Journal of *Marine Science and Engineering*

MDPI

Article

Scaled Conjugate Gradient Neural Intelligence for Motion Parameters Prediction of Markov Chain Underwater Maneuvering Target

Wasiq Ali [1,2,3,*], Habib Hussain Zuberi [1,2,3], Xin Qing [1,2,3,*], Abdulaziz Miyajan [4], Amar Jaffar [4] and Ayman Alharbi [4]

1 National Key Labortary of Underwater Acoustic Technology, Harbin Engineering University, Harbin 150001, China; habib_zuberi@hrbeu.edu.cn
2 Key Laboratory of Marine Information Acquisition and Security, Harbin Engineering University, Ministry of Industry and Information Technology, Harbin 150001, China
3 College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, China
4 Computer and Network Engineering Department, College of Computing, Umm Al-Qura University, Mecca 24231, Saudi Arabia; ammiyajan@uqu.edu.sa (A.M.); ayjaafar@uqu.edu.sa (A.J.); aarharbi@uqu.edu.sa (A.A.)
* Correspondence: wasiqali@hrbeu.edu.cn (W.A.); qingxin@hrbeu.edu.cn (X.Q.)

Abstract: This study proposes a novel application of neural computing based on deep learning for the real-time prediction of motion parameters for underwater maneuvering object. The intelligent strategy utilizes the capabilities of Scaled Conjugate Gradient Neural Intelligence (SCGNI) to estimate the dynamics of underwater target that adhere to discrete-time Markov chain. Following a state-space methodology in which target dynamics are combined with noisy passive bearings, nonlinear probabilistic computational algorithms are frequently used for motion parameters prediction applications in underwater acoustics. The precision and robustness of SCGNI are examined here for effective motion parameter prediction of a highly dynamic Markov chain underwater passive vehicle. For investigating the effectiveness of the soft computing strategy, a steady supervised maneuvering route of undersea passive object is designed. In the framework of bearings-only tracking technology, system modeling for parameters prediction is built, and the effectiveness of the SCGNI is examined in ideal and cluttered marine atmospheres simultaneously. The real-time location, velocity, and turn rate of dynamic target are analyzed for five distinct scenarios by varying the standard deviation of white Gaussian observed noise in the context of mean square error (MSE) between real and estimated values. For the given motion parameters prediction problem, sufficient Monte Carlo simulation results support SCGNI's superiority over typical generalized pseudo-Bayesian filtering strategies such as Interacting Multiple Model Extended Kalman Filter (IMMEKF) and Interacting Multiple Model Unscented Kalman Filter (IMMUKF).

Keywords: deep learning; motion parameters; state-space; underwater acoustics; passive bearings; Monte Carlo; pseudo-Bayesian

check for updates

Citation: Ali, W.; Zuberi, H.H.; Qing, X.; Miyajan, A.; Jaffar, A.; Alharbi, A. Scaled Conjugate Gradient Neural Intelligence for Motion Parameters Prediction of Markov Chain Underwater Maneuvering Target. *J. Mar. Sci. Eng.* **2024**, *12*, 240. https://doi.org/10.3390/jmse12020240

Academic Editor: Dong-Sheng Jeng

Received: 1 December 2023
Revised: 21 December 2023
Accepted: 25 January 2024
Published: 29 January 2024

Copyright: © 2024 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

The research community has recently shown a lot of interest in real-time motion parameters prediction of dynamic objects [1]. Many useful applications, both in civilian and defense like radar, sonar, unmanned aerial vehicle (UAV), enclosed tracking, intelligent transit system, air traffic control, surveillance, wireless radio network, drone navigation and bioinformatics have facilitated from the phenomenon of motion prediction [2,3]. Despite its widespread applications in many technical issues, robust motion parameters estimation still has potential for development and faces a number of difficulties to be solved [4]. The primary reasons that complicate motion parameters approximation of moving target are

uncertainty in modeling, complicated maneuverings in state equation of the target and wrongly observed bearings [5]. State and measurement models use numerical expression to construct the present and observed dynamics of the turning target. The analysis of motion parameters including position, velocity, acceleration, turnover rate, direction and how these change from their existing points at run time are described by these mathematical expressions of dynamic object [6]. A single dynamic model in estimating motion parameters cannot accurately describe all of the unpredictable maneuvering behaviors of an underwater passive vehicle. The performance of motion estimation techniques within a single model may be significantly affected if the state equation and the actual movements of the target do not match [7]. The most comprehensive and effective motion prediction models are those which calculate the statistical equations of object kinematics at each moment of the turning path [8].

Many significant studies have been investigated in the literature to achieve effective motion prediction of underwater dynamic target [9]. The initial attempt in this field involved the use of a single model Kalman filtering algorithm. However, the accuracy of its predictions is often significantly impacted by the irregular and complicated maneuverings of the undersea vehicle [10]. Following that, researchers suggested a typical comprehensive statistical design for increasing the efficiency of motion prediction phenomena in which the first-order Markov chain is perceived via second-order acceleration parameters [11]. Nevertheless, the model-based filtering method has led to improved accuracy in motion estimation for slight maneuverings, but it diverges in accuracy when dealing with steady speed and intricate underwater maneuvers. In addition, the maneuvering phenomena requires a greater amount of previous statistical data about the object than is typically available. Deterministic techniques have also drawn a great deal of interest from the scientific society in recent time because they appear to be superior for identifying and handling high maneuvers of underwater vehicles [12]. The input estimation algorithm and the variable dimension filter are two prominent instances of this paradigm [13–15].

The dynamics of the underwater turning object can be efficiently constructed using generalized pseudo-Bayesian (GPB) approaches. Design transition probabilities are practically challenging in GPB strategies, and it is also challenging to divide the target's dynamics into distinct multi motion models. Interacting multimodel (IMM) filters, which model the target's movements, use a network of sub-filters [16,17]. However, IMM motion prediction approaches already know about transition probability statistics and presumed dynamic models [18]. Model shifting probabilities are difficult to obtain in GPB algorithms, and it is also challenging to split the target's dynamics into multiple motion models [19]. Additionally, as the quantity of motion models escalates, the computational cost climbs dramatically, which negatively impacts the real-time effectiveness of the whole motion prediction methodology. However, IMM algorithms have certain drawbacks including unnecessary mathematical computations when dealing with maneuvering objects and potential accuracy divergence due to over motion modeling during maneuvering instants [20]. Simultaneously, by employing state-space modeling, the problem of motion parameters prediction can be effectively designed with time series approach [21]. All the approaches discussed in this context consider the state model of the dynamic underwater object using the phenomena of a white Gaussian process noise that is independent and identical (IDD) [22].

During the past two decades, nearly all engineering fields have seen the emergence of smart computing techniques [23,24]. The power of optimal techniques and strength of neural learning are utilized in artificial intelligence computing approaches to address a variety of challenging real-world scenarios [25,26]. Numerous divisions of applied sciences, including space science [27], plasma science [28], quantum mechanics [29], thermal physics [30], hydrodynamic [31], electrical motors and generators [32,33], electromagnetism [34], weather science [35], optical science [36], algebraic expressions [37], neuroinformatics [38], and microengineering [39], are finding extensive use for neural intelligence computing frameworks. These soft computing approaches outperformed all conventional methodologies and performed remarkably well in the aforementioned applications [40].

While compared to other conventional methods, neural networks have demonstrated significantly higher productivity and convergence rates in all relevant investigations.

Handling ambiguity and nonlinearity is a significant challenge in many realistic systems with time-varying dynamics and dependent on the present state of the system. Scaled conjugate gradient neural intelligence (SCGNI) methodology can be used for the aforementioned systems and exhibits an excellent convergence rate [41]. Dynamic inputs based on time-series data are successfully implemented by this artificial soft computational model in literature [42,43]. The use of artificial neural networks to predict time-series data is an uncertain method that does not require prior information of the given time series. The task of forecasting the maneuvering situation of underwater object is generally regarded as a nonlinear challenge due to the dynamic nature of the target's motion characteristics, which tend to fluctuate over time.

In our proposed work, SCGNI-based deep learning methodology is statistically built for proficiently analyzing the instantaneous motion parameters prediction of a navigating marine vehicle in multiple undersea circumstances. In this study, a comprehensive examination of the time series framework is conducted to approximate motion parameters from complicated passive mechanisms governing the motion of a two-dimensional moving object. Here, a novel neural processing platform is proposed and built for the purpose of achieving accurate motion parameters evaluation by considering various oceanic scenarios for highly maneuvering target. In our previously published work [17], a similar motion estimation paradigm is examined with Bayesian strategies like interacting multiple model extended Kalman filter (IMMEKF) and interacting multiple model unscented Kalman filter (IMMUKF). Here comparative results demonstrate the precision, robustness, and strength of SCGNI based neural intelligence computing. To evaluate the superiority of the proposed approach with standard filtering algorithms, numerical variations of observed Gaussian noise are carried out to create ideal and noisy underwater scenarios. This work employs the potency of the SCGNI computing to predict motion parameters like location, velocity, and rotation rate. Figure 1 depicts a detailed and concise graphical description of the proposed experiment. The following are the highlights of the conducted study:

(1) In order to accurately approximate the motion parameters of underwater Markov chain object, the ability and effectiveness of the SCGNI paradigm are thoroughly investigated.
(2) For a passive turning target, state estimations, position errors, velocity errors, rotation errors, deviation histograms, and regression analyses of SCGNI are calculated and compared to traditional Kalman filter variants that are nonlinear and use multiple models include IMMEKF and IMMUKF.
(3) The cutting-edge Wiener process velocity (WPV) and coordinated turn (CT) configurations are used to create the motion characteristics of the dynamical target in the high transition trajectory.
(4) As an evaluation criterion, measurement noise standard deviation is used, and its magnitudes are changed inside simulations in order to investigate the pattern of existing and proposed methodologies.
(5) Test metric minimal mean square error (MSE) is picked to compare all of the presented techniques.
(6) The simulation results show that effectiveness of SCGNI soft computing is significantly superior than multi model filters for predicting the motion parameters of an underwater moving target.

The remainder of the paper is structured as follows. In two-dimensional Cartesian coordinates, Section 2 presents the construction of a multiple model maneuvering parameter prediction framework based on bearings only tracking (BOT) model. This section also develops thorough mathematical modeling of continually turning object. Section 3 explains the structure and operation of the SCGNI network along with the training, testing, and validation procedures. Section 4 describes the results of the simulation and a discussion of the minimum MSE of the provided methods. The significant contribution is revealed and additional research objectives are stated in the final section of the proposed work.

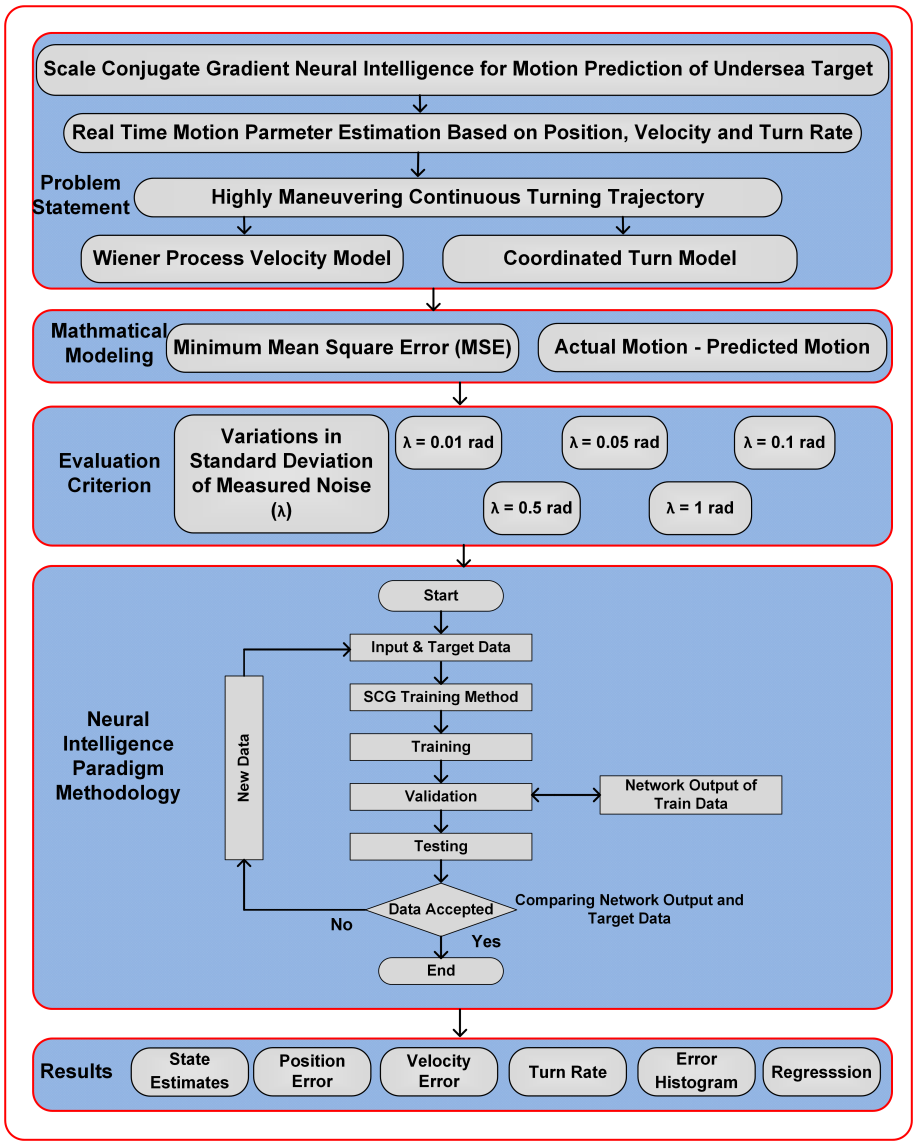


Figure 1. Detailed overview of the designed paradigm.

2. Markov Chain Maneuvering Motion Parameters Prediction Modeling

In this section of the study, a mathematically built bidirectional motion parameters prediction system of rectangular coordinates is used to model the Markov chain maneuvering object. This approach uses state space-based BOT technology to accurately predict the state of a continuously turning object in cluttered underwater medium. In order to gather passive bearings, eight equally spaced observer platforms are employed. The observers' positions are assumed to be known in prior. Only bearings of the moving object are obtained passively from observation platforms, and these bearings depend on the angle and orientation of each observer unit. Target movements are supposed to occur in the far field region of this proposed motion parameters prediction system model, following continuous turn trajectory which we intend to predict using nonlinear multimodel Kalman filters and neural computing. Passive bearings of the moving object is a term used to describe the nonlinear complicated data acquired at hydrophones, which is mostly depending on the location of each underwater observer. Figure 2 depicts the movement patterns of a navigating target and its motion parameters prediction mechanism.

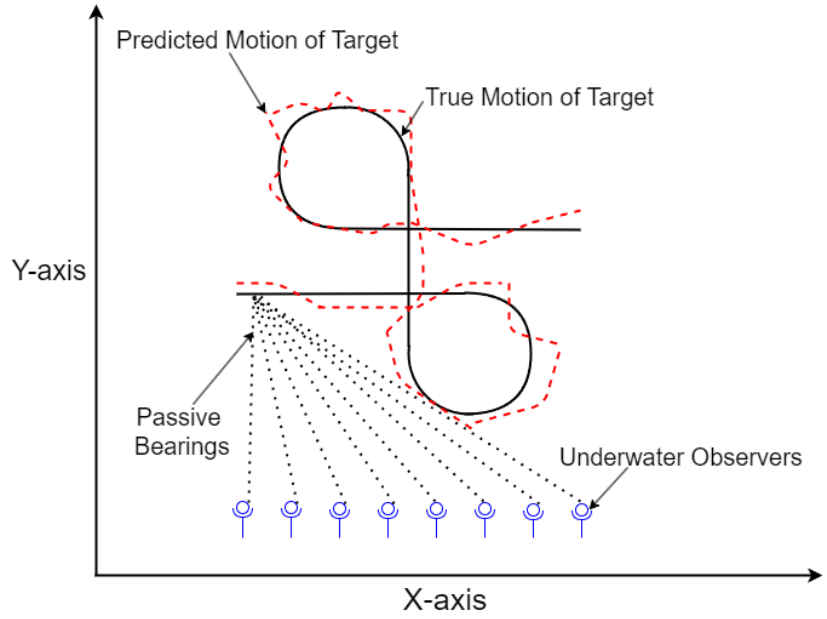


Figure 2. Motion parameters prediction model.

Many real-world systems have modeling parameters that change over time. These various system parameters cannot be defined by a single system model. Within the prediction process, it is possible for the modeling values to undergo changes in real-time motion parameters prediction applications. These systems fall under the categories of multimodel or Markov chain systems. The entire scheme may diverge in these cases if a single system model is chosen. As a result, it is essential to create a general motion model of dynamic objects that will govern various system models. In this research work, Wiener process velocity (WPV) along with coordinated turn (CT) models are used to define the mechanics of the underwater navigating vehicle.

2.1. Wiener Process Velocity (WPV) Framework

The state vector $Z_{\dagger}^{i}$ demonstrate the bidirectional rectangular coordinates real-time motion of the moving vehicle at time step † using position $(x_{\dagger}, y_{\dagger})$ and velocity $(\Delta x_{\dagger}, \Delta y_{\dagger})$ as follows:

$$Z_{\dagger}^{i} = \begin{bmatrix} x_{\dagger}^{i} & y_{\dagger}^{i} & \Delta x_{\dagger}^{i} & \Delta y_{\dagger}^{i} \end{bmatrix}^{\mathrm{T}}. \tag{1}$$

Meanwhile, the observer station's state vector in rectangular coordinates can be modeled as:

$$Z_{\dagger}^{j} = \begin{bmatrix} x_{\dagger}^{j} & y_{\dagger}^{j} & \Delta x_{\dagger}^{j} & \Delta y_{\dagger}^{j} \end{bmatrix}^{\mathrm{T}}. \tag{2}$$

The corresponding state expression of the dynamic target and the observing platform is as follows:

$$Z_{\dagger} = Z_{\dagger}^{i} - Z_{\dagger}^{j} = \begin{bmatrix} x_{\dagger} & y_{\dagger} & \Delta x_{\dagger} & \Delta y_{\dagger} \end{bmatrix}^{\mathrm{T}}. \tag{3}$$

Considering state-space technology, the kinetics of the maneuvering object are built using the discrete-time WPV model. This model can be used to describe the state equation as follows:

$$Z_{\dagger+1} = \mathbf{H}_{\dagger} Z_{\dagger} + \varpi_{\dagger}^{\mathrm{Y}}. \tag{4}$$

The motion switching matrix $\mathbf{H}_{\dagger}$ is expressed in the above state equation with entities of $m \times m$. The motion switching matrix in the state-space approach represents the

dynamic model's feedback. $\omega_{\dagger}^{Y}$ is representing Gaussian process noise with zero mean. The dispersion of motion switching matrix $\mathbf{H}_{\dagger}$ with sampling space $\Theta\dagger$ is expressed as:

$$H_{\dagger} = \begin{bmatrix} 1 & 0 & \Theta\dagger & 0 \\ 0 & 1 & 0 & \Theta\dagger \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

and this sampling space $\Theta\dagger$ is defined as:

$$\Theta\dagger = [(\dagger + 1) - \dagger]. \tag{6}$$

In order to accurately anticipate the motion parameters using the SCGNI scheme, the state space model described in Expression (4) should be represented in discrete time. The discrete-time nonlinear model is used because it has the capability to attain more accurate analysis of the system's performance at time intervals † which are multiples of the sampling space $\Theta\dagger$. The modified form of the discrete time state equation with its relevant parameters is as follows:

$$Z_{\dagger+1} = \underbrace{\begin{bmatrix} 1 & 0 & \Theta\dagger & 0 \\ 0 & 1 & 0 & \Theta\dagger \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{H_{\dagger}} \underbrace{\begin{bmatrix} x_{\dagger} \\ y_{\dagger} \\ \Delta x_{\dagger} \\ \Delta y_{\dagger} \end{bmatrix}}_{Z_{\dagger}} + \omega_{\dagger}^{Y}. \tag{7}$$

Gaussian process noise $\omega_{\dagger}^{Y}$ with covariance $\psi_{\dagger}$ is defined as:

$$\omega_{\dagger}^{Y} \approx N(0,\psi_{\dagger}). \tag{8}$$

In simulations, the variance of process noise is fixed to 0.05 for generating slightly slow bends of the target's trajectory.

2.2. Coordinated Turn (CT) Framework

The CT framework is a particular potential design that is typically employ for formulating the dynamic features of a passive object that is constantly following a rotational route. The state expression in this dynamic system model has an extra variable known as turning rate. This motion parameters vector of target in rectangular coordinates holds its orientation, velocity, and rotation rate. It can be represented using the CT model as:

$$Z_{\dagger}^{i} = \begin{bmatrix} x_{\dagger}^{i} & y_{\dagger}^{i} & \Delta x_{\dagger}^{i} & \Delta y_{\dagger}^{i} & \vartheta_{\dagger}^{i} \end{bmatrix}^{T}. \tag{9}$$

Additionally, the motion vector at the observer stage can be built as follows:

$$Z_{\dagger}^{j} = \begin{bmatrix} x_{\dagger}^{j} & y_{\dagger}^{j} & \Delta x_{\dagger}^{j} & \Delta y_{\dagger}^{j} & \vartheta_{\dagger}^{j} \end{bmatrix}^{T}. \tag{10}$$

A related state vector between the observer station and the moving object is defined as follows:

$$Z_{\dagger} = Z_{\dagger}^{i} - Z_{\dagger}^{j} = \begin{bmatrix} x_{\dagger} & y_{\dagger} & \Delta x_{\dagger} & \Delta y_{\dagger} & \vartheta_{\dagger} \end{bmatrix}^{T}. \tag{11}$$

The CT model's state equation is formulated as:

$$Z_{\dagger+1} = H_{\dagger}Z_{\dagger} + \hbar\omega_{\dagger}^{Y}, \tag{12}$$

with one additional parameter $\hbar$ that describes the distribution of Gaussian model noise $\omega_{\dagger}^{Y}$. The foregoing state expression in the CT model is nearly similar to the WPV framework

presented in (4). Like WPV model, the discrete-time pattern of state equation for this model is computed as:

$$Z_{\dagger+1} = \underbrace{\begin{bmatrix} 1 & 0 & \frac{\sin(\vartheta_\dagger \Theta \dagger)}{\vartheta_\dagger} & \frac{\cos(\vartheta_\dagger \Theta \dagger)-1}{\vartheta_\dagger} & 0 \\ 0 & 1 & \frac{1-\cos(\vartheta_\dagger \Theta \dagger)}{\vartheta_\dagger} & \frac{\sin(\vartheta_\dagger \Theta \dagger)}{\vartheta_\dagger} & 0 \\ 0 & 0 & \cos(\vartheta_\dagger \Theta \dagger) & -\sin(\vartheta_\dagger \Theta \dagger) & 0 \\ 0 & 0 & \sin(\vartheta_\dagger \Theta \dagger) & \cos(\vartheta_\dagger \Theta \dagger) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{H_\dagger} Z_\dagger + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\hbar} \omega_\dagger^{Y}. \tag{13}$$

The CT model assumes Gaussian process noise with zero mean and its covariance is defined as:

$$\omega_\dagger^{Y} \approx N(0, \psi_\dagger). \tag{14}$$

The CT model's numerical calculations are presented in Equation (13), despite the nonlinearity of its behavior. As a result, the CT model can be created using the following five algebraic equations:

$$x_{\dagger+1} = x_\dagger + \frac{\sin(\vartheta_\dagger \Theta \dagger)}{\vartheta_\dagger} \Delta x_\dagger + \frac{\cos(\vartheta_\dagger \Theta \dagger) - \Theta \dagger}{\vartheta_\dagger} \Delta y_\dagger, \tag{15}$$

$$y_{\dagger+1} = y_\dagger + \frac{1 - \cos(\vartheta_\dagger \Theta \dagger)}{\vartheta_\dagger} \Delta x_\dagger + \frac{\sin(\vartheta_\dagger \Theta \dagger)}{\vartheta_\dagger} \Delta y_\dagger, \tag{16}$$

$$\Delta x_{\dagger+1} = \cos(\vartheta_\dagger \Theta \dagger) \Delta x_\dagger - \sin(\vartheta_\dagger \Theta \dagger) \Delta y_\dagger, \tag{17}$$

$$\Delta y_{\dagger+1} = \sin(\vartheta_\dagger \Theta \dagger) \Delta x_\dagger + \cos(\vartheta_\dagger \Theta \dagger) \Delta y_\dagger, \tag{18}$$

$$\vartheta_{\dagger+1} = \vartheta_\dagger + \omega_\dagger^{Y}. \tag{19}$$

In simulations, for relatively high maneuverings of object, the Gaussian dynamic noise variance $\psi_\dagger$ for rotation rate $\vartheta_\dagger$ is fixed to numerical value of 0.15.

2.3. Measurement Framework

Both WPV and CT models in this motion parameters prediction approach share a measurement model that is also created using state-space phenomena. At time instant †, the mathematical formulation of measurement model is defined as:

$$M_{\dagger+1} = E(Z_{\dagger+1}, \Im_{\dagger+1}). \tag{20}$$

At time step †, the passive bearings from target are combined in a matrix $E(.)$, and it is also referred as a measurement matrix. This includes composite bearings, based on the point-slope tangent correlation strategy. The measured noise at time instant † is represented by a variable $\Im$ that follows an independent Gaussian dispersion in the above measurement expression. The following relationship is used to generate passive bearings, impinged at acoustic observers from the real motion of the dynamical target and placement of observers as:

$$E(Z_{\dagger+1}) = \underbrace{\arctan\left[\frac{y_\dagger - \Xi_y^n}{x_\dagger - \Xi_x^n}\right]}_{\text{bearings}}. \tag{21}$$

In the given measurement function, the real-time position of the moving target is denoted with $(x_\dagger, y_\dagger)$ in bidirectional rectangular coordinates, while localization of observer

n is denoted with (Ξ_x^n, Ξ_y^n). At time step †, for observer n, the measurement model M described in (20) can be expressed in an updated version as follows:

$$\mathrm{M}_{\dagger}^{n} = \arctan\left[\frac{y_{\dagger} - \Xi_y^n}{x_{\dagger} - \Xi_x^n}\right] + \Im_{\dagger}^{n}. \tag{22}$$

In the relation mentioned above, measurement noise $\Im_{\dagger}^{n}$ holds zero mean and its covariance $\mathrm{O}_{\dagger}$ is calculated as:

$$\Im_{\dagger}^{n} \cong \mathrm{N}(0, \mathrm{O}_{\dagger}), \tag{23}$$

$$\mathrm{O}_{\dagger} = \mathrm{diag}(\lambda_{\mathrm{M}}^2). \tag{24}$$

In Equation (24), the representation of measurement noise standard deviation is indicated by λ_{M}, that is indeed necessary for evaluating the precision of motion parameters prediction methods for underwater target tracking. The behavior of the underwater atmosphere is defined by fluctuating measured noise standard deviation. For the purpose of examining the robustness and potency of the neural learning and Bayesian methodologies, several values of the standard deviation of measurement noise are selected in our study.

The undersea object will attain a coordinated turning trajectory by following these steps.

(1) A moving object, with an initial position of (0,0) and a continuous velocity of 1 along the x-axis e.g., $(\Delta x_{\dagger}, \Delta y_{\dagger}) = (1, 0)$, begins its maneuvers at the origin.
(2) After 4 s, with a turn rate of $\vartheta_{\dagger} = -1$, the target makes a right turn.
(3) After 9 s, the target will no longer turn to the right and will instead continue travelling in a straight line at a constant velocity of 1.
(4) Marine object accelerates to left with rotation rate $\vartheta_{\dagger} = 1$ at 11 s.
(5) At 16 s, object stops turning to the left and sails straightaway with the same velocity for 4 s.

3. Neural Intelligence Modeling

For the motion parameters prediction of underwater moving object, we formed a SCGNI mathematical model in this section. The link between supplemental available data and defined time series in real-world applications cannot be overlooked. The estimation algorithm's effectiveness in motion prediction problems heavily rely on noisy bearings. Consequently, the understanding of noise deprived bearings or prior data can be utilized for modelling time series-based SCGNI to determine the optimal motion prediction performance.

3.1. Scaled Conjugate Gradient Neural Intelligence (SCGNI) Model

The time series phenomena effectively merges historical data to predict future values for the mathematical modelling of SCGNI. In the nonlinear framework of the SCGNI computing, the exterior input and subsequent output are used to predict the future values of time series data. For this reason, SCGNI employs a sophisticated multi-layer architecture that comprises a covert layer, an input layer, a delay layer, and an output layer. In this context we use a measurement series $\mathrm{M}(\dagger)$ with input delay of k to make predictions about the passive object's state $\mathrm{Z}(\dagger)$ for j prior values in the real data series Z. The neural network activity of SCGNI hidden nodes is represented here for time series forecasting as:

$$\mathrm{Z}(\dagger) = \beta\begin{pmatrix} \mathrm{M}(\dagger-1), \mathrm{M}(\dagger-2), \ldots \mathrm{M}(\dagger-k) \\ \mathrm{Z}(\dagger-1), \mathrm{Z}(\dagger-2), \ldots \mathrm{Z}(\dagger-j) \end{pmatrix} + e(\dagger). \tag{25}$$

In the aforementioned nonlinear model of SCGNI deep learning, the input vector $\mathrm{M}(\dagger)$ is α dimensional, whereas the output vector $\mathrm{Z}(\dagger)$ is β dimensional. The variable k represents input delay, j represents output delay while the variable $e(\dagger)$ denotes the error of the whole neural intelligence network.

$$\mathbf{A}(\dagger) = \gamma_{r_1}(\mathrm{M}(\dagger)\omega_{r_1} + \sigma_{r_1}), \tag{26}$$

$$Z(\dagger) = \gamma_{r_2}(\mathbf{A}(\dagger)\omega_{r_2} + \sigma_{r_2}). \tag{27}$$

In the given equation, the hidden layer vector is represented by the symbol $\mathbf{A}(\dagger)$, and it possesses a dimensionality of r. The weighted connection between the hidden layer and the delay layer is ω_{y_1}, whereas the weighted connection between output and delay layer is ω_{r_2}. The limits of the hidden and input layers are denoted as σ_{r_1} and σ_{r_2}, respectively. The γ_{r_1} is switching function of hidden nodes, whereas γ_{r_2} is used for activation of hidden nodes at output. The layered and overall structure of neural intelligence paradigm are shown in Figure 3 and Figure 4 respectively.

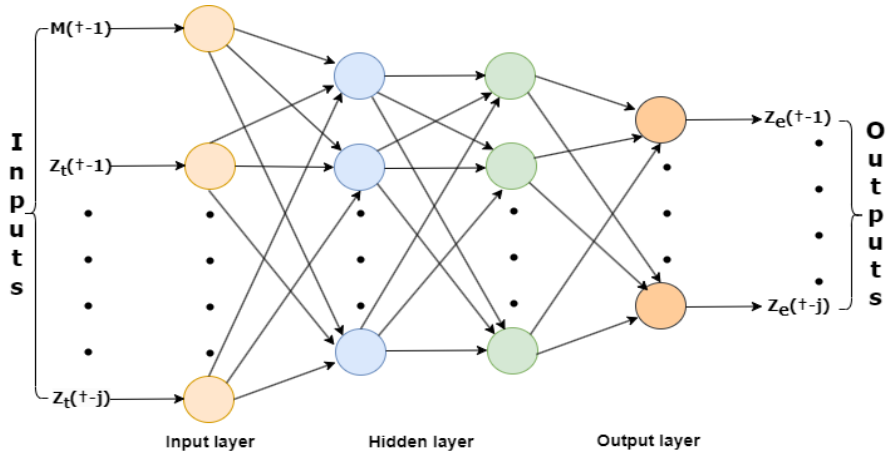


Figure 3. The layered nature of neural intellgence computing.

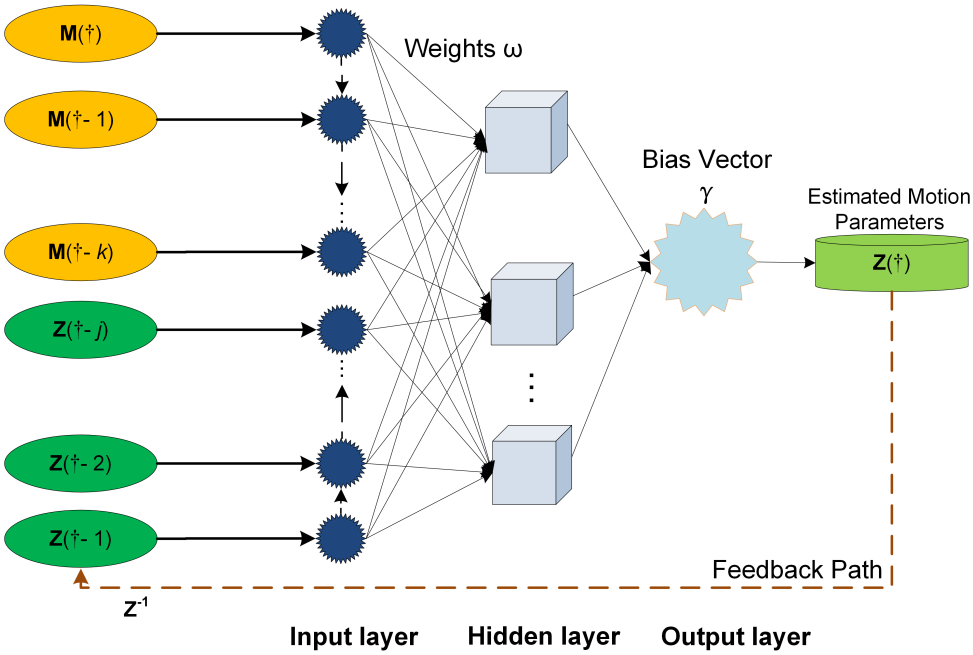


Figure 4. Overall structure of neural intelligence network.

3.2. SCGNI Computing Paradigm Architecture

It is imperative to acknowledge that SCGNI is regarded as a recurrent distinct time deep learning system in the domain of time series phenomena and formulated as follows:

$$Z(\dagger + 1) = \alpha\begin{pmatrix} Z(\dagger), Z(\dagger - 1), \ldots, Z((\dagger - a) + 1) \\ M(\dagger - (j+1)), \ldots, M((\dagger - a) - (j+1)) \end{pmatrix}. \tag{28}$$

If it is assumed that the idle time lag component j in the aforementioned model tends towards zero, the modified SCGNI model will be expressed as:

$$Z(\dagger+1) = \alpha\begin{pmatrix} Z(\dagger), Z(\dagger-1), \ldots Z(\dagger-a,+1) \\ M(\dagger), M(\dagger+1), \ldots, M(\dagger-a,+1) \end{pmatrix}. \tag{29}$$

The mathematical representation of the aforementioned model can be stated in vector form as follows:

$$\mathbf{Z}(\dagger+1) = \alpha(\mathbf{Z}(\dagger), \mathbf{M}(\dagger)). \tag{30}$$

In Equation (30), the vectors $\mathbf{M}(\dagger)$ and $\mathbf{Z}(\dagger)$ are denoting input and output parameters respectively. To ensure the optimal training of the neural framework, we implemented a training algorithm based on scaled conjugate gradient (SCG) methodology.

3.3. Scaled Conjugate Gradient (SCG) Based Training Algorithm

The training approach based on scaled conjugate gradient (SCG), was invented by Moller [44], which relies on the utilization of conjugate dimensions. The SCG-based training method has gained significant acceptance in the academic literature as an effective approach for building neural intelligence computing [45,46]. Compared to other conjugate gradient methods that necessitate a line check at each iteration, this particular technique does not incorporate a line search at every iteration. The SCG algorithm was developed with the intention of mitigating the time-consuming issue of line search phenomena. This training approach employed in the study is a second-order gradient-based supervised learning technique. The approach employed in this method involves the utilization of a trust-region step in contrast to a line-search step to adjust the magnitude of the step. The line search procedure includes additional factors in order to identify the appropriate step size, hence resulting in an increase in the training duration for any given learning technique. At each step of the trust-region method, the distance over which the model function can be trusted is changed. The model function is effective if it lies within the specified distance. Otherwise, an estimated minimal for the model function at the trust region boundary is employed. Trust-region algorithms provide more robustness compared to line-search methods. The SCG based training algorithm addresses the drawback of the line-search method by incorporating the trust-region method, similar to the approach utilized in the Levenberg–Marquardt method [47].

During each iteration, the training process based on the SCG algorithm calculates two first-order gradients for the variables in order to estimate the second-order information. This methodology has a network training function that updates weight and bias values regularly. The ability to train any network depends upon the existence of derivative functions for its weight, net input, and transfer functions. The step size in this context is determined by a quadratic estimation of the error function, resulting in enhanced robustness and less dependence on user-defined parameters. The estimation of the step size is conducted using a distinct approach. The calculation of the second order term is determined as:

$$\bar{d}_\dagger = \frac{e'(\bar{\omega}_\dagger + \phi_\dagger \bar{\Diamond}_\dagger) - e'(\bar{\omega}_\dagger)}{\phi_\dagger} + \eta_\dagger \bar{\Diamond}_\dagger. \tag{31}$$

In this context, the scalar value $\eta_\dagger$ is dynamically modified based on the variable $\ell_\dagger$. Weight vector is denoted with ω and $e'(\bar{\omega})$ is the global error parameter. The step size in training process is calculated as:

$$s_\dagger = \frac{v_\dagger}{\ell_\dagger} = \frac{-\bar{\Diamond}_\dagger^T e'_{g\omega}(r_1)}{\bar{\Diamond}_\dagger^T e''(\bar{\omega})\bar{\Diamond}_\dagger}. \tag{32}$$

The gradient of error is represented by $e''(\bar{\omega})$, quadratic estimation of error parameter is shown with $e'_{g\omega}(r_1)$, and non-zero weight vectors are $\bar{\Diamond}_1, \bar{\Diamond}_2 \ldots \bar{\Diamond}_\dagger$. Update $\eta_\dagger$ to ensure that:

$$\bar{\eta}_{\dagger} = 2(\eta_{\dagger} - \frac{\ell_{\dagger}}{|\bar{\Diamond}_{\dagger}|^2}). \tag{33}$$

If $\Theta_{\dagger} > 0.75$, then $\eta_{\dagger} = \frac{\eta_{\dagger}}{4}$. If $\Theta_{\dagger} < 0.25$, then $\eta_{\dagger} = \eta_{\dagger} + \frac{\ell_{\dagger}(1-\Theta_{\dagger})}{|\bar{\Diamond}_{\dagger}|^2}$. Likewise, $\Theta_{\dagger}$ is an evaluation function and computed as:

$$\Theta_{\dagger} = \frac{2\ell_{\dagger}[e(\bar{\omega}_{\dagger}) - e(\bar{\omega}_{\dagger} + s_{\dagger}\bar{\Diamond}_{\dagger})]}{v_{\dagger}^2}. \tag{34}$$

At the outset, the training values are chosen as, $0 < \phi \leq 10^{-4}$, $0 < \eta \leq 10^{-6}$ and $\bar{\eta} = 0$. A halt of training occurs by appearance of any of these conditions:

- The time limit has been surpassed.
- The upper limit of epochs has been attained.
- Performance is reduced to the target.
- The effect of the gradient appears lower than the minimal gradient.
- The performance of validation has exhibited a greater increase compared to the maximum number of failures since the previous occurrence of a fall in performance (when utilizing validation).

The real state of the maneuvering object $\mathbf{Z}(\dagger)$ as indicated by $\mathbf{y}(t)$ and the combination of passive bearings $\mathbf{M}(\dagger)$ at acoustics hydrophones are shown with $\mathbf{x}(t)$ in Figure 5. These are used as inputs to the SCGNI network in order to compute the approximated state $\mathbf{Z}(\dagger)$, which is also represented by the number $\mathbf{y}(t)$ at the neural network's output. The simulations are conducted using the SCGNI neural network toolkit in the MATLAB 2018a software environment. The SCGNI Model's toolbox structure comprises of three layers: input, hidden, and output, as depicted in the picture below.

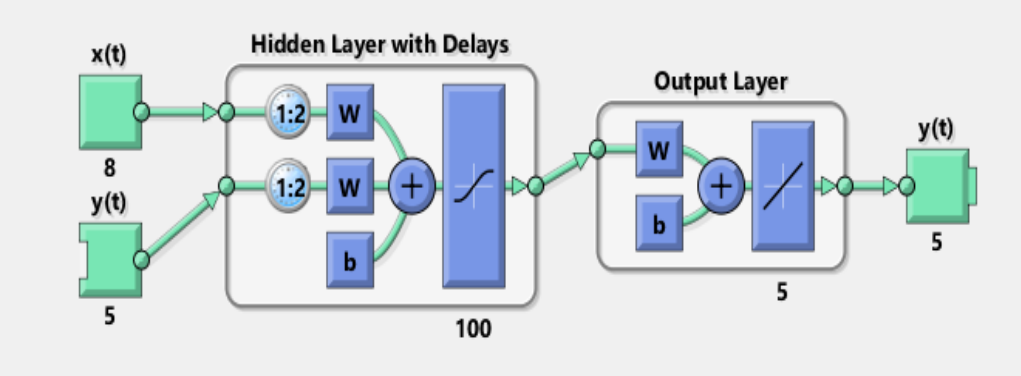


Figure 5. Neural network toolbox flow for SCGNI.

The true state vector in Equation (9) comprises five entities, including x-y positions, x-y velocities, and a turning parameter. This state vector is utilized as one input in the SCGNI model. Additionally, passive bearings from different acoustic hydrophones are utilized as another input. These inputs are employed to estimate the state vector, consisting on five components. In the context of simulations, a hidden layer consisting of 100 neurons is employed, with the activation of these neurons being facilitated by a sigmoid function. The training of weights is accomplished through SCG-based training approach, implementing the backpropagation through time (BPTT) strategy. Likewise, epoch based method is implemented during the training phase of the soft computing technique. During the neural learning of SCGNI paradigm, 70% of the entire data is allocated for the training phase, whereas the remainder 30% is evenly shared between the testing and validating steps for evaluating the produced outcomes.

3.4. Criteria for Performance Evaluation

The assessment criterion for deep neural intelligence methodology involves formulating a minimal mean square error (MSE) among the real and predicted motion parameters of a turning object at each time instant †. This paper presents an analysis of the reliability and robustness exhibited by the neural intelligence technique. As a consequence, the MSE formula for SCGNI, IMMEKF, and IMMUKF is derived separately for each individual Monte Carlo simulation as:

$$\mathrm{MSE}(\dagger) = \frac{1}{P}\sum_{\dagger=1}^{P}\left\|\mathbf{Z}_{\dagger}^{Actual} - \mathbf{Z}_{\dagger}^{Estimated}\right\|^{2}. \tag{35}$$

In the mentioned MSE function, actual motion parameters of the turning object are denoted as $\mathbf{Z}_{\dagger}^{Actual}$. On the other hand, approximated motion parameters of the object, obtained through the utilization of SCGNI and Bayesian filtering approaches, are expressed by $\mathbf{Z}_{\dagger}^{Estimated}$. The total number of data points are denoted by P, with a value of 200 in simulations, whereas † = 1 represents the starting data point. The computation of motion parameter errors for the turning trajectory occurs at each time step.

4. Findings and Analysis of Simulation Results

This section of the study provides a concise explanation of the simulation findings for the proposed smart computing based on SCGNI. The results include real-time motion parameters estimates, location error, velocity divergence, rotation estimates, error histogram, and regression study. Five distinct scenarios in simulations are performed and standard deviation of observed noise is varied for checking the performance of motion prediction algorithms. This parameter is systematically tuned within the range of 0.01 to 1 radian. The measured noise exhibits a complex marine environment when the value of λ is at its greatest, which is 1 radian. Conversely, the ideal atmosphere is represented by a minimal value of λ , which is 0.01 radian.

In the execution of simulations, the precise adjustment of mathematical variables throughout motion parameters prediction modeling is essential in order to achieve the desired performance outcomes. Table 1 provides appropriate values for the variables of motion estimation.

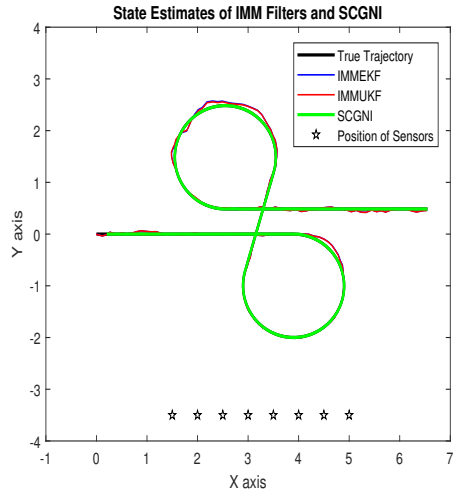
Table 1. Configuration of several factors for estimating of motion parameters.

Variables	Suitable Values
Initial motion parameters of the object	$\mathbf{Z}_0 = [0\ 0\ 1\ 0\ 0]^{\mathrm{T}}$
Acoustic hydrophones placement function	(Ξ_x^n, Ξ_y^n)
Number of acoustic hydrophones	n = 8
Hydrophones space	0.5
The standard deviation values of the observed noise	$\lambda = 0.01 \rightarrow 1$ radians
WPV framework process noise variance	$\psi_{\dagger}$ = 0.05
CT framework process noise variance	$\psi_{\dagger}$ = 0.15
Sample space	$\Theta\dagger$ = 0.01
Delays of input and output layer	k, j = 2
Quantity of sample values	200
Hidden nerve cells	100
SCGNI target points	1000

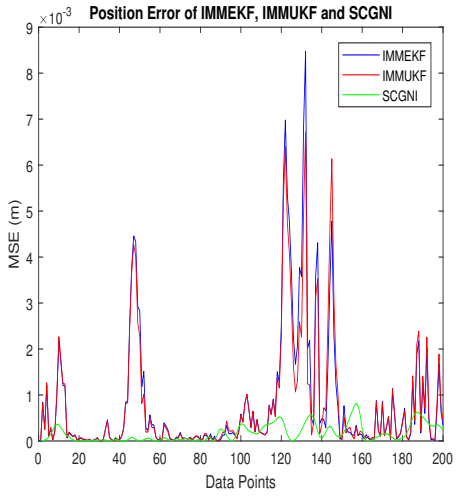
4.1. Analysis of Markov Chain Maneuvering Target Motion Characteristics under Various Conditions of Measurement Noise Standard Deviation

This subsection presents an explanation of the simulation results and a brief discussion of the real-time prediction of motion parameters for maneuvering object. The computing

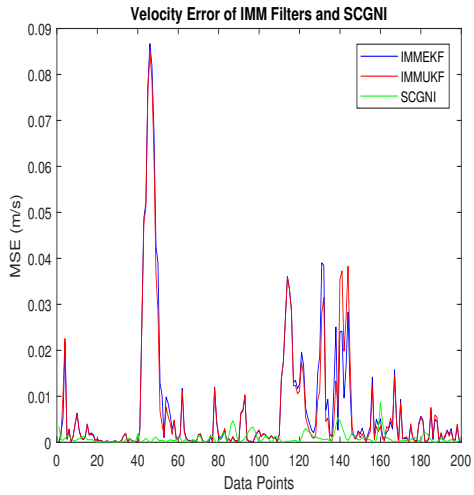
techniques employed in this study are IMMEKF, IMMUKF, and SCGNI. In this study, we conduct a comparative analysis of the accuracy and convergence performance of the SCGNI algorithm in relation to IMM filters. IMM filters with same system model are investigated in our previous recorded study [17]. The analysis specifically focuses on five different values of observed noise standard deviation. Figures 6–15 present various outcomes of the GPB algorithms and SCGNI computing, including parameters prediction, orientation error, velocity error, rotation predicts, error histogram, and regression study. The following study presents evaluation of five distinct scenarios, accompanied by their corresponding mathematical formulations and simulation results.



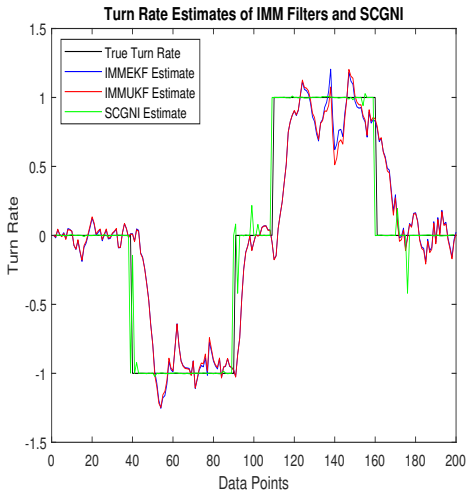
(**a**) Motion predictions of maneuvering object



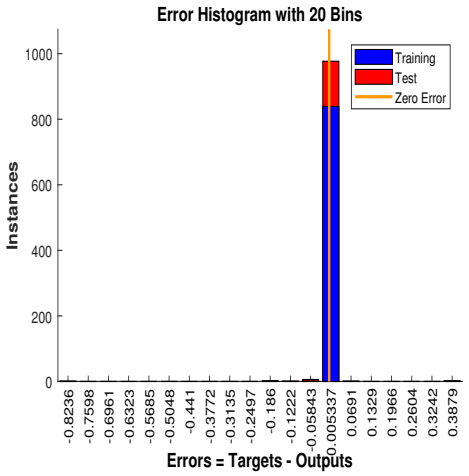
(**b**) Position error for maneuvering path



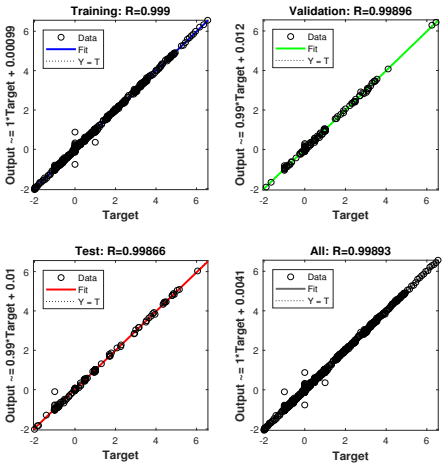
(**c**) Velocity error for maneuvering path



(**d**) Turn rate estimates for maneuvering path



(**e**) Error histogram of output and target data



(**f**) Regression result of SCGNI computing

Figure 6. Motion parameters results of IMMEKF, IMMUKF, and SCGNI for λ = 0.01 rad.

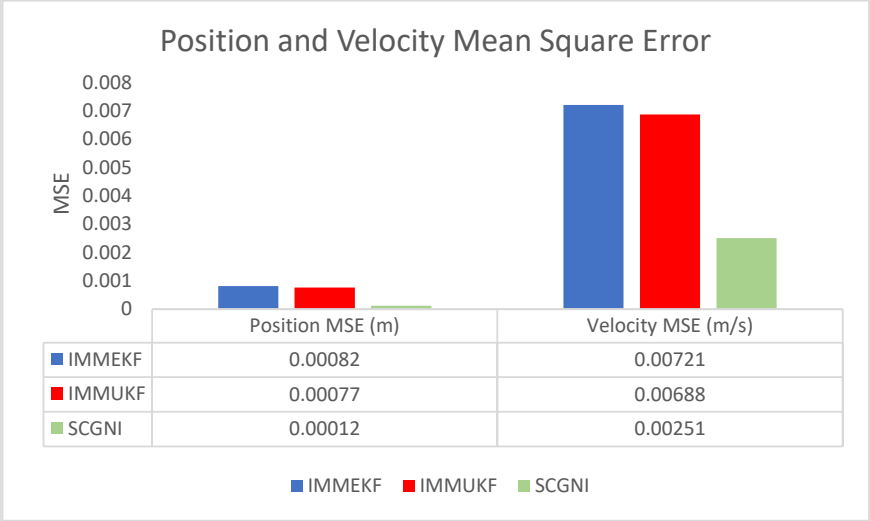
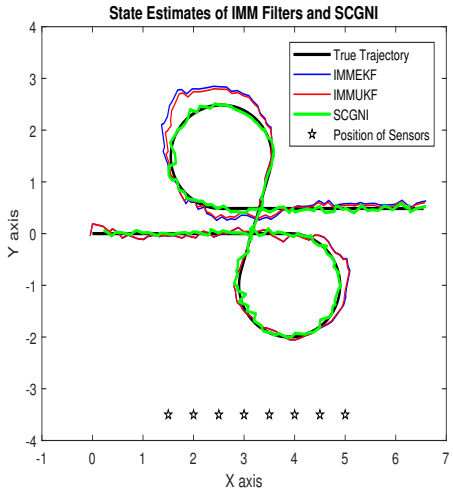
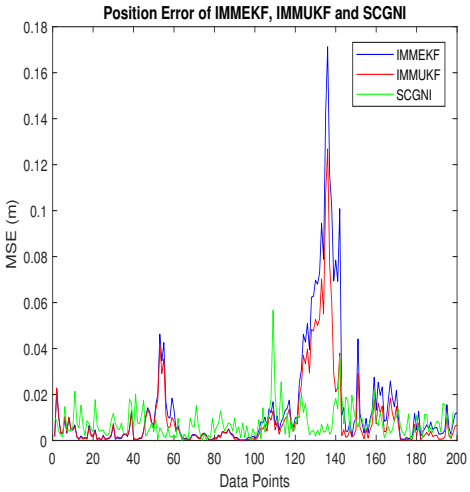


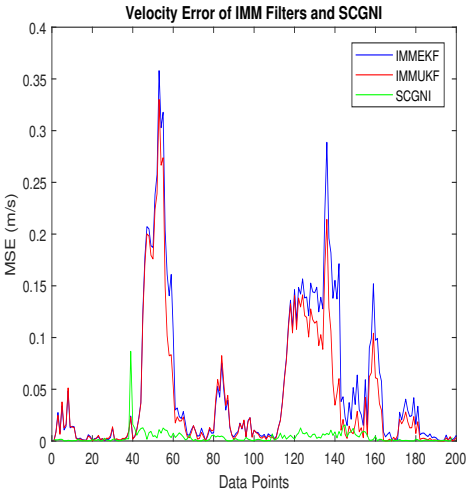
Figure 7. The IMMEKF, IMMUKF, and SCGNI's average MSEs for predicting position and velocity in scenario 1.



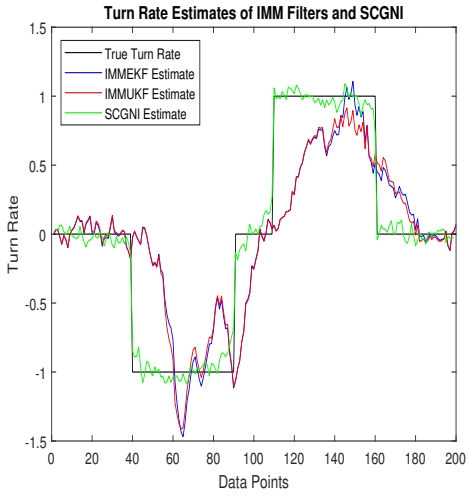
(**a**) Motion predictions of maneuvering object



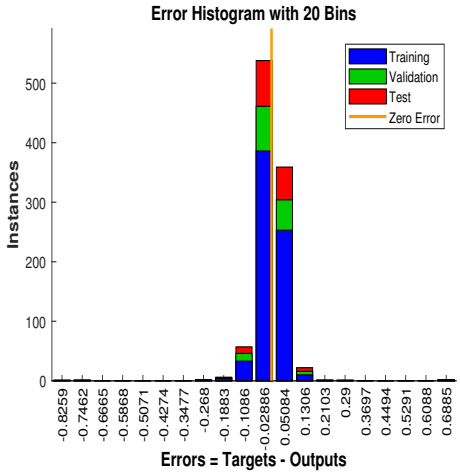
(**b**) Position error for maneuvering path



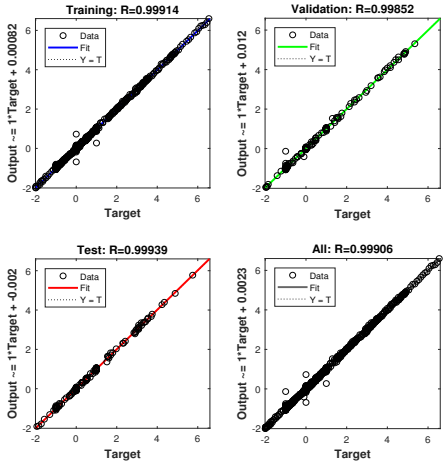
(**c**) Velocity error for maneuvering path



(**d**) Turn rate estimates for maneuvering path



(**e**) Error histogram of output and target data



(**f**) Regression result of SCGNI computing

Figure 8. Motion parameters results of IMMEKF, IMMUKF, and SCGNI for $\lambda = 0.05$ radians.

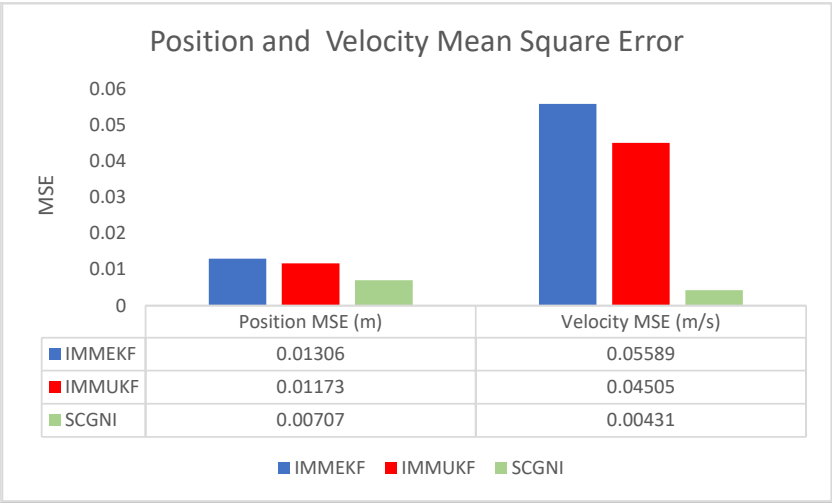
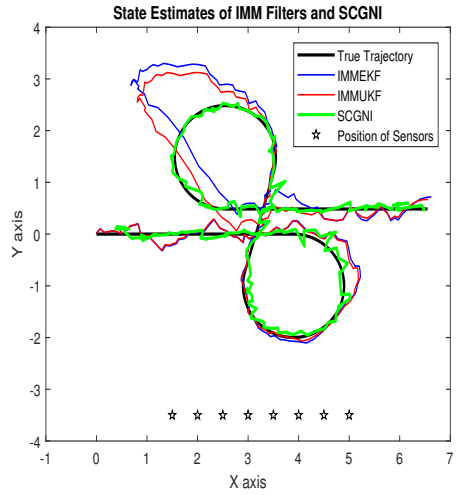
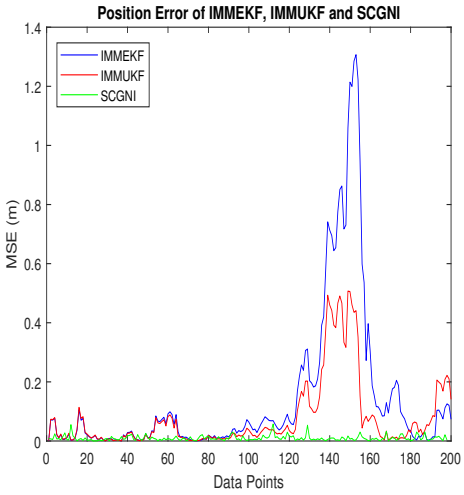


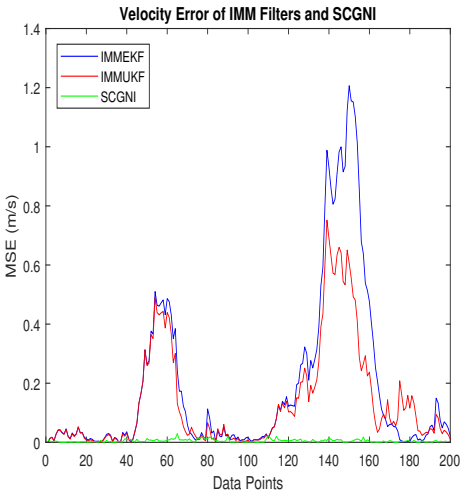
Figure 9. The IMMEKF, IMMUKF, and SCGNI's average MSEs for predicting position and velocity in scenario 2.



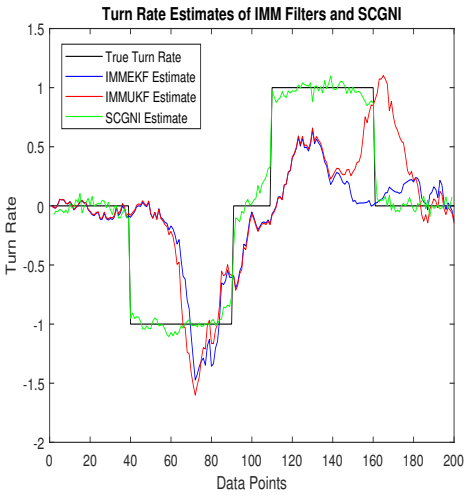
(**a**) Motion predictions of maneuvering object



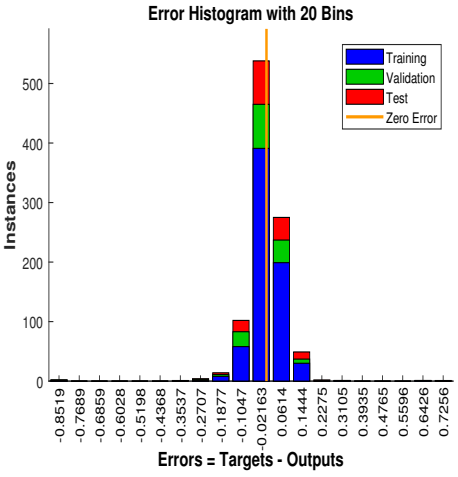
(**b**) Position error for maneuvering path



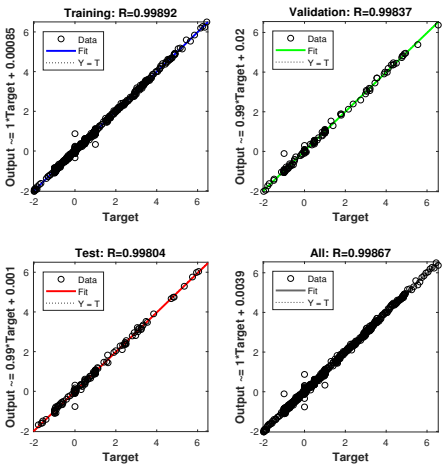
(**c**) Velocity error for maneuvering path



(**d**) Turn rate estimates for maneuvering path



(**e**) Error histogram of output and target data



(**f**) Regression result of SCGNI computing

Figure 10. Motion parameters results of IMMEKF, IMMUKF, and SCGNI for $\lambda = 0.1$ rad.

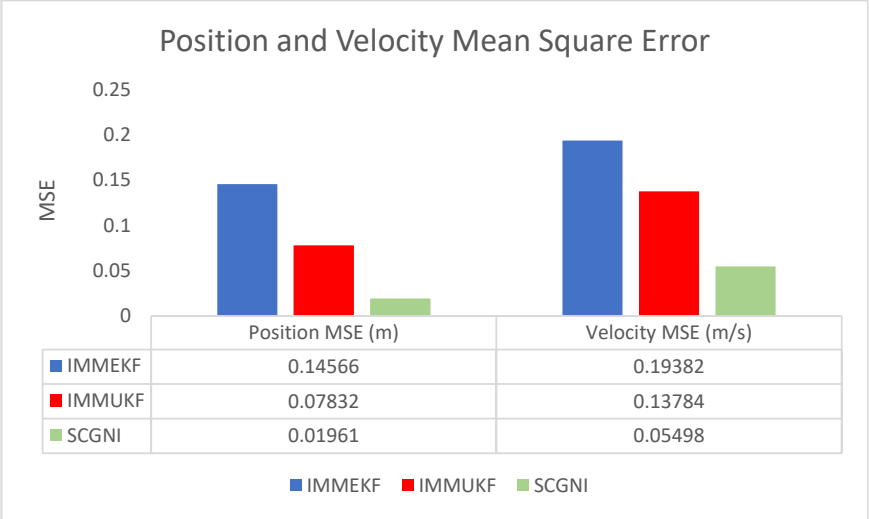
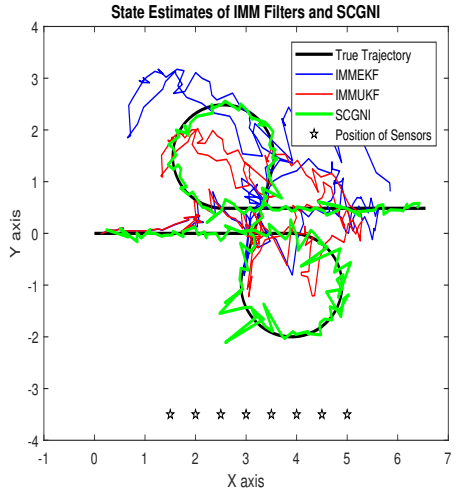
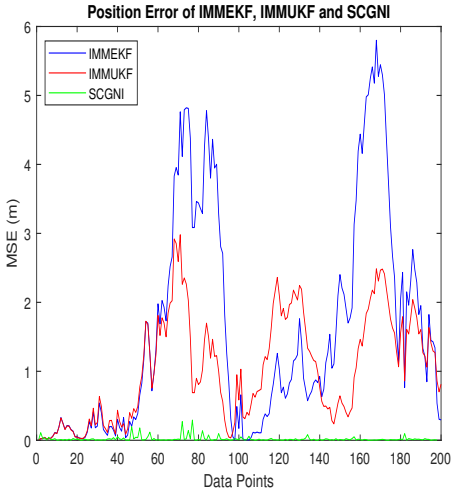


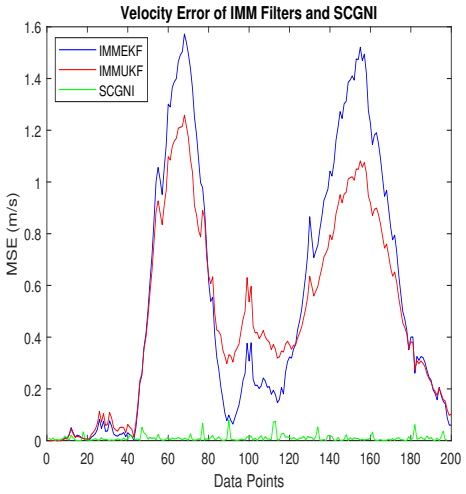
Figure 11. The IMMEKF, IMMUKF, and SCGNI's average MSEs for predicting position and velocity in scenario 3.



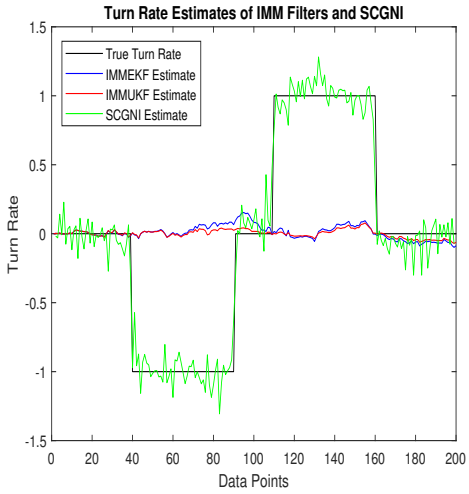
(**a**) Motion predictions of maneuvering object



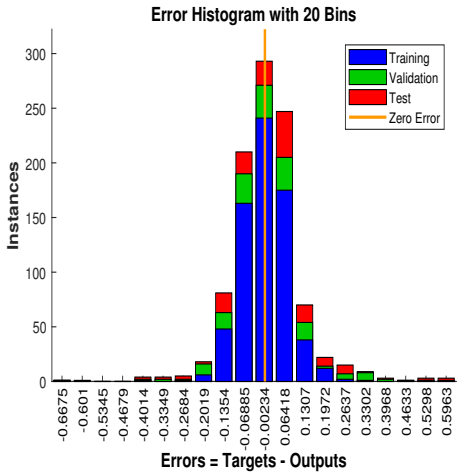
(**b**) Position error for maneuvering path

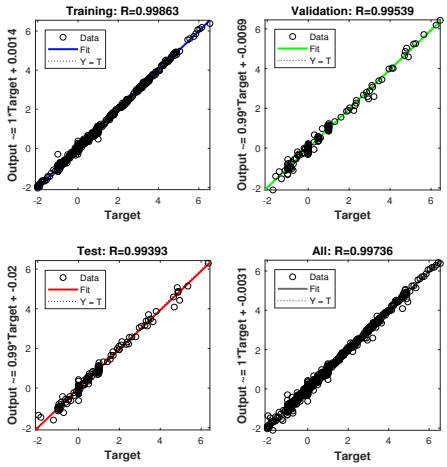


(**c**) Velocity error for maneuvering path



(**d**) Turn rate estimates for maneuvering path





(**e**) Error histogram of output and target data (**f**) Regression result of SCGNI computing

Figure 12. Motion parameters results of IMMEKF, IMMUKF, and SCGNI for $\lambda = 0.5$ rad.

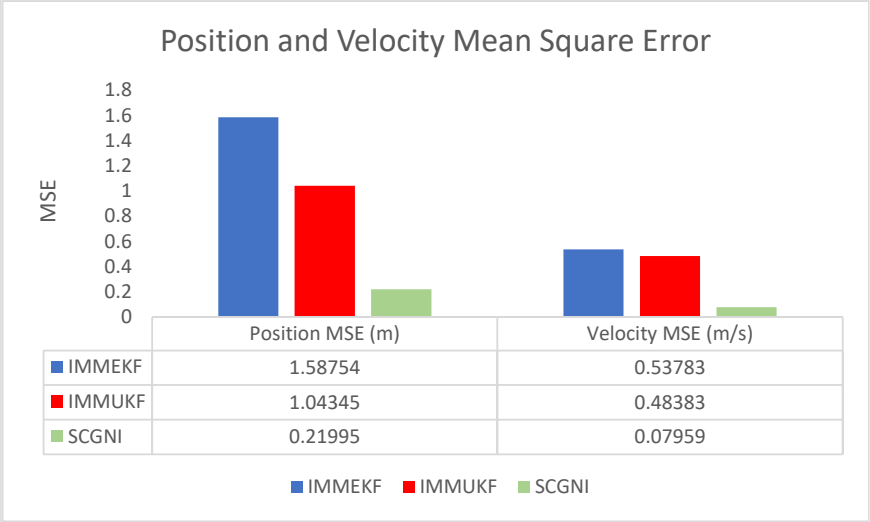
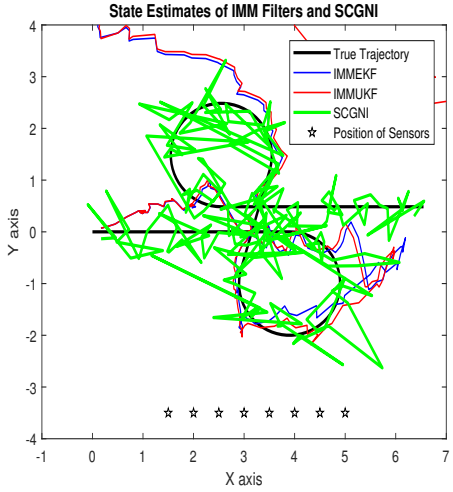
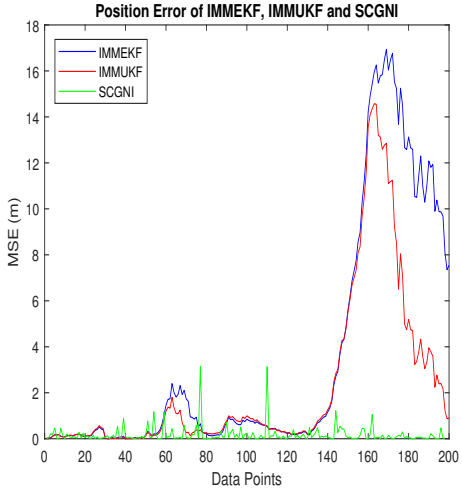
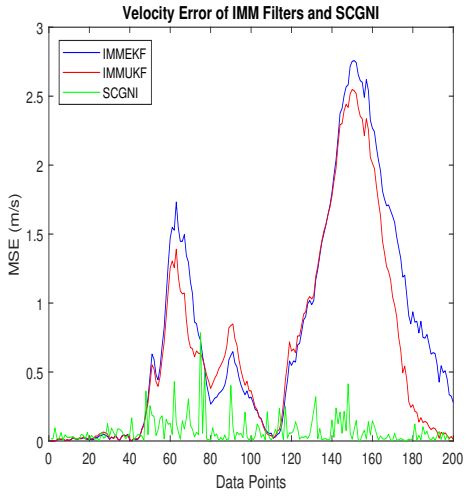


Figure 13. The IMMEKF, IMMUKF, and SCGNI's average MSEs for predicting position and velocity in scenario 4.

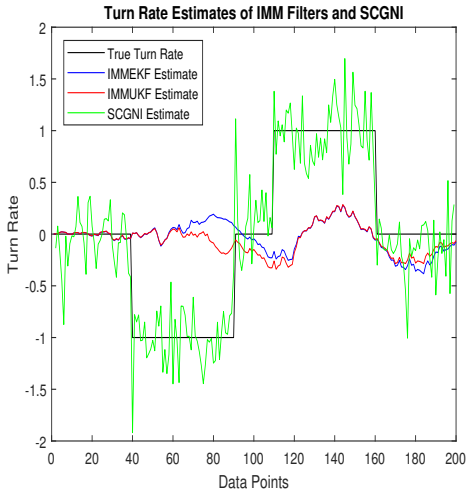




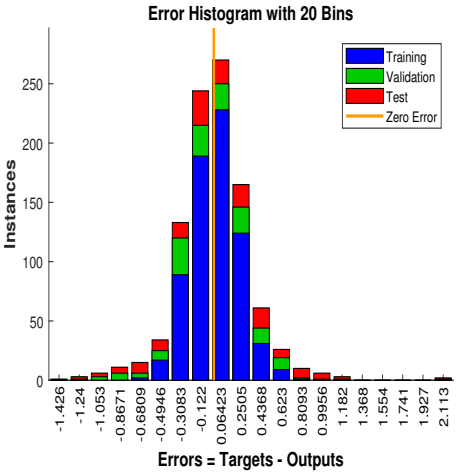
(**a**) Motion predictions of maneuvering object (**b**) Position error for maneuvering path



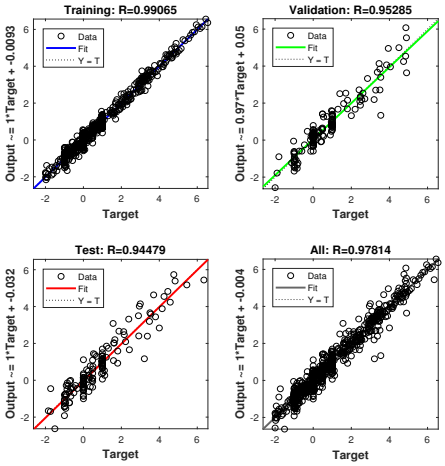
(**c**) Velocity error for maneuvering path



(**d**) Turn rate estimates for maneuvering path



(**e**) Error histogram of output and target data



(**f**) Regression result of SCGNI computing

Figure 14. Motion parameters results of IMMEKF, IMMUKF, and SCGNI for $\lambda = 1$ radian.

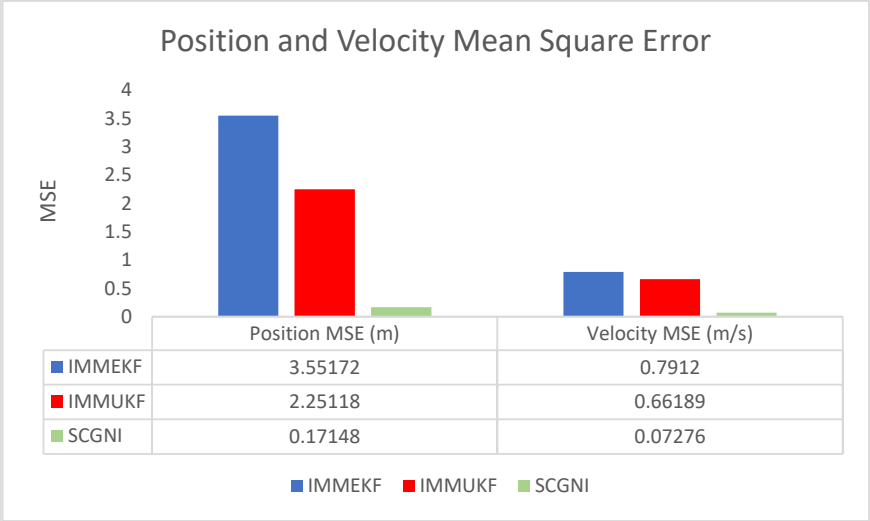


Figure 15. The IMMEKF, IMMUKF, and SCGNI's average MSEs for predicting position and velocity in scenario 5.

4.1.1. Scenario 1: The Measurement Noise Standard Deviation λ = 0.01 Rad

In the first scenario, a standard deviation of 0.01 radians is used to represent the measured Gaussian noise λ. This value is indicative of an undersea environment that is close to being perfect and characterized by smooth conditions. The covariance in this particular scenario involves utilizing the measured noise standard deviation as:

$$\mathrm{O}_{\dagger} = \mathrm{diag}(\lambda_{\mathrm{M}}^{2}). \tag{36}$$

The observed Gaussian noise at given time instant † for n sensor is determined from covariance as:

$$\Im_{\dagger}^{n} \approx \mathrm{N}(0, \mathrm{O}_{\dagger}). \tag{37}$$

The above determined measured noise is incorporating in measurement function M as follows for all passive bearings at time step † for sensor n:

$$\mathrm{M}_{\dagger}^{n} = \arctan\left[\frac{y_{\dagger} - \Xi_{y}^{n}}{x_{\dagger} - \Xi_{x}^{n}}\right] + \Im_{\dagger}^{n}. \tag{38}$$

The SCGNI deep learning model uses the above-designed measurement model as one of its inputs, as time series data. The target's time series is another input for the SCGNI model, in which the actual state vector described below is used to find motion parameters for underwater maneuvering target.

$$\mathrm{Z}_{\dagger} = \mathrm{Z}_{\dagger}^{i} - \mathrm{Z}_{\dagger}^{j} = \begin{bmatrix} x_{\dagger} & y_{\dagger} & \Delta x_{\dagger} & \Delta y_{\dagger} & \vartheta_{\dagger} \end{bmatrix}^{\mathrm{T}}. \tag{39}$$

Real-time state approximates, mean real and predicted location and velocity divergence, actual and predicted rotation rate, error histogram, and regression study of true and computed maneuvering track are shown for this observed noise standard deviation in the first scenario as follows.

In this situation, the assumption of a quiet oceanic atmosphere is made by selecting a minimal numerical threshold for the measured noise.

- Figure 6a compares the performance of SCGNI computing with Bayesian filtering algorithms like IMMEKF and IMMUKF for predicting motion parameters of a maneuvering object following turning trajectory. It is evident that SCGNI accurately traces the actual turning path of the maneuvering target.
- The assessment of position errors in a mean square sense is depicted in Figure 6b, where it is observed that SCGNI computing exhibits the lowest error compared to other approaches.
- Figure 6c displays a difference between the actual velocity and the estimated velocity, as measured by the MSE. The SCGNI deep learning mechanism demonstrates superior efficiency in computing velocity equated to nonlinear Kalman estimators for 200 samples in this outcome.
- Figure 6d depicts the rotation parameter prediction of the IMMEKF, IMMUKF, and SCGNI algorithms. The success rate of SCGNI over IMM estimators for the entire turning trajectory is being confirmed by prediction of turning rate $\vartheta_{\dagger}$ for all samples.
- Figure 6e shows the error histogram between the target time series $\mathbf{Z}(\dagger - 1)$, $\mathbf{Z}(\dagger - 2), \ldots, \mathbf{Z}(\dagger - \mathrm{j})$, , and the predicted output $\mathbf{Z}_{\dagger}^{Estimated}$. The error histogram is comprised of a collection of divergence points, which may include both negative and positive values. These error values quantify the distinction between target and estimated time series. The comprehensive error of the SCGNI network is partitioned into 20 bins in this investigation, visually represented by vertical strips. In the centre of histogram, a bar exhibits distinctiveness in comparison to other bins, as it manifests an error of 0.005337 at a height of 1000 occurrences. This observation suggests that a number of data points within the time series exhibit errors falling within this specific

range. The zero-error bar is likewise situated within this bin, which establishes the neural network's zero error.

- Figure 6f illustrates the regression evaluation of the SCGNI strategy for training, validation, and testing. The time series dataset utilized in the soft computing architecture is partitioned into three subsets: training, validation, and testing. These subsets are allocated proportions of 70%, 15%, and 15% correspondingly. This analysis incorporates statistical factors to demonstrate the correlation between the output vector $\mathbf{Z}_{\dagger}^{Estimated}$ and the required vector $\mathbf{Z}_{\dagger}^{Actual}$. In the regression analysis, the actual target and estimated output exhibit a high degree of overlap. A linear pattern is detected in both data that indicate the level of effectiveness of SCGNI soft computing.

In addition, we calculate the MSE for both location and velocity of the object by comparing the actual values with the anticipated values. These findings indicate that the precision of SCGNI surpasses that of IMM Kalman filters in terms of position and velocity error responses. This demonstrates the efficacy of utilizing neural network applications for predicting motion parameters in maneuvering target applications. The graph below displays the location and velocity errors for the IMMEKF, IMMUKF, and SCGNI methods.

4.1.2. Scenario 2: The Measurement Noise Standard Deviation λ = 0.05 Rad

In scenario 2, a value of λ = 0.05 radians is selected as the standard deviation of observation noise. This choice is made to introduce a certain level of measurement noise into the computation system. In the present context, the calculation of variance is derived from the measured noise standard deviation as:

$$\mathrm{O}_{\dagger} = \mathrm{diag}(\lambda_{\mathrm{M}}^{2}). \tag{40}$$

The measured noise at time interval † for sensor n follows a Gaussian distribution, with its characteristics determined by the previously estimated covariance as:

$$\Im_{\dagger}^{n} \approx \mathrm{N}(0, \mathrm{O}_{\dagger}). \tag{41}$$

The measurement framework for sensor n includes measurement noise at each instant † as:

$$\mathrm{M}_{\dagger}^{n} = \arctan\left[\frac{y_{\dagger} - \Xi_{y}^{n}}{x_{\dagger} - \Xi_{x}^{n}}\right] + \Im_{\dagger}^{n}. \tag{42}$$

Similarly, this established measurement model $\mathrm{M}_{\dagger}^{n}$ is used as an input time series by the neural paradigm. The target time series applied at another input of SCGNI is represented by the following motion parameters vector.

$$\mathrm{Z}_{\dagger} = \mathrm{Z}_{\dagger}^{i} - \mathrm{Z}_{\dagger}^{j} = \begin{bmatrix} x_{\dagger} & y_{\dagger} & \Delta x_{\dagger} & \Delta y_{\dagger} & \vartheta_{\dagger} \end{bmatrix}^{\mathrm{T}}. \tag{43}$$

The motion parameters that are significant for estimation using SCGNI computing are the real-time orientation of the maneuvering target $(x_{\dagger}, y_{\dagger})$, velocity $(\Delta x_{\dagger}, \Delta y_{\dagger})$, and rotation rate $\vartheta_{\dagger}$. The following part presents the simulation results, which include motion prediction, location error, velocity error, rotation approximates, divergence histogram, and regression study for the current scenario.

- Figure 8a presents a comparison of the predicted motion characteristics for all three approaches in relation to the integrated turn route of the navigating target. It is noteworthy to observe that, in the context of this specific standard deviation of observed noise, the motion estimations and convergence of SCGNI surpass those of typical nonlinear filtering methods.
- The figure labeled as Figure 8b illustrates the intermediate MSE between the true and forecasted positions of a submerged navigation object. This result also serves as a representation of the precision of the SCGNI method with respect to IMMEKF and IMMUKF methodologies.

- Figure 8c presents a comparison between the real and predicted velocity of a maneuvering vehicle, as calculated by IMM filters and the SCGNI network. The results indicate that the intelligence neural methodology outperforms the filtering techniques in terms of accuracy.
- Figure 8d illustrates the turn rate approximates for the current scenario. Once again, it is evident that the SCGNI method outperforms Bayesian filtering techniques in terms of accurately determining the turning parameter.
- Figure 8e displays the error histogram outcome, which compares the target time series dataset $\mathbf{Z}(\dagger-1), \mathbf{Z}(\dagger-2), \ldots, \mathbf{Z}(\dagger-\mathrm{j})$, with the expected output value $\mathbf{Z}_{\dagger}^{Estimated}$ of the target's motion properties. The histogram displays a vertical bin at its center, representing the error of -0.02886. This bin corresponds to the training dataset and has a height that is close to 500 occurrences. In contrast, the validation and testing datasets are represented by bins that range between 400 and 500 instances. In this particular instance, the zero error is situated below the vertical bar, which has a center at -0.02886.
- Figure 8f presents the regression analysis of SCGNI computation for the training, testing, and validation stages. In regression graph, effectiveness of the SCGNI scheme is evident that how the real target and estimated output respond head-to-head.

The MSEs are also calculated to determine the discrepancy between the real and projected location and velocity of the turning object in the presence of observed noise.

The location and velocity deviations further support the findings that the predictability of SCGNI surpasses that of IMM Kalman filters. This highlights the effectiveness of neural networks in predicting motion parameters. The location and velocity errors obtained from SCGNI and multi model Kalman filters are listed in the above chart.

4.1.3. Scenario 3: The Measurement Noise Standard Deviation λ = 0.1 Rad

In this circumstance, the measured noise standard deviation gets larger to λ = 0.1 radians, indicating the presence of a sufficient level of noise introduced into the entire system. The calculation of the covariance O at instantaneous time †, given a standard deviation of 0.1 radian for Gaussian measured noise, is as follows:

$$\mathrm{O}_{\dagger} = \mathrm{diag}(\lambda_{\mathrm{M}}^{2}). \tag{44}$$

The observed noise arising from the previously derived covariance for n sensors at time step † is formally presented as:

$$\Im_{\dagger}^{n} \approx \mathrm{N}(0, \mathrm{O}_{\dagger}). \tag{45}$$

The produced Gaussian measured noise is being included in the observation modelling as:

$$\mathrm{M}_{\dagger}^{n} = \arctan\left[\frac{y_{\dagger} - \Xi_{y}^{n}}{x_{\dagger} - \Xi_{x}^{n}}\right] + \Im_{\dagger}^{n}. \tag{46}$$

The measurement framework M at time step † for hydrophone n is shown in the above equation. This framework incorporates the passive angles impinged at acoustic hydrophones, and linked with observed noise that follows white Gaussian distribution. The measurement model formulation is utilized as the input dataset for the deep learning model. The additional input of the neural network comprises on state vector of the motion parameters given as:

$$\mathrm{Z}_{\dagger} = \mathrm{Z}_{\dagger}^{i} - \mathrm{Z}_{\dagger}^{j} = \begin{bmatrix} x_{\dagger} & y_{\dagger} & \Delta x_{\dagger} & \Delta y_{\dagger} & \vartheta_{\dagger} \end{bmatrix}^{\mathrm{T}}. \tag{47}$$

The subsequent portion presents the simulation outcomes referring to the fluctuation of measured noise in the form of motion approximates, minimal location error, velocity error, rotation predicts, error histogram, and regression evaluation.

- The results of motion parameter prediction for coordinated turn trajectory are presented in Figure 10a for all approaches. It can be seen that the soft computation paradigm based on SCGNI reveals higher accuracy compared to the typical IMMEKF and IMMUKF techniques. It is noticeable that multimode filters have greater challenges in accurately estimating the motion parameters of maneuvering objects during turns of the trajectory compared to the SCGNI approach, which highlights the proficiency of the neural paradigm.
- The results presented in Figure 10b illustrate the real-time positional error of IMM filters and SCGNI, specifically in the sense of the mean square. The accuracy of SCGNI outweighs that of other approaches for every instance of turning track.
- Figure 10c displays the velocity error, measured in meters per second, of the maneuvering object for all data points across each technique. The performance of filtering algorithms deteriorates at the turning points of the trajectory. However, it is seen that the SCGNI algorithm outperforms the IMM filters at all data points.
- Figure 10d displays the turn rate projections for all techniques, revealing that neural computing offers much superior estimation performance compared to Kalman filters.
- Figure 10e presents a comparison analysis of the target time series $\mathbf{Z}(\dagger - 1)$, $\mathbf{Z}(\dagger - 2), \ldots, \mathbf{Z}(\dagger - \mathrm{j})$, and the predicted value $\mathbf{Z}_{\dagger}^{Estimated}$ of the target's kinetics in the format of histogram. The histogram displays a vertical bar at its center, which represents an error of -0.02163. The peak of the vertical stack utilized for the procedure of training exceeds 500 points, whereas the testing and validation datasets consist of a range of 400 to 500 occurrences. The analysis indicates that the zero error is located within the vertical bin, namely at a value of -0.02163.
- Figure 10f provides an illustration of the regression phenomena observed during the neural learning procedure. The graph depicts a minimal deviation among the target number and the output variable, that can be attributed to a rise in the observed noise standard deviation.

In the present scenario, the MSEs of the orientation and velocity calculated in the units of meters and meters/second respectively, for both the actual and anticipated values of the target's location and velocity. The location and velocity errors detected, further validate the aforementioned findings, indicating that the neural network presents much superior accuracy compared to multimodel filtering approaches. The chart given in Figure 11 displays the location and velocity errors that have been determined employing the IMMEKF, IMMUKF, and SCGNI methods.

4.1.4. Scenario 4: The Measurement Noise Standard Deviation λ = 0.5 Rad

In scenario 4, the standard deviation of observed noise is $\lambda = 0.5$ radians, so injecting a significant level of Gaussian noise into the motion prediction process. The symbolic representation of variance, which incorporates the arithmetic quantity of standard deviation of measured noise, can be expressed as:

$$\mathrm{O}_{\dagger} = \mathrm{diag}(\lambda_{\mathrm{M}}^{2}). \tag{48}$$

The computation of Gaussian measurement noise is derived from the covariance as:

$$\Im_{\dagger}^{n} \approx \mathrm{N}(0, \mathrm{O}_{\dagger}). \tag{49}$$

The noise measured at time step † for each sensor n is included into the measurement formulation of the overall system as:

$$\mathrm{M}_{\dagger}^{n} = \arctan\left[\frac{y_{\dagger} - \Xi_{y}^{n}}{x_{\dagger} - \Xi_{x}^{n}}\right] + \Im_{\dagger}^{n}. \tag{50}$$

The input time series provided to the SCGNI neural model consists of the whole measurement model equation $M_{\dagger}^{n}$, which covers passive bearings and measured noise. The target time series is an additional input that represents the real state vector, as seen below:

$$Z_{\dagger} = Z_{\dagger}^{i} - Z_{\dagger}^{j} = \begin{bmatrix} x_{\dagger} & y_{\dagger} & \Delta x_{\dagger} & \Delta y_{\dagger} & \vartheta_{\dagger} \end{bmatrix}^{T}. \tag{51}$$

The instantaneous locations, velocities, and rotations of a passive dynamical object have been formulated for the exact bending route of an undersea target. These parameters are then input into a neural network based on SCGNI to generate a target time series.

The figure above shows simulation results for motion predicts, location error, velocity error, rotation approximates, deviation histogram, and regression interpretation.

- Figure 12a displays the motion parameter predictions of the IMMEKF, IMMUKF, and SCGNI algorithms under conditions of high measurement noise. It is evident that all three algorithms encounter challenges in accurately tracking the actual trajectory. However, even within the presence of a disruptive environment, it is shown that the estimations of SCGNI exhibit a higher degree of convergence with the actual trajectory compared to the other two algorithms that are considered.
- The figure labeled as Figure 12b illustrates the MSE between the existing and computed positions of the kinetic target. The outcomes indicates that filtering algorithms exhibit a significant level of error, while the SCGNI deep learning approach demonstrates a more accurate estimation of the object's turning trajectory with reduced position error.
- Figure 12c displays the velocity error resulting from various approaches, so providing validation for the effectiveness of the artificial neural network.
- Figure 12d presents the turn rate approximations in this particular scenario, showcasing the superior prediction capability of neural computing compared to IMM filtering approaches.
- Figure 12e displays a histogram illustrating the occurrence of errors between the target values $\mathbf{Z}(\dagger - 1), \mathbf{Z}(\dagger - 2), \ldots, \mathbf{Z}(\dagger - \mathrm{j})$, and the estimated output value $\mathbf{Z}_{\dagger}^{Estimated}$ for the target's motion. A discrepancy of -0.00234 is seen in the tall box located at the midpoint of the graph. The histogram represents the distribution of a training dataset, with a vertical length of around 300 samples. Likewise, dataset for testing and validation consist of approximately 250 to 300 points. The zero-error bar in the histogram is positioned below the vertical line centered at -0.00234.
- The regression analysis in this particular scenario is presented in Figure 12f, which clearly illustrates a notable disparity between the true value and the anticipated output. This divergence can be attributed to a boost in the standard deviation of the observed noise.

In this situation, the MSE is computed to determine the disparity between the true and projected velocity and location of the dynamic vehicle. The location and velocity error outcomes further validate the earlier findings that SCGNI shows superior performance compared to Kalman filters. The location and velocity variations, as computed by GPB and SCGNI methodologies, are illustrated in Figure 13.

4.1.5. Scenario 5: The Measurement Noise Standard Deviation λ = 1 Rad

In the final scenario of this analysis, a peak value of λ = 1 radian is adopted to represent an atmosphere with high level of noise. Additionally, the motion prediction system will be subjected to a measurement model characterized by significant levels of noise. The highest value of λ corresponds to the definition of covariance $O_{\dagger}$ as:

$$O_{\dagger} = \mathrm{diag}(\lambda_{M}^{2}). \tag{52}$$

At instantaneous time †, the computation of Gaussian distributed measurement noise $\Im_{\dagger}^{n}$ for n sensor is performed using the covariance $O_{\dagger}$ as:

$$\Im_{\dagger}^{n} \approx N(0, O_{\dagger}). \tag{53}$$

The measurement expression now incorporates the inclusion of extreme values of independent white Gaussian measurement noise as:

$$\mathrm{M}_{\dagger}^{n} = \arctan\left[\frac{y_{\dagger} - \Xi_{y}^{n}}{x_{\dagger} - \Xi_{x}^{n}}\right] + \Im_{\dagger}^{n}. \tag{54}$$

The input time series $\mathrm{M}_{\dagger}^{n}$ of the SCGNI deep learning network consists of passive measurements with maximal measured noise. Another input of neural intelligence is derived from the state expression, as specified below:

$$\mathrm{Z}_{\dagger} = \mathrm{Z}_{\dagger}^{i} - \mathrm{Z}_{\dagger}^{j} = \begin{bmatrix} x_{\dagger} & y_{\dagger} & \Delta x_{\dagger} & \Delta y_{\dagger} & \vartheta_{\dagger} \end{bmatrix}^{\mathrm{T}}. \tag{55}$$

The SCGNI smart computing integrates both input and target datasets for accurately predicting the output time series, which comprises the expected vector of motion parameters. The following statistics are shown in this circumstance: state predictions, location velocity disparities, rotation forecasts, error bar graph, and regression model.

- The motion parameter estimations of the IMMEKF, IMMUKF, and SCGNI algorithms in this particular situation are illustrated in Figure 14a. The divergence of motion prediction results across all methods can be attributed to the presence of significant noise in the undersea domain. Tracking a true trajectory poses significant challenges for all algorithms. However, in this specific scenario, the neural computing technology known as SCGNI demonstrates superior performance for estimating turning trajectory as compared to other two approaches.
- The average gap between real and predicted positions of the underwater maneuvering object is depicted in Figure 14b as the MSE. The results clearly demonstrate that the SCGNI computing network presents lower position error compared to the IMMEKF and IMMUKF methods.
- Figure 14c displays the velocity error outcomes of all approaches, revealing that SCGNI has occasional peaks but, on the whole, demonstrates superior performance compared to IMM filters.
- Figure 14d presents the turn rate estimates in this particular scenario, whereby the turning parameter is more accurately estimated using SCGNI.
- Figure 14e illustrates the assessment of a neural network through the utilization of an error histogram. This study includes the target dataset $\mathbf{Z}(\dagger - 1), \mathbf{Z}(\dagger - 2), \ldots, \mathbf{Z}(\dagger - \mathrm{j})$, as well as the approximated output vector $\mathbf{Z}_{\dagger}^{Estimated}$. An error of 0.06423 is detected in a vertical bin with a training dataset height of around 250 steps, while the datasets used for confirmation and testing possess heights ranging from 150 to 250 instances. During this particular circumstance, the zero-divergence value is situated below the vertical box having a central value of 0.06423.
- The figure labeled as Figure 14f illustrates the regression of the SCGNI soft learning network, specifically depicting the relationship among the desired outcome and the projected result in this particular scenario. The regression analysis reveals a notable disparity between the observed target values and the anticipated output, which can be attributed to the substantial presence of normal distributed noise in the motion prediction framework.

Within this cluttered environment, the MSEs pertaining to the orientation and velocity of the vehicle are computed in the units of meters and meters per second separately, by comparing the actual values with their corresponding estimated values. The obtained location and velocity error outcomes validate the prior findings that SCGNI exhibits superior accuracy compared to Kalman filters, especially in highly noisy underwater conditions. Figure 15 presents a compilation of position and velocity errors that have been determined through the IMMEKF, IMMUKF, and SCGNI methods.

The aforementioned data from various circumstances indicates that when the standard deviation of measured noise λ is high, all motion parameter prediction algorithms en-

counter difficulties in accurately tracking the true state of a turning maneuvering item in an underwater environment. When considering various methods comparatively, it is evident that deep learning based on SCGNI exhibits superior performance. This effectiveness is demonstrated in its ability to accurately anticipate nonlinear real-time motion parameters in underwater scenarios.

4.2. Computational Load Analysis

This section focuses on analyzing the computational complexity of the proposed SCGNI computing and comparing it with the operational difficulty of traditional IMM filters. We evaluate the quantity of multiplication and addition steps incorporated in these approaches. The instantaneous computations performed at every time instant † necessary for the joint operations of SCGNI and the Bayesian filters, are represented in Table 2. Here Q represents the quantity of filters in the filter-bank of IMMEKF and IMMUKF, whereas c and d represent the sizes of state vector $Z_{\dagger}$ and measurement vector $\mathbf{H}_{\dagger}$ correspondingly. The IMM filter necessitates the real-time computations of the Bayesian gain and the merging of covariance, as well as the prediction and update steps, at each moment. It is assumed that the matrix reversal tasks, which are needed to calculate the filter gains, are performed utilizing Cholesky transformation [48]. As well as for SCGNI, m g, and z are representing inputs, hidden layers and outputs of the network respectively.

Table 2. Computational load analysis of IMMEKF, IUMMUKF and SCGNI

Algorithm	Operation	Additions	Multiplications
IMMEKF IMMUKF	Combining covariance	$Q^2(c^3+c)$	$Q^2(c^3+2c)$
	Estimating covariance	$2Qc^3$	$Q(2c^3-c^2)$
	Updating covariance	$Q^2(c^2d+c^3)$	$Q(c^3+c^2(d-1))$
	Filter gain	$Q(2c^2d+2d^2c+0.5d^3+1.5d^2)$	$Q[cd(2n-3)+2d^2c+0.5d^3-0.5d^2]$
SCGNI	Neurons	$g+z$	$m(g^2+z)$
	Memory	$g[o(m)]+z[o(g)]$	$o[(m+z)*g^2]$
	Feed time	$g*o(m)+z*o(g)$	$o[m+z(g^3)]$

Additionally, the complexity assessment for the IMMEKF, IMMUKF, and SCGNI techniques is conducted using statistical parameters of mean and standard deviation over time in seconds. The outcomes of complexity operators for all observed noise conditions are presented in Table 3 for all techniques. The IMMEKF algorithm demonstrates the shortest execution time in seconds, with IMMUKF algorithm closely following. The computational time required for the SCGNI computing strategy is more than that of multi model Kalman filters. However, this difference becomes less significant because to the higher accuracy of the SCGNI.

Table 3. Complexity analysis for different values of Measurement Noise Standard Deviation.

Algorithm	λ = 0.01 rad		λ = 0.05 rad		λ = 0.1 rad		λ = 0.5 rad		λ = 1 rad	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
IMMEKF	0.05	0.02	0.07	0.03	0.14	0.09	0.37	0.26	0.64	0.42
IMMUKF	0.08	0.03	0.17	0.06	0.26	0.14	0.58	0.43	0.83	0.75
SCGNI	0.12	0.04	0.29	0.13	0.53	0.25	0.92	0.69	1.73	1.23

5. Conclusions

This study examines the deep learning computing paradigm utilizing the robustness of SCGNI for the real-time prediction of motion parameters of bearings-only maneuvering Markov chain undersea object. In the two-dimensional X-Y coordinate framework, the instantaneous motion characteristics of a kinematic turning target are predicted at each time instant. At first, a mathematical BOT approach is used to create a state space prediction design for a dynamic and observed framework. After that, a neural computing method based on SCGNI is developed to foresee the motion characteristics of a Markov chain passive object. We evaluated the performance of SCGNI supervised neural computing for absolute turning course of target movements through the use of rotation prediction, minimum mean square location, real-time motion approximation, velocity difference, inaccuracy histogram, and regression on 200 data samples. Subsequently, a sufficient number of numerical values representing Gaussian dispersed observed noise are utilized to evaluate the success rate of the intended approach. The final section presents the simulation outcomes, which demonstrate that the neural network offers a higher level of accuracy compared to conventional nonlinear multi model Bayesian filtering algorithms such as IMMEKF and IMMUKF. Even so, the outcomes of all approaches demonstrate an exponential deterioration in a noisy underwater atmosphere. Hence, the acquisition of precise motion characteristics within a complex oceanic scenario remains an enormous task, presenting significant potential for further advancement.

In future research tasks, the exploration of recurrent and radial base neural strategies can be pursued to enhance the estimation of motion parameters for highly maneuvering objects in the presence of non-Gaussian measurement noise. This area of investigation remains a captivating research avenue within the domain of underwater single or multi-target scenarios.

Author Contributions: Conceptualization, W.A. and X.Q.; methodology, W.A. and H.H.Z.; software, X.Q.; validation, H.H.Z., X.Q. and A.A.; formal analysis, W.A., X.Q. and A.J.; investigation, H.H.Z. and A.J.; resources, X.Q., A.A. and A.J.; data curation, W.A. and X.Q.; Writing—original draft, W.A.; Writing—review and editing, H.H.Z., X.Q. and A.M.; visualization, W.A. and X.Q.; supervision, X.Q.; project administration, A.A., A.J. and X.Q.; funding acquisition, A.A., A.J. and A.M. 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 (Grant No. 11774074).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Jacinto, E.; Martinez, F.; Martinez, F. Navigation of Autonomous Vehicles using Reinforcement Learning with Generalized Advantage Estimation. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 1401103. [CrossRef]
2. Liu, K.; Pan, L.; Zhang, L.; Gao, R.; Xu, C.; Zhang, L.; Zhang, Q. Indoor localization for an unknown radio emitter employing graph-based optimization and improved RSSD. *AEU Int. J. Electron. Commun.* **2023**, *171*, 154909. [CrossRef]
3. Wang, S.; Wang, Z.; Dong, H.; Chen, Y. Distributed state estimation under random parameters and dynamic quantizations over sensor networks: A dynamic event-based approach. *IEEE Trans. Signal Inf. Process. Netw.* **2020**, *6*, 732–743. [CrossRef]
4. Theiler, M.; Schneider, D.; Endisch, C. Kalman Filter Tuning Using Multi-Objective Genetic Algorithm for State and Parameter Estimation of Lithium-Ion Cells. *Batteries* **2022**, *8*, 104. [CrossRef]
5. Zhang, H.; Xie, W. Constrained unscented Kalman filtering for bearings-only maneuvering target tracking. *Chin. J. Electron.* **2020**, *29*, 501–507. [CrossRef]
6. Nok To, C.; Milani, S.; Marzbani, H.; Jazar, R.N. Improvement of the autodriver algorithm for autonomous vehicles using roll dynamics. *Arch. Motoryz.* **2021**, *91*, 5–23.
7. Koller, T.L.; Frese, U. The interacting multiple model filter and smoother on boxplus-manifolds. *Sensors* **2021**, *21*, 4164. [CrossRef]

8. Li, P.; Pei, X.; Chen, Z.; Zhou, X.; Xu, J. Human-like motion planning of autonomous vehicle based on probabilistic trajectory prediction. *Appl. Soft Comput.* **2022**, *118*, 108499. [CrossRef]
9. Wang, W.; Li, X.; Zhang, K.; Shi, J.; Shi, W.; Ali, W. Robust Direction Finding via Acoustic Vector Sensor Array with Axial Deviation under Non-Uniform Noise. *J. Mar. Sci. Eng.* **2022**, *10*, 1196. [CrossRef]
10. Deng, F.; Levi, C.; Yin, H.; Duan, M. Identification of an Autonomous Underwater Vehicle hydrodynamic model using three Kalman filters. *Ocean Eng.* **2021**, *229*, 108962. [CrossRef]
11. Zhang, Z.; Liu, Z.; Zhang, H. Underwater mobile gravity measurement data processing using continuous-discrete Kalman filter. *AIP Adv.* **2021**, *11*, 085104. [CrossRef]
12. Fan, D.; Shen, H.; Dong, L. Multi-Agent Distributed Deep Deterministic Policy Gradient for Partially Observable Tracking. *Actuators* **2021**, *10*, 268. [CrossRef]
13. 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.* **2022**, *68*, 2941–2952. [CrossRef]
14. Mahmoudreza, H.; Hamid, K.; Mohammadtaghi, B. Augmented input estimation in multiple maneuvering target tracking. *J. Syst. Eng. Electron.* **2019**, *30*, 841–851.
15. Sheng, H.; Zhao, W.; Wang, J. Interacting multiple model tracking algorithm fusing input estimation and best linear unbiased estimation filter. *IET Radar Sonar Navig.* **2017**, *11*, 70–77. [CrossRef]
16. Zubača, J.; Stolz, M.; Seeber, R.; Schratter, M.; Watzenig, D. Innovative interaction approach in imm filtering for vehicle motion models with unequal states dimension. *IEEE Trans. Veh. Technol.* **2022**, *71*, 3579–3594. [CrossRef]
17. 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]
18. Grootveld, A.; Bugayev, V.I.; Lackey, L.; Klein, A.G.; Vedula, K.; Brown, D.R. Tracking of dynamical processes with model switching using temporal convolutional networks. In Proceedings of the 2021 IEEE Aerospace Conference (50100), Big Sky, MT, USA, 6–13 March 2021; pp. 1–9.
19. Kumar, M.; Mondal, S. Recent developments on target tracking problems: A review. *Ocean Eng.* **2021**, *236*, 109558. [CrossRef]
20. Han, P.; Ting, C.; Xi, L. De-correlated unbiased sequential filtering based on best unbiased linear estimation for target tracking in Doppler radar. *J. Syst. Eng. Electron.* **2020**, *31*, 1167–1177. [CrossRef]
21. Ali, W.; Khan, W.U.; Raja, M.A.Z.; He, Y.; Li, Y. Design of nonlinear autoregressive exogenous model based intelligence computing for efficient state estimation of underwater passive target. *Entropy* **2021**, *23*, 550. [CrossRef]
22. Akbari, B.; Zhu, H. Tracking dependent extended targets using multi-output spatiotemporal Gaussian processes. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18301–18314. [CrossRef]
23. Sarker, I.H. Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective. *SN Comput. Sci.* **2021**, *2*, 377. [CrossRef]
24. Hartmann, M.; Hashmi, U.S.; Imran, A. Edge computing in smart health care systems: Review, challenges, and research directions. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3710. [CrossRef]
25. Gad, A.G. Particle swarm optimization algorithm and its applications: A systematic review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [CrossRef]
26. Telikani, A.; Tahmassebi, A.; Banzhaf, W.; Gandomi, A.H. Evolutionary machine learning: A survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [CrossRef]
27. Rodríguez, J.V.; Rodríguez-Rodríguez, I.; Woo, W.L. On the application of machine learning in astronomy and astrophysics: A text-mining-based scientometric analysis. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1476. [CrossRef]
28. Döpp, A.; Eberle, C.; Howard, S.; Irshad, F.; Lin, J.; Streeter, M. Data-driven science and machine learning methods in laser–plasma physics. *High Power Laser Sci. Eng.* **2023**, *11*, e55. [CrossRef]
29. Zubatiuk, T.; Nebgen, B.; Lubbers, N.; Smith, J.S.; Zubatyuk, R.; Zhou, G.; Koh, C.; Barros, K.; Isayev, O.; Tretiak, S. Machine learned Hückel theory: Interfacing physics and deep neural networks. *J. Chem. Phys.* **2021**, *154*, 244108. [CrossRef]
30. Masi, F.; Stefanou, I. Multiscale modeling of inelastic materials with Thermodynamics-based Artificial Neural Networks (TANN). *Comput. Methods Appl. Mech. Eng.* **2022**, *398*, 115190. [CrossRef]
31. Garnier, P.; Viquerat, J.; Rabault, J.; Larcher, A.; Kuhnle, A.; Hachem, E. A review on deep reinforcement learning for fluid mechanics. *Comput. Fluids* **2021**, *225*, 104973. [CrossRef]
32. Junior, R.F.R.; dos Santos Areias, I.A.; Campos, M.M.; Teixeira, C.E.; da Silva, L.E.B.; Gomes, G.F. Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals. *Measurement* **2022**, *190*, 110759. [CrossRef]
33. Souza, R.M.; Nascimento, E.G.; Miranda, U.A.; Silva, W.J.; Lepikson, H.A. Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Comput. Ind. Eng.* **2021**, *153*, 107060. [CrossRef]
34. Zhang, P.; Hu, Y.; Jin, Y.; Deng, S.; Wu, X.; Chen, J. A Maxwell's equations based deep learning method for time domain electromagnetic simulations. *IEEE J. Multiscale Multiphys. Comput. Tech.* **2021**, *6*, 35–40. [CrossRef]
35. Danandeh Mehr, A.; Rikhtehgar Ghiasi, A.; Yaseen, Z.M.; Sorman, A.U.; Abualigah, L. A novel intelligent deep learning predictive model for meteorological drought forecasting. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 10441–10455. [CrossRef]
36. Zuo, C.; Qian, J.; Feng, S.; Yin, W.; Li, Y.; Fan, P.; Han, J.; Qian, K.; Chen, Q. Deep learning in optical metrology: A review. *Light Sci. Appl.* **2022**, *11*, 39. [CrossRef]

37. Sakshi; Kukreja, V. A dive in white and grey shades of ML and non-ML literature: A multivocal analysis of mathematical expressions. *Artif. Intell. Rev.* **2023**, *56*, 7047–7135. [CrossRef]
38. Massoud, Y.M.; Ahmad, A.A.; Abdelzaher, M.; Kuhlmann, L.; Abd El Ghany, M.A. Hardware implementation of deep neural network for seizure prediction. *AEU Int. J. Electron. Commun.* **2023**, *172*, 154961. [CrossRef]
39. Bai, H.; Wu, S. Deep-learning-based nanowire detection in AFM images for automated nanomanipulation. *Nanotechnol. Precis. Eng. (NPE)* **2021**, *4*, 013002. [CrossRef]
40. 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. [CrossRef] [PubMed]
41. Farizawani, A.; Puteh, M.; Marina, Y.; Rivaie, A. A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches. *J. Phys. Conf. Ser.* **2020**, *1529*, 022040. [CrossRef]
42. Miao, X.; Li, J.; Mu, Y.; He, C.; Ma, Y.; Chen, J.; Wei, W.; Gao, D. Time Series Forest Fire Prediction Based on Improved Transformer. *Forests* **2023**, *14*, 1596. [CrossRef]
43. Landin, C.; Liu, J.; Katsarou, K.; Tahvili, S. Time Series Anomaly Detection using Convolutional Neural Networks in the Manufacturing Process of RAN. In Proceedings of the 2023 IEEE International Conference On Artificial Intelligence Testing (AITest), Athens, Greece, 17–20 July 2023; pp. 90–98.
44. Møller, M.F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* **1993**, *6*, 525–533. [CrossRef]
45. Cao, J.; Wu, J. A conjugate gradient algorithm and its applications in image restoration. *Appl. Numer. Math.* **2020**, *152*, 243–252. [CrossRef]
46. Yuan, G.; Li, T.; Hu, W. A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems. *Appl. Numer. Math.* **2020**, *147*, 129–141. [CrossRef]
47. Yan, Z.; Zhong, S.; Lin, L.; Cui, Z. Adaptive Levenberg–Marquardt algorithm: A new optimization strategy for Levenberg–Marquardt neural networks. *Mathematics* **2021**, *9*, 2176. [CrossRef]
48. Hunger, R. *Floating Point Operations in Matrix-Vector Calculus*; Insttute for Circuit Theory and Signal, Munich University of Technology: Munchen, Germany, 2005; Volume 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.