Proceedings of the 30th International Conference on Ocean, Offshore and Arctic Engineering
OMAE2011
June 19-24, 2011, Rotterdam, The Netherlands

OMAE2011-50248

COLLABORATED AND CONSTRAINED NEURAL-EKF ALGORITHM FOR THE VESSEL TRAFFIC MONITORING AND INFORMATION SYSTEM

Lokukaluge P. Perera
Centre for Marine Technology and Engineering (CENTEC),
Technical University of Lisbon, Instituto Superior Técnico,
Av. Rovisco Pais,1049-001 Lisboa, Portugal
Email : prasad.perera@mar.ist.utl.pt

Paulo Oliveira
Institute for Systems and Robotics (ISR),
Technical University of Lisbon, Instituto Superior Técnico,
Av. Rovisco Pais,1049-001 Lisboa, Portugal
Email : pjcro@isr.ist.utl.pt

C. Guedes Soares
Centre for Marine Technology and Engineering (CENTEC),
Technical University of Lisbon, Instituto Superior Técnico,
Av. Rovisco Pais,1049-001 Lisboa, Portugal
Email : guedess@mar.ist.utl.pt

ABSTRACT

Maneuvering vessel detection and tracking in cooperation with vessel state estimation and navigational trajectory prediction are important tasks for the Vessel Traffic Monitoring and Information Systems (VTMIS) to improve maritime safety and security in ocean navigation. In this study, collaborated and constrained Neural-EKF algorithm is proposed for the above purpose. The proposed methodology consists of two main units: an Artificial Neural Network based Vessel Detection and Tracking Unit and an Extended Kalman Filter based State Estimation and Trajectory Prediction Unit. Finally, the proposed algorithm, is implemented on the MATLAB software platform, and successfully illustrate the results attainable in respect to vessel detection and tracking, vessel state estimation and navigational trajectory prediction in ocean navigation is also presented in this study.

NOMENCLATURE

R_{L1}	Neural network constrained learning range
R_{L2}	The sensor maximum measurement range
$r_j(k)$	j-th position range vector
$r(k)$	Accumulated range vector
$\vartheta_j(k)$	j-th position bearing vector
$\vartheta(k)$	Accumulated bearing vector
${}^c x_j(k)$, ${}^c y_j(k)$	j-th position x, y coordinates
$x_j(k), y_j(k), z_j(k)$	Normalized j-th position x, y, z coordinates
$p(k)$	Accumulated normalized position vector
$p_j(k)$	j-th position vector
$W(k)$	Accumulated prototypes vector
$w_i(k)$	i-th prototype vectors
$n(k)$	Neural network input
$a(k)$	Neural network output vector
$\boldsymbol{\theta}_i(k)$	Angle between $p_j(k)$ and $w_i(k)$
α	Learning rate of the Instar rule
$x(t)$, $y(t)$	Vessel position x, y coordinates
$v_x(t)$, $v_y(t)$	Vessel x, y directional velocity components
$a_n(t)$, $a_t(t)$	Vessel normal, tangential acceleration
$V_a(t)$	Vessel speed
$\chi_a(t)$	Vessel course
$x(t)$	Nonlinear vessel state vector
$f(x(t))$	Nonlinear vessel state function
$w(t) \sim N(0,Q(t))$	Vessel state noise vector
$z(t)$	Measurement vector
$z_x(k)$, $z_y(k)$	Measurement x, y vessel coordinates
$h(x(t))$	Measurement function
$v(t) \sim N(0,R(t))$	Measurement noise vector
$Q(t)$	Vessel state noise covariance
$R(k)$	Measurement noise covariance
$x(t)$	Nonlinear vessel state vector
$\tilde{x}(t)$	Vessel state error vector
$\hat{x}(t)$	Estimated vessel state vector
$x(k^-)$, $x(k^+)$	Estimated prior and posterior state vectors
$f(x(t))$	Nonlinear vessel state function

w(t)	Vessel state noise vector
z(t)	Measurement vector
h(x(k))	Measurement function
v(k)	Measurement noise vector
Q(t)	Vessel state noise covariance
R(k)	Measurement noise covariance
P(t)	Estimated error covariance
$P(k^-)$, $P(k^+)$	Estimated prior and posterior error covariance of state vectors
K(k)	Kalman filter gain

INTRODUCTION

The European Union (EU) is surrounded by one of the busiest and most complex sea route systems in the world. Over 90% of the EU external trade goes by sea and over 3.7 billion tones of the freight per year are transferred through the EU ports. In addition, passenger traffic in the seas around the EU regions is presently approximated to 350 million passenger journeys per year [1].

With the increase demand for maritime transportation of passengers and freights in the EU maritime regions, the safety and security issues in ocean navigation are highlighted. Furthermore, a maritime monitoring mechanism for that purpose is proposed by the EU Directive 2002/59 [2], where the highly dense maritime traffic regions are to be equipped with the regional Vessel Traffic Monitoring and Information Systems (VTMISs). The conventional ocean navigational systems and VTMISs are equipped with several maritime surveillance systems (i.e. Radar, Laser, Ladar, Automatic Radar Plotting Aid (ARPA), Automatic Identification System (AIS), and Long-Range Identification and Tracking System (LRIT)) for the same purpose. However, there are many challenges faced by those maritime surveillance systems as presented in [3]: The larger surveillance volume, synchronization among targets and sensors, noisy signal propagation environment and multi-target observations. Therefore, the integration of advanced features into the maritime surveillance systems as proposed in this study are required to overcome the challenges faced in ocean navigation.

However, there are some advanced features developed under current maritime surveillance systems of ARPA and AIS. The ARPA system provides accurate information on range and bearing of nearby vessels and the AIS is capable of giving information on the vessel structural data, position, course, and speed, etc. The AIS marine traffic simulator, aiming to perform navigation safety and security studies, is presented in [4]. Even though the ARPA and AIS systems were developed to provide navigation aids (ie. detection and tracking facilities) to vessels, the vessel state estimation and its navigational trajectory prediction tasks are still under developed.

The main advantages of vessel state estimation and navigational trajectory prediction tasks are the intention of the vessel and the collision risk among vessels to be predicted ahead of time. Furthermore, those tasks are also important tools to Long-Range Identification and Tracking (LRIT) system, that will be implemented as an international maritime security network, where each respective country has sufficient time to evaluate and to response to security risk that is posted by each vessel to its coastline [2]. Furthermore, by obeying to the conditions of national and international maritime laws, by each vessel, can also be monitored by these systems where the coastal security and safety issues should not be compromised.

The VTMIS proposed in this work is presented in Figure 1. As depicted in the figure, the system consists of two main units: ANN (Artificial Neural Network) based Vessel Detection unit and Tracking Unit and EKF (Extended Kalman Filter) based Trajectory Estimation and Prediction Unit. The combination of these units, collaborated and constrained Neural-EKF algorithm that is the main objective in this paper, is described in the following sections.

The work presented in this study is a part of the ongoing effort to formulate an Intelligent Collision Avoidance System in ocean navigation, as further described by [5] and [6]. The organization of this paper is as follows. The second section contains an overview of proposed Vessel Traffic Monitoring and Information System (VTMIS). An ANN based vessel detection and tracking process and an EKF based state estimation and trajectory prediction process are presented in the third and forth sections respectively. The fifth section contains a detailed description of computational simulations and discussions. Finally, the conclusion and future work are presented in the sixth section.

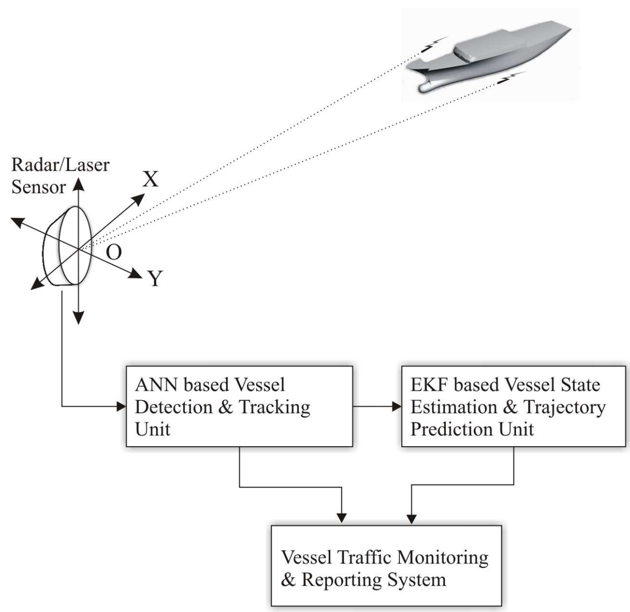

Figure 1. The Proposed VTMIS

VESSEL TRAFFIC MONITORING AND INFORMATION SYSTEM

The mathematical formulation of the VTMIS that is presented in Figure 1 is further discussed in this section. A multi-vessel situation under the radar/laser sensor measurements is presented in Figure 2. The radar/laser sensor is located in position O (0, 0) and the i-th vessel at the k-th time instant is located at the position $A_i(k)$. As presented in the figure, each vessel is identified as a cluster of data points that is observed by the radar/laser sensor. The sensor range is divided into two region with respect to radius of R_{L2} and R_{L1} as presented in the figure.

In the region between regions R_{L2} and R_{L1} the vessel detection and tracking processes are executed. In the region R_{L1} the vessel states (i.e. position, velocity and acceleration) estimation and navigational trajectory prediction processes are executed. Furthermore, the initial neurons that are used for the vessel detection and tracking process are located in the boundary layer region of R_{L2} as further described in the third section.

Sensor Measurement and Coordinate Systems

After radar/laser scan, the sensor generates the corresponding range and bearing values of the vessels and obstacles in the environment, at k-th time instant can be written as:

$$\begin{aligned} r(k) &= [r_1(k)\ r_2(k) \dots r_R(k)] \\ \vartheta(k) &= [\vartheta_1(k)\ \vartheta_2(k) \dots \vartheta_R(k)] \end{aligned} \quad (1)$$

The range and bearing values are transformed into the Cartesian coordinated in the j-th position at the k-th time instant can be written as:

$$\begin{aligned} {}^c x_j(k) &= r_j(k) \cos(\vartheta_j(k)) \\ {}^c y_j(k) &= r_j(k) \sin(\vartheta_j(k)) \end{aligned} \quad (2)$$

However, these position data points should be normalized with respect to the maximum range of the radar/laser sensor. The normalized j-th position coordinates of the vessel at the k-th time instant can be written as:

$$\begin{aligned} x_j(k) &= \frac{{}^c x_j(k)}{R_{L2}} \\ y_j(k) &= \frac{{}^c y_j(k)}{R_{L2}} \end{aligned} \quad (3)$$

However, for the fair neural competition these position vectors should be formulated for unit magnitude conditions as further described in the following section, where the third dimension, $z_i(k)$, is introduced, and that can be written as:

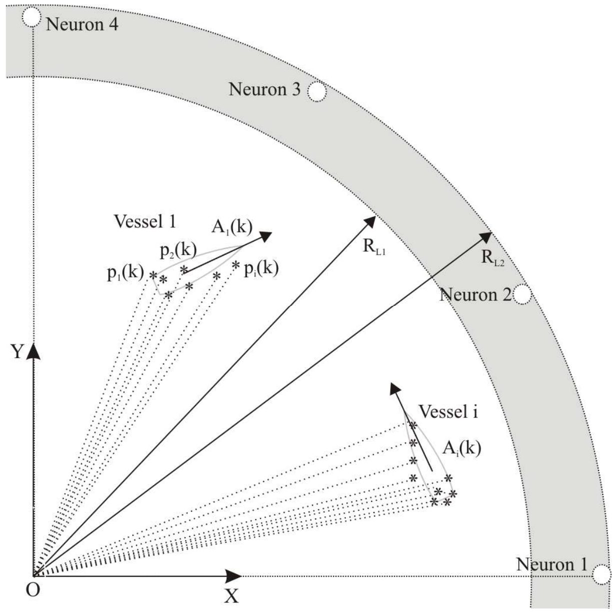

Figure 2. Multi-vessel Radar/Laser Measurement

$$\sqrt{x_j^2(k) + y_j^2(k) + z_j^2(k)} = 1 \quad (4)$$

The coordinate of the $z_i(k)$ can be calculated considering the above unit magnitude conditions for the j-th data point and can be written as:

$$z_j(k) = \sqrt{1 - x_j^{\,2}(k) - y_j^{\,2}(k)} \quad (5)$$

One should note that the introduction of a third dimension ($z_i(k)$) can be evaluated as a transformation of 2D position coordinates into 3D position coordinates. The complete 3D j-th position data point at the k-th time instant can be written as:

$$p_j(k) \equiv [x_j(k)\ \ y_j(k)\ \ z_j(k)] \quad (6)$$

Therefore, the complete data cluster (see Figure 3) that is generated by the sensor measurements can formulate as:

$$p(k) \equiv [p_1(k)\ \ p_2(k)\ \dots\ p_R(k)] \quad (7)$$

ANN BASED VESSEL DETECTION AND TRACKING

The theoretical foundation of artificial neurons is formulated from observations of biological concepts as well as inspirations of the behavior of human brain and nervous system. An artificial neuron consists of several inputs that correspond to the synapses of a biological neuron. Furthermore, it consists of only one output that corresponds to the axon of a biological neuron. However, each input corresponds to a certain weight value that influences the corresponding signal over the neuron output.

Furthermore, in an artificial neuron this concept is mathematically formulated by a transfer function. The Transfer function calculates the sum of the net-input with respect to the weight values and compare with certain threshold levels to generate the neurons output [7]. The accumulation of several neurons in series and /or parallel formation can categorize as an Artificial Neural Network (ANN).

Competitive Neural Network

Artificial neural network (ANN) approach has been successful applied in the problems of clustering and classification in recent literature with respect to stationary data conditions [8]. However, in this study, moving data conditions are proposed and this is one of the novel contributions. The constrained Competitive Neural Networks (CNN) approach [9] associated with the Instar rule is proposed in this study for the detection and tracking of multi-vessel conditions in ocean navigation. The proposed constrained CNN is presented in Figure 3.

Detection and Tracking of Multi-vessel Conditions

The constrained CNN is formulated for detection and tracking of vessels. However, each vessel is approximated for a cluster of data points, where the constrained CNN is trained to detect each moving data clusters that are entering into region R_{L2} by competing its neurons. The vessels entering into region R_{L2} are detected by the neurons that are assigned in the boundary layer of the region R_{L2} (see Figure 2). Twelve neurons around the boundary layer of region R_{L2} are assigned to detect new vessels, as multiple data clusters, entering into the sensor range. Therefore, when a vessel enters to the boundary R_{L2} as a data cluster, the neuron closer to the data cluster gets excited and starts to detect the vessels.

The proposed constrained CNN is presented in Figure 3. As presented in the figure, the constrained CNN consists of three sub-units: Constrained Weight vector sub-unit (**W**), Constrained Competition sub-unit (**C**), and Feedback-loop (Instar Rule). The input to the CNN consists of a accumulated data position vector p(k). The prototypes vectors (neurons), W(k), are stored as rows vectors in the Weight vector sub-unit that are the weights of the CNN. Initially, twelve neurons are considered and that are distributed along the boundary layer of the region R_{L1}(see Figure 2). However, each new vessel entering into the region R_{L2}, the system observed the neuron that get excited by the data cluster. Then the excited neuron will add as a new neuron into the weight vector and it modified weight values will be reset to the original values. Hence, there are will be twelve neurons always located alone the boundary layer of the region R_{L2} to detect new vessel entering into the sensor region.

However, in the approach the number of the Weight vectors always increase with the number of vessels entering into the region R_{L2}. Therefore, the growing number of the Weight vector

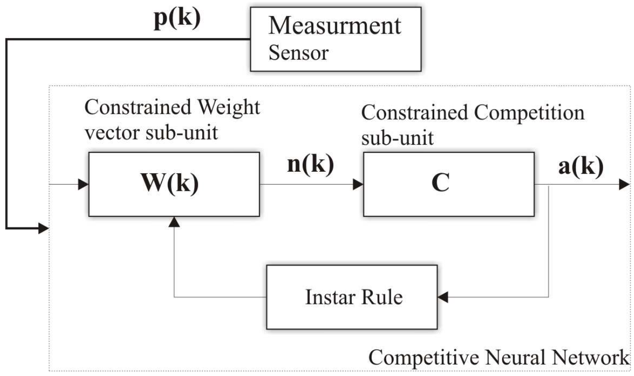

Figure 3. ANN based Detection and Tracking Unit

could originate computational problems in this approach. Hence, to overcome this problem, the system monitors the changes in the values of the added Weight vectors and if no changes observed after several iterations, then the Weight vectors will be removed. Therefore, the proposed mechanism is implemented on the Constrained Weight vector sub-unit, that will remove the additional neurons with the exit of the vessel from region R_{L2}.

The net input, n(k), is output of the constrained Weight vector sub-unit that is a vector product related to the distance among position vectors $p_j(k)$ and each prototype vectors $w_i(k)$ where $W(k) = [w_1(k)\ w_2(k)\ \dots\ w_S(k)]$. Hence, for a fair competition among neurons, each position vector, $p_j(k)$ and prototype vectors $w_i(k)$ should have unit magnitude conditions. The calculations for unit magnitude conditions for each position vector, $p_j(k)$ was presented in the previous section. Similar unit magnitude conditions for the prototype vectors $w_i(k)$ is adopted in this study.

The net input of the i-th neuron $n_i(k)$, the vector dot product between data input vector $p_j(k)$ and the prototype vector $w_i(k)$, and that can be written as:

$$\mathrm{n}_i(k) = \mathrm{W}(k)\mathrm{p}_j(k) = \begin{bmatrix} \mathrm{w}_1^{\mathrm{T}}(k)\mathrm{p}_j(k) \\ \mathrm{w}_2^{\mathrm{T}}(k)\mathrm{p}_j(k) \\ \vdots \\ \mathrm{w}_S^{\mathrm{T}}(k)\mathrm{p}_j(k) \end{bmatrix} = \begin{bmatrix} \cos\theta_1(k) \\ \cos\theta_2(k) \\ \vdots \\ \cos\theta_S(k) \end{bmatrix} \tag{8}$$

In the competitive unit **C** (see Figure 2) of the CNN, the distances among input vector $p_j(k)$ to each prototype vector $w_i(k)$ are compared using a vector dot product. Then the neuron whose weight vector in the direction closest to the input vector is assigned output of 1 and others are assigned by 0 by the transfer function. The competitive unit, **C,** consists of a transfer function that is used to generate competition among neurons. Hence, the neural competition can be written as:

$$a_i(k) = compet\left(n_i(k)\right) = compet\left(W(k)\, p_j(k)\right) \tag{9}$$

The competitive (compet) transfer function can be defined as:

$$compet\left(n_i(k)\right) = \begin{cases} 1 & \text{for neron with max } n(k) \\ 0 & \text{all other nerons} \end{cases} \tag{10}$$

The net input, $n(k) = [n_1(k)\ n_2(k)\ \ldots\ n_R(k)]$., is the input to the Competition sub-unit, **C**, and the output vector a(k), is the output from Competition unit, **C**, at the time k-th time instant. Finally, the feedback loop, associated with the Instar Rule, is proposed to adapt the weight values of prototypes vectors. At the initial stage, the values of the prototype weight vectors, W(k) of the CNN, are assumed to be unknown that layer around the region R_{L2}. However, to adapt neurons to track the moving vessels that a formulated by clusters of data points, the adaption of the weight vectors, W(k) is required and that could be done by a learning rule.

The learning rule is proposed in this section to calculate the appropriate values for the weight vectors. This concept is called the unsupervised learning. When the competitive layer excites the neuron that is closest to the data cluster, then the learning rule will be used to adapt appropriate weight values in the prototype vectors of the CNN. Further Instar Rule is proposed for the unsupervised adaptation of the CNN to change the default weights values of the of the CNN and can be written as:

$$W(k) = W(k-1) + \alpha a(k)\left(p^T(k) - W(k-1)\right) \tag{11}$$

More details on the Instar Rule that is proposed as the unsupervised learning rule for training of the CNN in this study can be found in [10].

Constrained Neural Competition

The above process can be further elaborated as a situation where the closest neuron gets excited by the data cluster and the winning neuron takeovers all the data points in the respective data cluster. However, after winning the data cluster, the weight values of the respective neuron should be adapted. Hence newly updated wining neuron gets closer to the respective data cluster in real-time implementation. This process will continue with the complete dynamic data cluster in each time instant. Therefore, the Instar learning rule is introduced to facilitate proper dynamic weight update in winning neurons.

However, several drawbacks are observed implementing CNN in complex environments with several stationary and moving targets as presented in [10]: The several neurons can track different parts of the same target and/or one neuron can track several targets in close proximity. Therefore, the following constrained are introduced into the CNN sub-unit (see

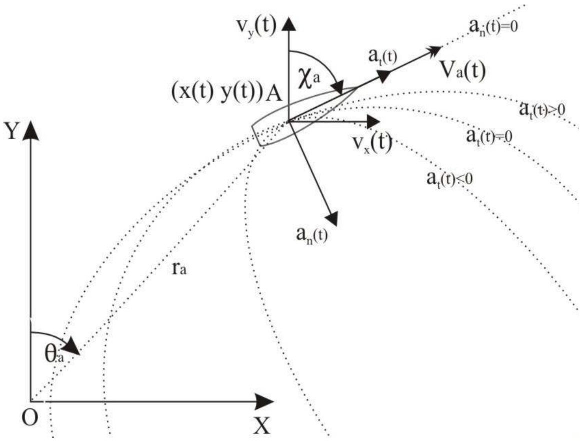

Figure 4. Curvilinear Motion Reference System

Figure 3): Constrained Weight vector sub-unit and constrained Competition sub-unit.

In the first region (Region between R_{L1} and R_{L2}) the constrained Weight vector sub-unit is implemented. When a target vessel enters into this region the neurons in the boundary layer start to track the respective data cluster. However, in a normal situation the neuron closer to the vessel get excited and start to track the respective data cluster. However, in some situations two neurons may get excited in the CNN. However, this can happen due to either two vessels are entering into the same region or two neurons are tracking the different parts of the same data clusters of the respective vessel.

Therefore, to overcome the above drawback constrains into the Weight vector sub-unit is introduced. When two neurons, Weight vectors, closer to each other get excited there is a mechanism is introduced to check the distance between two neurons. If the distance is smaller then a given threshold value, then the both neurons are considered as a one neuron, otherwise it will be treated as a two separate vessel and two neurons tracking will be continued.

In the second region (Region R_{L1}) the constrained Competition sub-unit is implemented. When several data cluster get closer to each other, there is a possibility of respective neurons of the data clusters can jump in-between data clusters in close proximity. This is similar to the situation, when several neurons are tracking different parts of the same target and/or one neuron is tracking several targets, as discussed preciously. Therefore to overcome the above failures and some network constrains into the Competition sub-unit are introduced.

When a neuron is tracking a vessel, a mechanism is introduced to check the distance between neuron and the data cluster: If the distance among neuron and the data clusters are grater then a give threshold values, then the neuron will ignore the respective data clusters. Therefore, this mechanism can stop the neurons jumping between data clusters in close proximity.

STATES ESTIMATION AND TRAJECTORY PREDICTION

The next step in this process, the average position of respective vessel that is observed by the winning neuron of the constrained CNN is forwarded to the EKF based vessel states estimation and navigational trajectory prediction sub-system that is further discussed in this section. Therefore, this section is elaborated into three sections: The Target Motion Model (TMM), Measurement Model and Associated Techniques (MAT) and Trajectory Tracking and Estimation (TTE).

Target Motion Model

A suitable mathematical model for tacking an ocean vessel in ocean navigation is considered in this section for the vessel states estimation and navigational trajectory prediction. In the scenario considered, with the average vessel position values from the constrained CNN, the vessel motion model is assumed to be a point target with negligible dimensions.

Considering the above requirements, a continuous-time curvilinear motion model is proposed, as depicted in Figure 4. In fact, the curvilinear motion model capability of capturing the multi-model features is one of the advantages in this approach. The standard continuous-time curvilinear motion model as the target motion model can be written as [11]:

$$\begin{aligned}
\dot{\chi}_a(t) &= \frac{a_n(t)}{V_a(t)} \\
\dot{\mathbf{V}}_{\mathbf{a}}(t) &= a_t(t) \\
v_x(t) &= V_a(t)\sin(\chi_a(t)) \\
v_y(t) &= V_a(t)\cos(\chi_a(t))
\end{aligned} \tag{12}$$

The summarized curvilinear motion model considering Equations (12) are formulated as a target motion model:

$$\dot{x}(t) = f(x(t)) + w(t) \tag{13}$$

where

$$x(t) = \begin{bmatrix} x_v(t) \\ v_x(t) \\ y_v(t) \\ v_y(t) \\ a_t(t) \\ a_n(t) \end{bmatrix}, \qquad f(x(t)) = \begin{bmatrix} v_x(t) \\ a_t(t) f^{vx} + a_n(t) f^{vy} \\ v_y(t) \\ a_t(t) f^{vy} - a_n(t) f^{vx} \\ 0 \\ 0 \end{bmatrix}.$$

$$f^{vx} = \frac{v_x(t)}{\sqrt{v_x^2(t) + v_y^2(t)}}, \quad f^{vy} = \frac{v_y(t)}{\sqrt{v_x^2(t) + v_y^2(t)}}$$

The Jacobian matrix of the target motion model can be written as:

$$\frac{\partial}{\partial x}\left(f(x(t))\right) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_t(t) f_{vx}^{vx} + a_n(t) f_{vx}^{vy} & 0 & a_t(t) f_{vy}^{vx} + a_n(t) f_{vy}^{vy} & f^{vx} & f^{vy} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_t(t) f_{vx}^{vy} - a_n(t) f_{vx}^{vx} & 0 & a_t(t) f_{vy}^{vy} - a_n(t) f_{vy}^{vx} & f^{vy} & -f^{vx} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

where

$$f_{vx}^{vx} = \frac{v_y^2(t)}{\left(v_x^2(t) + v_y^2(t)\right)^{3/2}}, \qquad f_{vy}^{vx} = -\frac{v_y(t) v_x(t)}{\left(v_x^2(t) + v_y^2(t)\right)^{3/2}}$$

$$f_{vx}^{vy} = -\frac{v_x(t) v_y(t)}{\left(v_x^2(t) + v_y^2(t)\right)^{3/2}}, \qquad f_{vy}^{vy} = \frac{v_x^2(t)}{\left(v_x^2(t) + v_y^2(t)\right)^{3/2}}$$

Measurement Models and Associated Technique

The measurement model is formulated as a discrete-time linear model due to ocean vessel positions available in discrete time instants, and can be written as:

$$z(k) = h(x(k)) + v(k) \tag{15}$$

where the measurement states can be written as:

$$z(k) = \begin{bmatrix} z_x(k) \\ z_y(k) \end{bmatrix}, \; h(x(k)) = \begin{bmatrix} x_v(k) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & y_v(k) & 0 & 0 & 0 \end{bmatrix}$$

The Jacobian matrix of the measurement model can be written as:

$$\frac{\partial}{\partial x}\left(h(x(k))\right) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Trajectory Tracking and Estimation

The Kalman Filter (KF) is the optimal solution to estimate the state of linear systems, thus it is inadequate for the problem at hand. However, the Extended Kalman Filter (EKF) is a popular technique to solve a number of non-linear system applications. The summarized Extended Kalman Filter algorithm is presented in [12], and can be written as:

- System Model

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + w(t) \\ w(t) &\sim N(0, Q(t)) \end{aligned} \tag{16}$$

- Measurement Model

$$\begin{aligned} z(k) &= h(x(k)) + v(k) \\ v(k) &\sim N\left(0, R(k)\right), \; k = 1,2,... \end{aligned} \tag{17}$$

- Error Conditions

$$\tilde{x}(k) = \hat{x}(k) - x(k) \tag{18}$$

- State Initial Conditions

$$x(0) \sim N\left(\hat{x}(0), P(0)\right) \tag{19}$$

where $\hat{x}(0)$ is the state initial estimate and P(0) is the state initial covariance values, describing the uncertainty present on the initial estimates. All stochastic disturbances are assumed as zero mean and Gaussian.

- Uncorrelated process and measurements noises

$$E\left[v(t);\ w(k)\right] = 0 \quad \text{for all } k, t \tag{20}$$

- State Estimation Propagation

$$\dot{\hat{x}}(k) = f(\hat{x}(k)) \tag{21}$$

- Error Covariance Extrapolation

$$\dot{P}(t) = F(\hat{x}(t))P(t) + P(t)F^{T}(\hat{x}(t)) + Q(t)$$
$$F(\hat{x}(t)) = \frac{\partial}{\partial x(t)} f(x(t)) \bigg|_{x(t)=\hat{x}(t)} \tag{22}$$

- Estimate State Update

At each step, after measurement data is available from the sensors, the state estimates can be updated

$$\hat{x}(k^{+}) = \hat{x}(k^{-}) + K(k)\left[z(k) - h_{k}\left(\hat{x}(k^{-})\right)\right] \tag{23}$$

- Error Covariance Update

$$P(k^{+}) = \left[1 - K(k)H_{k}\left(\hat{x}\left(k^{-}\right)\right)\right] P(k^{-})$$
$$H\left(\hat{x}\left(k^{-}\right)\right) = \frac{\partial}{\partial x(k)} h(x(k)) \bigg|_{x(k)=\hat{x}\left(k^{-}\right)} \tag{24}$$

- Kalman Gain Computation

$$K(k) = P(k^{-})H\left(\hat{x}\left(k^{-}\right)\right)\left[H\left(\hat{x}\left(k^{-}\right)\right)P(k^{-})H\left(\hat{x}\left(k^{-}\right)\right)^{T} + R(k)\right]^{-1} \tag{25}$$

The detailed description of the EKF implementation for state estimation and trajectory prediction also can be found in [13].

COMPUTATIONAL SIMULATIONS AND DISCUSSION

The computational simulations of detection and tracking of vessels represented by three groups of data clusters, its state estimation, and navigational trajectory prediction are detailed in this section. The computational simulations are implemented in the MATLAB software platform and consist of three loops: i) the vessel scanning loop, ii) the vessel detection and tracking loop, and iii) the state estimation and navigational trajectory prediction loop.

Vessel detection and tracking

The computational simulations of tracking of three vessels, represented by three groups of data clusters are presented in Figure 4. The three data clusters that are moving under varying acceleration conditions consist of vessel position data points that are simulated as actual trajectories in this study.

The main objective of the LMS target scanning loop is to scan the environment and collect the new position of the target as a new data cluster. Then, this data will be transferred into the vessel detection and tracking loop. The main objective of the vessel detection and tracking loop is to adapt the constrained CNN to track the new position of the vessel by updating its weights of the CNN by the learning rule.

As depicted in the figure, the initial weight values of the constrained CNN are assigned alone to the boundary layer of region R_{L2}. Finally, respective neurons adapt their weight values to track the movement of each cluster of data points. Furthermore, it can be observed that each tracker converges to the approximate mean value of the respective data cluster. This position is considered as the targets measurement position, at each time instant, represented by small circles in Figure 5.

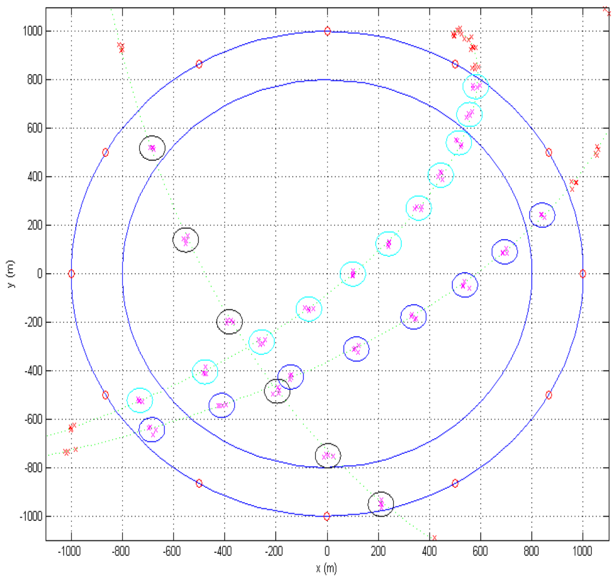

Figure 5. Target Detection and Tracking Simulation

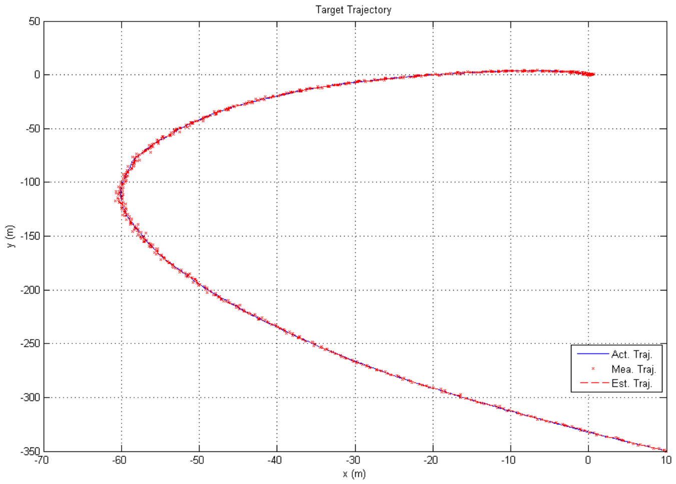

Figure 6. Navigational Trajectories

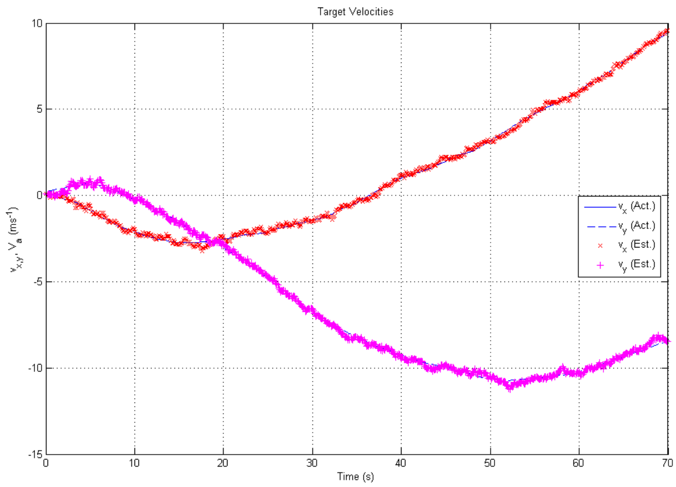

Figure 7. Vessel Velocities

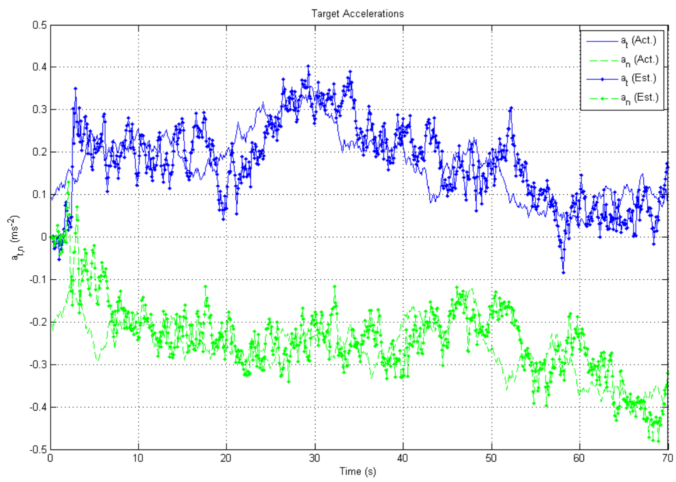

Figure 8. Vessel Accelerations

Vessel state estimation and trajectory prediction

The simulations for vessel state estimation and navigational trajectory prediction for ocean navigation, based on the EKF algorithm, are presented in Figures 6, 7 and 8. The data cluster that is moving under varying acceleration conditions consists of vessel position is simulated as an actual trajectory in this study. Furthermore, the actual trajectory positions added with sensor noise are considered as the measurement trajectory positions in this study.

Figure 6 presents the actual trajectory (Act. Traj), measured trajectory (Mea. Traj.) and estimated trajectory (Est. Traj.) of vessel navigation. As noted from the Figure, the EKF estimated vessel navigations trajectory successfully. The vessel velocity components of $v_x(t)$ and $v_y(t)$ of actual (Act.) and estimated (Est.) values are depicted in Figure 7. The figure represents the estimation process of the estimated (Est.) velocity components to the actual (Act.) velocity components. As noted from the Figure, the EKF estimated vessel velocity components successfully. The acceleration components of $a_t(t)$ and $a_n(t)$ of actual (Act.) and estimated (Est.) values are depicted in Figure 8. The figure represents the estimation process of the estimated accelerations (Act.) into the actual accelerations (Est.) for normal and tangential acceleration components. As noted from the Figure, the EKF estimated vessel acceleration components successfully with some small error variations.

CONCLUSION

The novelty in this study is that the target dimensions are explicitly considered during the detection and tracking process, as most of the target tracking methods assume that the target is a point or approximated small data cluster. Furthermore, a popular state-of-the-art machine learning application for neural network unsupervised learning algorithms is successfully implemented and validated in simulation. Moreover, experimental results are obtained for vessel detection and tracking.

Even though in general neural network applications are extensively used for recognition of clusters of static data patterns, the recognition of clusters of moving data patterns can also be tackled by the proposed method. Furthermore, the neural network approach can be further developed not only for target tracking but also for classification and identification of targets. Hence, more advanced neural network integration approaches for object classification and identification will be considered as future developments of this study.

The estimation of ocean vessel position, velocities and accelerations has been also successfully achieved by the EKF given the curvilinear motion model selected and the linear measurement model provided by commercially available ranging sensors. The estimated values for the velocity components have small errors, varying around the actual values and with negligible errors due to errors on the acceleration estimation. The ocean navigation consists of changing acceleration conditions, as presented in this paper. One should note that the acceleration estimation is achieved by using only position measurement data collected from the vessel navigation, which is another contribution in this study. Therefore, theses estimated vessel state conditions could be used for successful estimation of the vessel navigation trajectories that eventually helps on the prediction process of the intension of the vessel and the collision risk among vessels in ahead of time.

ACKNOWLEDGMENTS

This work has been made within the project ”Methodology for ships maneuverability tests with self-propelled models”, which is being funded by the Portuguese Foundation for Science and Technology under contract PTDC/TRA/74332 /2006. The research work of the first author has been supported by a Doctoral Fellowship of the Portuguese Foundation for Science and Technology under contract SFRH/BD/46270/2008.

REFERENCES

[1] EMSA Report, “Operational activities : European maritime safety agency,” http://www.emsa.europa.eu/, January 2009.

[2] Directive 2002/59/EC of the European Parliament and of the Council, “Establish a community vessel traffic monitoring and information system and repealing council directive 93/75/EEC,” *Official Journal of the European Communities*, no. 208, pp. 10–27, 2002.

[3] D. L. Hall and J. Llinas, “An introduction to multisensor data fusion,” in *Proceedings of the IEEE*, vol. 85, no. 1, 1997.

[4] K. Hasegawa, “Advanced marine traffic automation and management system for congested waterways and coastal areas,” in *Proc. Int. Conf. in Ocean Engineering (ICOE)*, Chennai, India, 2009, pp. 1–10.

[5] Moreira, L.; Fossen, T. I., and Guedes Soares, C. Path Following Control System for a Tanker Ship Model, *Ocean Engineering.* 34:2074-2085. 2007;

[6] L. P. Perera, J. P. Carvalho, and C. Guedes Soares, Fuzzy-logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. *Journal of Marine Science and Technology,* 2011, doi 10.1007/s00773-010-0106-x

[7] M. Cirstea, A. Dinu, J. Khor, and M. McCormick, Eds., *Neural and Fuzzy Logic Control of Drives and Power Systems*, 1st ed. MA, USA: Elsevier Science, 2002.

[8] K. L. Du, “Cluster: A neural network approach,” *Neural Networks*, vol. 23, pp. 89–107, 2010.

[9] M. T. Hagan, H. B. Demuth, and M. H. Beale, Eds., *Neural Network Design*. Boston: PWS Publishing, 1996.

[10] L. P. Perera and C. Guedes Soares, “Laser measurement system based maneuvering target tracking formulated by adaptive competitive neural networks,” in *In Proc. 2nd Int. Conf. on Adaptive and Self-adaptive Systems and Applications*, Lisbon, Portugal, 2010, pp 84-90.

[11] R. A. Best and J. P. Norton, “A new model and efficient tracker for target with curvilinear motion,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 1030–1037, July 1997.

[12] A. Gelb, J. F. Kasper, Jr., R. A. Nash, Jr., C. F. Price, and A. A. Sutherland, Jr., *Applied Optimal Estimation*. MA. USA: The MIT Press, 2000.

[13] L. P. Perera and C. Guedes Soares, “Ocean vessel trajectory estimation and prediction based on Extended Kalman filter,” in *Proc. 2nd Int. Conf. on Adaptive and Self-adaptive Systems and Applications*, Lisbon, Portugal, 2010, pp 14-20.