Figure 12. State estimation performance of IMMEKF, IMMUKF, and NARX for $\omega = 0.5$ radian.

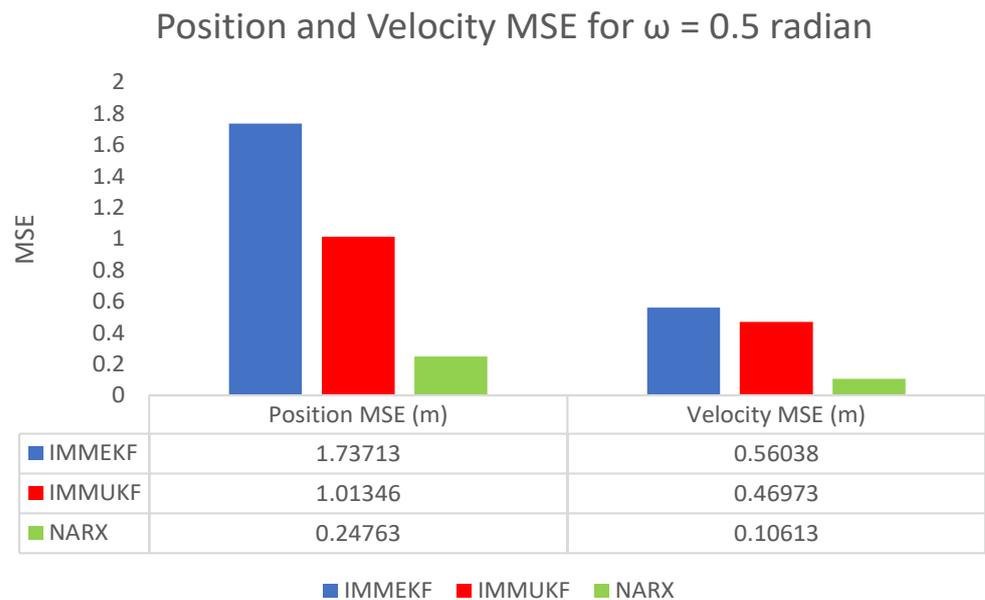


Figure 13. Average MSEs of predicting the position and velocity for IMMEKF, IMMUKF, and NARX in case 4.

4.1.5. Case 5: The Standard Deviation of Measurement Noise = 1 Radian

In the last case of this study, we take a maximum value $\omega = 1$ radian for assuming an extremely noisy atmosphere, and the state estimation system is facing a noisy measurement model. For this maximum value of ω , covariance T_t is defined as:

$$T_t = \text{diag}(\omega^2_{\chi}), \tag{53}$$

At time instant t , Gaussian distributed measurement noise ζ_t^{eM} is computed for e sensor from covariance T_t as:

$$\zeta_t^{eM} \approx N(0, T_t). \tag{54}$$

Finally, this extreme value of independent white Gaussian measurement noise is included in measurement expression as:

$$\mathbf{X}_t^e = \text{arctangent} \left[\frac{y_t - \Lambda_y^e}{x_t - \Lambda_x^e} \right] + \zeta_t^{eM}. \tag{55}$$

These passive measurements incorporated with maximum measurement noise are used as input time series \mathbf{X}_t^e of NARX neural architecture, while the target time series of the neural network is taken from an actual state vector, which is defined below:

$$\mathbf{Y}_t = \mathbf{Y}_t^a - \mathbf{Y}_t^b = [x_t \quad y_t \quad \dot{x}_t \quad \dot{y}_t \quad \varphi_t]^T. \tag{56}$$

The NARX based neural network is combining input time series and target time series to efficiently estimate output time series, which consist of the predicted state vector. In this case, results of state estimates, position error, velocity error, turn rate estimates, error histogram, and regression are shown in Figure 14.

- State estimates of IMMEKF, IMMUKF, and NARX in this case are presented in Figure 14a. State estimation results from all algorithms are diverging because of an extremely noisy underwater environment. All algorithms are facing severe difficulties in tracking true trajectory, but, in comparison, neural computing methodology NARX is performing better to estimate turning trajectory than the other two algorithms in this case.

- Average real-time error among the actual and predicted position of underwater maneuvering object is represented in Figure 14b in the form of MSE, which is clearly indicating that the NARX neural network is producing less position error as compared to IMMEKF and IMMUKF.
- In Figure 14c, velocity error results from all techniques are shown in which NARX is experiencing some large peaks but overall has better performance from IMM filters.
- Turn rate estimates in this case are representing in Figure 14d in which NARX is better estimating the turning parameter.
- An analysis of neural network in the form of error histogram is represented in Figure 14e among target dataset $Y(t - 1), Y(t - 1), \dots, Y(t - n)$, and predicted output Y_t^{Est} . A vertical bin is observed having an error of 0.1166 with a height near 250 steps for training while validation and testing datasets have a height between 200 and 250 steps. The zero error point lies at 0.1166 beneath the vertical bin.
- The regression of the NARX neural network between target and estimated output in this case is represented in Figure 14f. In regression results, a sufficient divergence is observed among the target and estimated results. This is because of a higher value of Gaussian noise involved in the state estimation system.

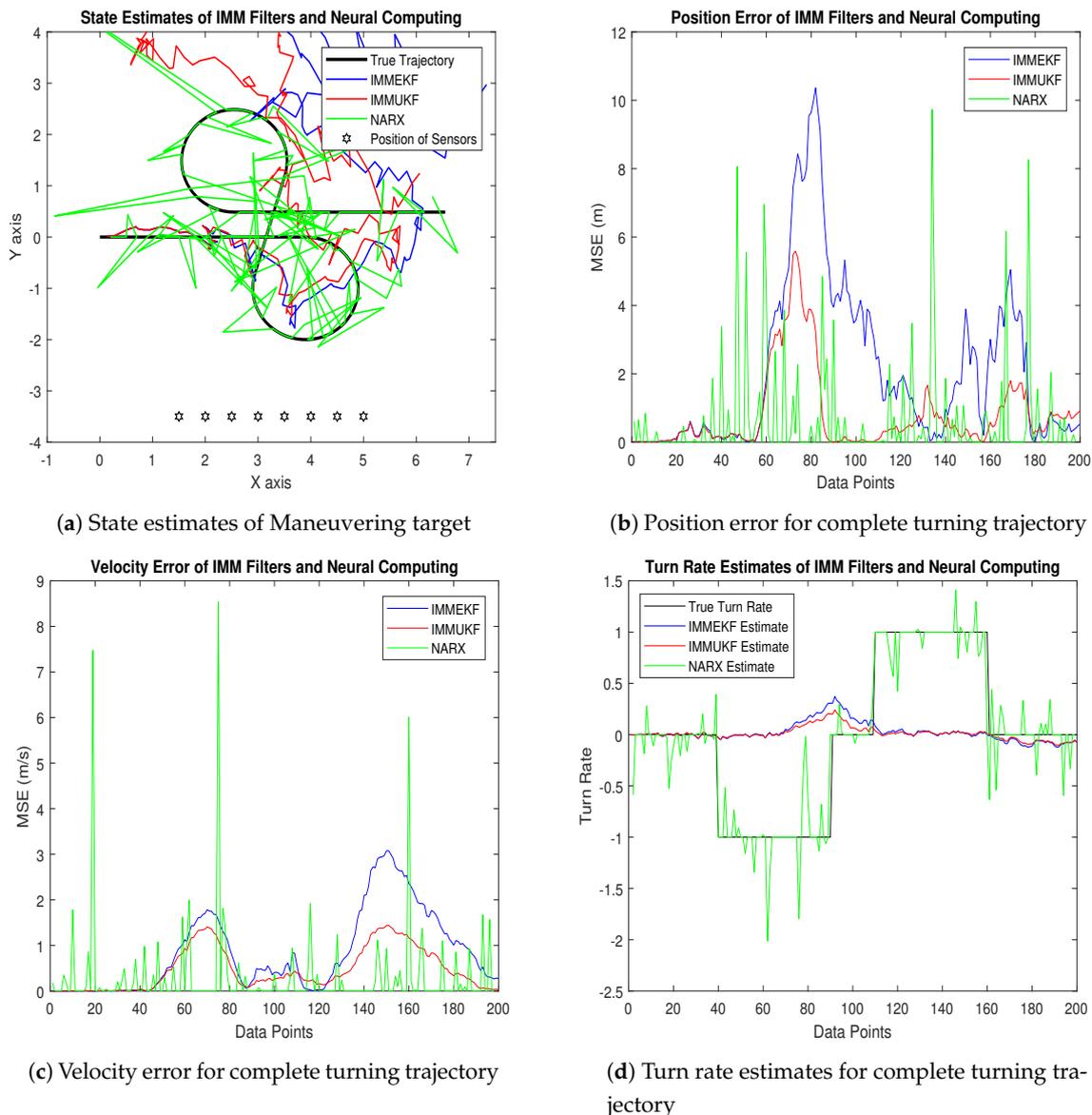
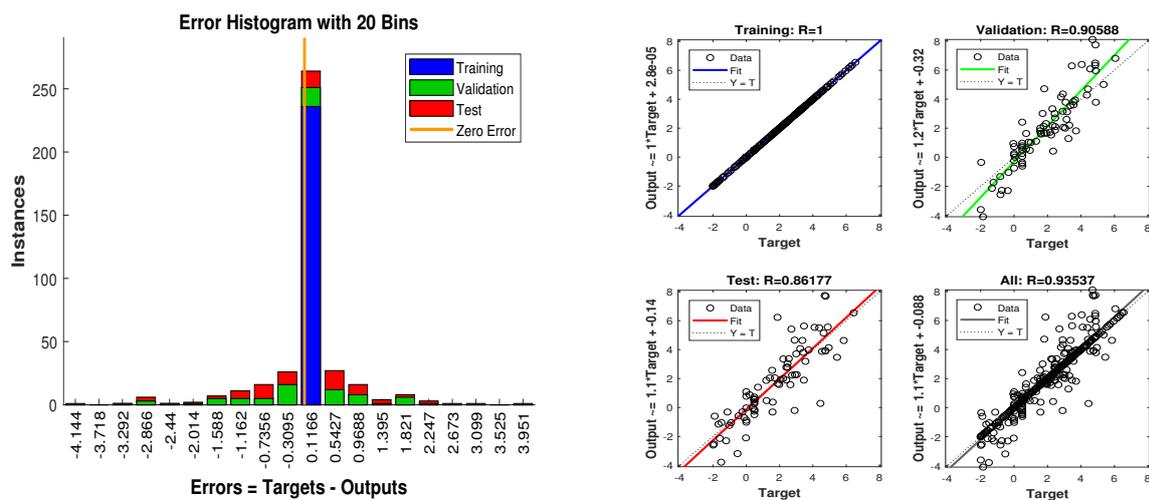


Figure 14. Cont.



(e) Error histogram between target data and output data

(f) Regression analysis of NARX

Figure 14. State estimation performance of IMMEKF, IMMUKF, and NARX for $\omega = 1$ radian.

An extreme noisy environment position and velocity MSEs between actual and predicted position and velocity of the target are calculated in meters and m/s, respectively. These position and velocity error results confirm previous results that the accuracy of NARX is better from Kalman filters even in an extremely noisy underwater environment. Position and velocity errors computed from IMMEKF, IMMUKF, and NARX are enlisted in Figure 15.

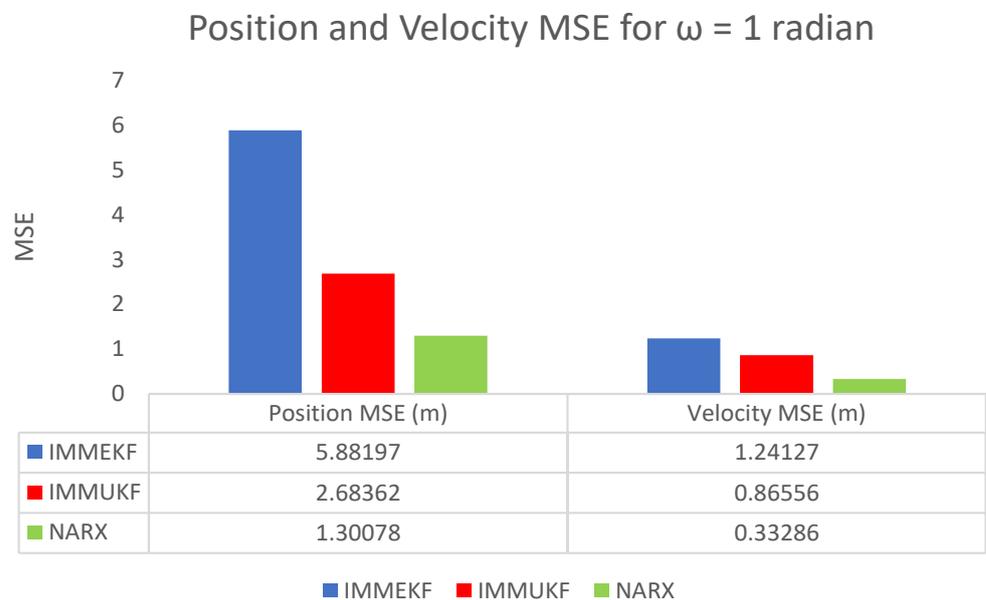


Figure 15. Average MSEs of predicting the position and velocity for IMMEKF, IMMUKF, and NARX in case 2.

The above figures of all simulation results prove that all state prediction methods are facing problems with following the real state of turning maneuvering objects at higher arithmetic values of measurement noise ω in an underwater atmosphere. While in a comparative point of view among all methods, NARX based neural intelligence computing is performing far better, which is proving its effectiveness for nonlinear real-time state prediction applications in the underwater scenario.

5. Conclusions

In this paper, we investigated the neural intelligence computing paradigm based on NARX for real-time state estimation of a bearings only maneuvering Markov chain underwater object. At each time instant, the instantaneous positions of a continuously turning object are estimated in the bi-dimensional Cartesian geometric system. Initially, the state space-based estimation framework for a dynamic and measurement model using BOT phenomena are designed mathematically. Then, the neural computing technique based on NARX is designed for efficient state estimation of a passive maneuvering Markov chain target. In this context, the NARX neural strategy is examined in MATLAB for 200 data points, and we assessed the robustness of neural computation for an absolute turning trajectory of object motion in the sense of turn rate estimation, current position estimation, real-time state estimation, and velocity error along with error histogram and regression. Later, enough variations of standard deviation of measurement Gaussian noise are used for examining the performance of the designed method. State estimation results endorsed the competency of the NARX neural technique as compared with typical nonlinear multiple model Bayesian filtering methods like IMMUKF and IMMUKF. However, an exponential decay in the simulation results of all methods is observed in a noisy underwater scenario. Therefore, determining precise state prediction performance in a cluttered sea atmosphere is still a challenge.

In the future, this work can be extended by investigating the radial base function (RBF) neural methodology for getting more accurate state estimation of a constant velocity multiple model turning single or multi objects by applying non-Gaussian noise [45,46], which is an interesting research domain in the field of acoustic signal processing. In addition, in future studies, a detailed complexity analysis can be conducted for analyzing time delay between NARX neural computing and filtering strategy.

Author Contributions: Conceptualization, W.A., M.A.Z.R., and Y.L.; Methodology, W.A. and M.A.Z.R.; Software, W.A.; Investigation, W.A. and Y.L.; Resources, Y.L. and Y.H.; Writing—original draft preparation, W.A.; Writing—review and editing, W.U.K., Y.L., Y.H., and M.A.Z.R.; Visualization, W.A. and M.A.Z.R.; Supervision, Y.L. M.A.Z.R., and Y.H.; Project administration, Y.L.; Funding acquisition, Y.H. and W.U.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant No. 51977153, 51977161, 51577046, the State Key Program of National Natural Science Foundation of China under Grant No. 51637004, the national key research and development plan “important scientific instruments and equipment development” Grant No. 2016YFF0102200, Equipment research project in advance Grant No. 41402040301, Wuhan Science and Technology Plan Project Grant No. 20201G01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Konrad, T.; Gehrt, J.-J.; Lin, J.; Zweigel, R.; Abel, D. Advanced state estimation for navigation of automated vehicles. *Annu. Rev. Control* **2018**, *46*, 181–195. [[CrossRef](#)]
2. Hu, S.; Zhou, J.; Chen, C. State estimation for dynamic systems with unknown process inputs and applications. *IEEE Access* **2018**, *6*, 14857–14869. [[CrossRef](#)]
3. Hu, J.; Liu, G.-P.; Zhang, H.; Liu, H. On state estimation for nonlinear dynamical networks with random sensor delays and coupling strength under event-based communication mechanism. *Inf. Sci.* **2020**, *511*, 265–283. [[CrossRef](#)]
4. Smiley, A.J.; Harrison, W.K.; Plett, G.L. Postprocessing the outputs of an interacting multiple-model Kalman filter using a Markovian trellis to estimate parameter values of aged Li-ion cells. *J. Energy Storage* **2020**, *27*, 101043. [[CrossRef](#)]
5. Zhang, H.; Li, L.; Xie, W. Constrained multiple model particle filtering for bearings-only maneuvering target tracking. *IEEE Access* **2018**, *6*, 51721–51734. [[CrossRef](#)]

6. Marzbani, H.; Khayyam, H.; To, C.N.; Quoc, Đ.V.; Jazar, R.N. Autonomous vehicles: Autodriver algorithm and vehicle dynamics. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3201–3211. [[CrossRef](#)]
7. Khalid, S.S.; Abrar, S. A low-complexity interacting multiple model filter for maneuvering target tracking. *AEU Int. J. Electron. Commun.* **2017**, *73*, 157–164. [[CrossRef](#)]
8. Rudenko, A.; Palmieri, L.; Herman, M.; Kitani, K.M.; Gavrila, D.M.; Arras, K.O. Human motion trajectory prediction: A survey. *Int. J. Robot. Res.* **2020**, *39*, 895–935. [[CrossRef](#)]
9. Luo, J.; Han, Y.; Fan, L. Underwater acoustic target tracking: A review. *Sensors* **2018**, *18*, 112. [[CrossRef](#)]
10. Song, H.; Hu, S. Effect of Faults on Kalman Filter of State Vectors in Linear Systems. *Technology* **2019**, *5*, 45–56. [[CrossRef](#)]
11. Patel, Z.; Boje, E. A Hybrid, Coupled Approach to the Continuous-Discrete Kalman Filter. *IEEE Control Syst. Lett.* **2020**, *5*, 827–832. [[CrossRef](#)]
12. You, S.; Diao, M.; Gao, L.; Zhang, F.; Wang, H. Target tracking strategy using deep deterministic policy gradient. *Appl. Soft Comput.* **2020**, *95*, 106490. [[CrossRef](#)]
13. Nguyen, N.H.; Doğançay, K. Instrumental variable based Kalman filter algorithm for three-dimensional AOA target tracking. *IEEE Signal Process. Lett.* **2018**, *25*, 1605–1609. [[CrossRef](#)]
14. Karsaz, A.; Adineh, A. Innovative Hybrid Backward Input Estimation and Data Fusion for High Maneuvering Target Tracking. *Int. J. Ind. Electron. Control Optim.* **2019**, *2*, 305–318.
15. Hadaegh, M.R.; Khaloozadeh, H.; Beheshti, M.T.H. Modification of Standard Kalman Filter Based on Augmented Input Estimation and Deadbeat Dissipative FIR Filtering. *IETE J. Res.* **2020**, 1–12. [[CrossRef](#)]
16. Ali, W.; Li, Y.; Raja, M.A.Z.; Ahmed, N. Generalized pseudo Bayesian algorithms for tracking of multiple model underwater maneuvering target. *Appl. Acoust.* **2020**, *166*, 107345. [[CrossRef](#)]
17. Ahn, C.K.; Shi, P.; Shmaliy, Y.; Liu, F. Bayesian state estimation for Markovian jump systems: Employing recursive steps and pseudocodes. *IEEE Syst. Man, Cybern. Mag.* **2019**, *5*, 27–36.
18. Samuel, K.; Choi, J. Improved IMM filter for tracking a highly maneuvering target with mixed system noises. *Int. J. Control Autom. Syst.* **2018**, *16*, 2763–2771. [[CrossRef](#)]
19. Cho, S.Y.; Ju, H.; Cha, J.; Park, C.G.; Yoo, K.; Park, C. IMM-based INS/EM-Log Integrated Underwater Navigation with Sea Current Estimation Function. *J. Position Navig. Timing* **2018**, *7*, 165–173.
20. Soken, H.E.; Sakai, S.-I. A new likelihood approach to autonomous multiple model estimation. *ISA Trans.* **2020**, *99*, 50–58. [[CrossRef](#)]
21. Liu, H.; Xia, L.; Wang, C. Maneuvering target tracking using simultaneous optimization and feedback learning algorithm based on Elman neural network. *Sensors* **2019**, *19*, 1596. [[CrossRef](#)] [[PubMed](#)]
22. Ali, W.; Li, Y.; Chen, Z.; Raja, M.A.Z.; Ahmed, N.; Chen, X. Application of spherical-radial cubature bayesian filtering and smoothing in bearings only passive target tracking. *Entropy* **2019**, *21*, 1088. [[CrossRef](#)]
23. Raptodimos, Y.; Lazakis, I. Application of NARX neural network for predicting marine engine performance parameters. *Ships Offshore Struct.* **2020**, *15*, 443–452. [[CrossRef](#)]
24. Peng, J.; Tang, Q. Application of NARX Dynamic Neural Network in Quantitative Investment Forecasting System. In Proceedings of the International Symposium on Intelligence Computation and Applications, Guangzhou, China, 16–17 November 2019; pp. 628–635.
25. Mehmood, A.; Zameer, A.; Ling, S.H.; ur Rehman, A.; Raja, M.A.Z. Integrated computational intelligent paradigm for nonlinear electric circuit models using neural networks, genetic algorithms and sequential quadratic programming. *Neural Comput. Appl.* **2020**, *32*, 10337–10357. [[CrossRef](#)]
26. Waseem, W.; Sulaiman, M.; Kumam, P.; Shoaib, M.; Raja, M.A.Z.; Islam, S. Investigation of singular ordinary differential equations by a neuroevolutionary approach. *PLoS ONE* **2020**, *15*, e0235829. [[CrossRef](#)] [[PubMed](#)]
27. Ahmad, I.; Raja, M.A.Z.; Bilal, M.; Ashraf, F. Bio-inspired computational heuristics to study Lane–Emden systems arising in astrophysics model. *SpringerPlus* **2016**, *5*, 1866. [[CrossRef](#)] [[PubMed](#)]
28. Raja, M.A.Z.; Shah, F.H.; Tariq, M.; Ahmad, I. Design of artificial neural network models optimized with sequential quadratic programming to study the dynamics of nonlinear Troesch’s problem arising in plasma physics. *Neural Comput. Appl.* **2018**, *29*, 83–109. [[CrossRef](#)]
29. Sabir, Z.; Manzar, M.A.; Raja, M.A.Z.; Sheraz, M.; Wazwaz, A.M. Neuro-heuristics for nonlinear singular Thomas–Fermi systems. *Appl. Soft Comput.* **2018**, *65*, 152–169. [[CrossRef](#)]
30. Ahmad, I.; Raja, M.A.Z.; Bilal, M.; Ashraf, F. Neural network methods to solve the Lane–Emden type equations arising in thermodynamic studies of the spherical gas cloud model. *Neural Comput. Appl.* **2017**, *28*, 929–944. [[CrossRef](#)]
31. Raja, M.A.Z.; Mehmood, A.; ur Rehman, A.; Khan, A.; Zameer, A. Bio-inspired computational heuristics for Sisko fluid flow and heat transfer models. *Appl. Soft Comput.* **2018**, *71*, 622–648. [[CrossRef](#)]
32. Ahmad, I.; Ahmad, F.; Raja, M.A.Z.; Ilyas, H.; Anwar, N.; Azad, Z. Intelligent computing to solve fifth-order boundary value problem arising in induction motor models. *Neural Comput. Appl.* **2018**, *29*, 449–466. [[CrossRef](#)]
33. Raja, M.A.Z.; Niazi, S.A.; Butt, S.A. An intelligent computing technique to analyze the vibrational dynamics of rotating electrical machine. *Neurocomputing* **2017**, *219*, 280–299. [[CrossRef](#)]
34. Khan, J.A.; Raja, M.A.Z.; Rashidi, M.M.; Syam, M.I.; Wazwaz, A.M. Nature-inspired computing approach for solving nonlinear singular Emden–Fowler problem arising in electromagnetic theory. *Connect. Sci.* **2015**, *27*, 377–396. [[CrossRef](#)]

35. Bukhari, A.H.; Sulaiman, M.; Islam, S.; Shoaib, M.; Kumam, P.; Raja, M.A.Z. Neuro-fuzzy modeling and prediction of summer precipitation with application to different meteorological stations. *Alex. Eng. J.* **2020**, *59*, 101–116. [[CrossRef](#)]
36. Ahmad, I.; Ahmad, S.; Awais, M.; Ahmad, S.U.I.; Raja, M.A.Z. Neuro-evolutionary computing paradigm for Painlevé equation-II in nonlinear optics. *Eur. Phys. J. Plus* **2018**, *133*, 184. [[CrossRef](#)]
37. Sabir, Z.; Wahab, H.A.; Umar, M.; Sakar, M.G.; Raja, M.A.Z. Novel design of Morlet wavelet neural network for solving second order Lane–Emden equation. *Math. Comput. Simul.* **2020**, *172*, 1–14. [[CrossRef](#)]
38. Raja, M.A.Z.; Umar, M.; Sabir, Z.; Khan, J.A.; Baleanu, D. A new stochastic computing paradigm for the dynamics of nonlinear singular heat conduction model of the human head. *Eur. Phys. J. Plus* **2018**, *133*, 364. [[CrossRef](#)]
39. Mehmood, A.; Zameer, A.; Ling, S.H.; Raja, M.A.Z. Design of neuro-computing paradigms for nonlinear nanofluidic systems of MHD Jeffery–Hamel flow. *J. Taiwan Inst. Chem. Eng.* **2018**, *91*, 57–85. [[CrossRef](#)]
40. Carrera, D.A.; Mayorga, R.V.; Peng, W. A Soft Computing Approach for group decision-making: A supply chain management application. *Appl. Soft Comput.* **2020**, *91*, 106201. [[CrossRef](#)]
41. Solanki, V.; Joshi, M. Energy Efficient NARX Model for Target Tracking in Wireless Sensor Network. In Proceedings of the 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 18–19 May 2018; pp. 260–265.
42. Liu, J.; Li, T.; Zhang, Z.; Chen, J. NARX prediction-based parameters online tuning method of intelligent PID system. *IEEE Access* **2020**, *8*, 130922–130936. [[CrossRef](#)]
43. Rangel, E.; Cadenas, E.; Campos-Amezcuca, R.; Tena, J.L. Enhanced Prediction of Solar Radiation Using NARX Models with Corrected Input Vectors. *Energies* **2020**, *13*, 2576. [[CrossRef](#)]
44. Hatata, A.Y.; Eladawy, M. Prediction of the true harmonic current contribution of nonlinear loads using NARX neural network. *Alex. Eng. J.* **2018**, *57*, 1509–1518. [[CrossRef](#)]
45. Chen, D.; Wang, T.; Yuan, P.; Sun, N.; Tang, H. A positional error compensation method for industrial robots combining error similarity and radial basis function neural network. *Meas. Sci. Technol.* **2019**, *30*, 125010. [[CrossRef](#)]
46. Jahan, K.; Rao, S.K. Implementation of underwater target tracking techniques for Gaussian and non-Gaussian environments. *Comput. Electr. Eng.* **2020**, *87*, 106783. [[CrossRef](#)]