GEODEZJA I KARTOGRAFIA
t. XLIX, z. 4, s.183-193 (2000)

*Andrzej Stateczny*

Department of Navigation
Maritime University of Szczecin
email: astat@wsm.szczecin.pl

# Neural state estimation in sea navigation

The article presents the putting into practice of a neural state estimator for navigational measurements. One of the basic problems of sea navigation is considered, which is the statistic working out of navigational measurements with the purpose of determining the estimated vector of the vessel's movement and position.

## 1. *Introduction*

One of the basic problems of sea navigation is reproducing the actual vector of an object's state on the basis of gathered information, frequently burdened with errors and incomplete. A solution to this problem was provided, i.a. by Kalman. A computer estimation of state by means of a Kalman filter requires a large calculation expenditure, and the determination of filtration error covariance matrix is a numerically unstable process, which creates the necessity of applying special factorisation methods. The applied methods of non-linearity approximation do not always warrant the correctness of object state reproduction, in our case the vessel's state vector, and the Kalman filters thus obtained, called extended filters, make it considerably more difficult and increase the calculation expenditure when putting it into practice.

Taking into account the above conditioning a lot of sub-optimal algorithms of Kalman filter [2,3] and calculation procedures were worked out, thus increasing the effectiveness of putting the designed filters into practice computer-wise. Independently of results obtained in this range, the problem of high numerical load and adapting the filters for strongly non-linear objects still remains. Within this context, the application of neural networks for solving problems of dynamic objects' state reproduction seems to be justified for two main reasons. Firstly, there is the possibility of obtaining a fast (parallel transformation) and signal-indeterminacy resistant (adaptation and learnability) Kalman filter both for linear and non-linear objects. Secondly, there seems to be possible a neural production of a filter for strongly non-linear objects, for which an extended Kalman filter constitutes an inadmissible approximation.

## 2. *Numerical methods of state estimation*

Let the dynamic system be given by means of a system of vector differential equations with the matrices of time-dependent coefficients [1, 2]:

$$x_{i+1/i} = F_{i+1} \times x_{i+1} + W_{i+1}$$
$$z_{i+1} = G_{i+1} \times x_{i+1} + V_{i+1}$$

where:

$x_{i+1/i}$ – $n$ - dimensional state vector,
$W_{i+1}$ – $n$ - dimensional vector of non-measurable input,
$z_{i+1}$ – measurable output vector,
$V_{i+1}$ – $m$ - dimensional vector of measurement disturbances,
$F_{i+1}$ – system matrix of dimensions $(n \times m)$,
$G_{i+1}$ – measurement system matrix of dimensions $(n \times m)$,
$i = 1,2,\ldots$ – discrete time.

It is assumed that [1,2]:

- $W_{i+1}$ is an uncorrelated vector of zero average constituting a random sequence of Gaussian distribution,
- $V_{i+1}$ is an uncorrelated vector of random sequences of Gaussian distribution,
- initial condition $x(0)$ is a vector of random sequences of normal distribution,
- there is no correlation between $W_{i+1}$ and $V_{i+1}$ and the state vector $x(0)$.

There should be determined the estimate $x_{i+i}$ of the state vector $x_i$ on the basis of a measurably accessible sequence of output vectors $Y_1$, $Y_2$,...,$Y_i$, in such a way that the estimate $x_{i+i}$ should minimise the accuracy indicator.

If the above-mentioned conditions are fulfilled, then the optimal Kalman filter is described by the following system of differential equations [1,2]:

- vector equations of $x_{i+i}$ estimate:

$$x_{i+i} = x_{i+1/i} + K_{i+1} \times [z_{i+1} - G_{i+1} \times x_{i+1/i}]$$
$$x_{i+1/i} = F_{i+1} \times x_i$$

- covariance matrix equations of estimation error $P_{i+1}$ and the amplification matrix:

$$K_{i+1} = P_{i+1/i} \times G_{i+1}^T \times \left[G_{i+1} \times P_{i+1/i} \times G_{i+1}^T + R_{i+1}\right]^{-1}$$
$$P_{i+1/i} = F_{i+1} \times P_{zi} \times F_{i+1}^T + P_i$$
$$P_{i+1} = P_{i+1/i} - K_{i+1} \times G_{i+1} \times P_{i+1/i}$$

where:

$x_{i+1/i}$ – extrapolation vector of $x_{i+1}$ estimate one step forward (vector of estimation position coordinates),
$K_{i+1}$ – amplification matrix,
$z_{i+1}$ – vector of measurements,

$G_{i+1}$ – matrix of gradients,

$P_{i+1/i}$ – covariance matrix of estimated position coordinates, which takes into account errors due to estimating the position during correction,

$P_i$ – covariance matrix of previous estimated position coordinates,

$P_{zi}$ – covariance matrix of accretion vector of estimated position coordinates,

$F_{i+1}$ – transition matrix,

$R_{i+1}$ – covariance matrix of measurement vector.

Kalman filter functions correctly in the period of stable work, but it shows considerable discrepancies in the initial estimation period. The generalised method of least squares uses the estimated vector $x_{zi}$ based on running measurements instead of the vector from the previous estimation step. The form of the generalised method of least squares is as follows [2]:

$$
\begin{aligned}
x_{i+1} &= x_{zi+1} + K_{i+1} \times \left[z_{i+1} - G_{i+1} \times x_{zi+1}\right] \\
K_{i+1} &= P_{i+1/i} \times G_{i+1}^T \times \left[G_{i+1} \times P_{i+1/i} \times G_{i+1}^T + R_{i+1}\right]^{-1} \\
P_{i+1/i} &= F_{i+1} \times P_{zi} \times F_{i+1}^T + P_i \\
P_{i+1} &= P_{i+1/i} - K_{i+1} \times G_{i+1} \times P_{i+1} \\
x_{zi+1} &= F_{i+1} \times x_{zi}
\end{aligned}
$$

The ship's movement is characterized by variable parameters in the range of its stability. During stable work of the vessel's propulsion system (rudder and screws) the changes of movement parameters are small and therefore the information about real movement parameters in relation to the bottom obtained from the previous correction period are more valuable than in the case when there are big changes in speed and course. Therefore, during stable movement the Kalman filter method should be applied; during considerable changes in the movement parameters on the other hand and in the initial estimation period – the method of least squares. A method like this, which combines the merits of the methods described above, is the intermediate estimation method. The algorithm of this estimation method is described by the following relationships [2]:

$$
\begin{aligned}
x_{i+1} &= x_{pi+1} + K_{i+1} \times \left[z_{i+1} - G_{i+1} \times x_{i+1/i}\right] \\
K_{i+1} &= P_{i+1/i} \times G_{i+1}^T \times \left[G_{i+1} \times P_{i+1/i} \times G_{i+1}^T + R_{i+1}\right]^{-1} \\
P_{i+1/i} &= F_{i+1} \times P_{zi} \times F_{i+1}^T + P_i \\
P_{i+1} &= P_{i+1/i} - K_{i+1} \times G_{i+1} \times P_{i+1/i} \\
x_{pi+1} &= F_{i+1} \times x_{pi}
\end{aligned}
$$

for $x_{pi} = w_1 \times x_{zi} + w_2 \times x_i$

at the same time: $w_1 + w_2 = 1$

where: $w_1$, $x_2$ – weight coefficients.

It can be noticed that when $w_1 = 0$ the Kalman filter is obtained; in the case of $w_2 = 0$, on the other hand, the generalized method of least squares. A smooth change in the weight coefficients should allow to use the merits of both above-mentioned methods in the highest degree.

As numerous numerical experiments have shown this methods helps to obtain a much better estimate of object state than traditional methods of state estimation. To adduce results of experiments conducted would exceed the scope of this work.

## 3. *A neural state estimator*

The perceptron type of neural network, i.e. with feedforward signal flow, was used for building the model of a neural state estimator. The network is made up of three neural layers. The first layer, also called the input layer, contains two neurons and it fulfils the function of entrance into the network. The input signal from the first layer is given, after multiplication of the weight of particular entrances, to the hidden layer, which contains from twenty to two hundred neurons. Every neuron of the hidden layer determines its answer to the input signals obtained and similarly transmits it to each of the two neurons of the output layer. Similarly to the previous case signals determined by the hidden layer are multiplied by the weight coefficient of the output layer neural entrances. There is a strictly defined direction of signal flow – from the input (where the network is given signals that are input data, specifying the task to be solved), to the output, where the network gives the established solution.

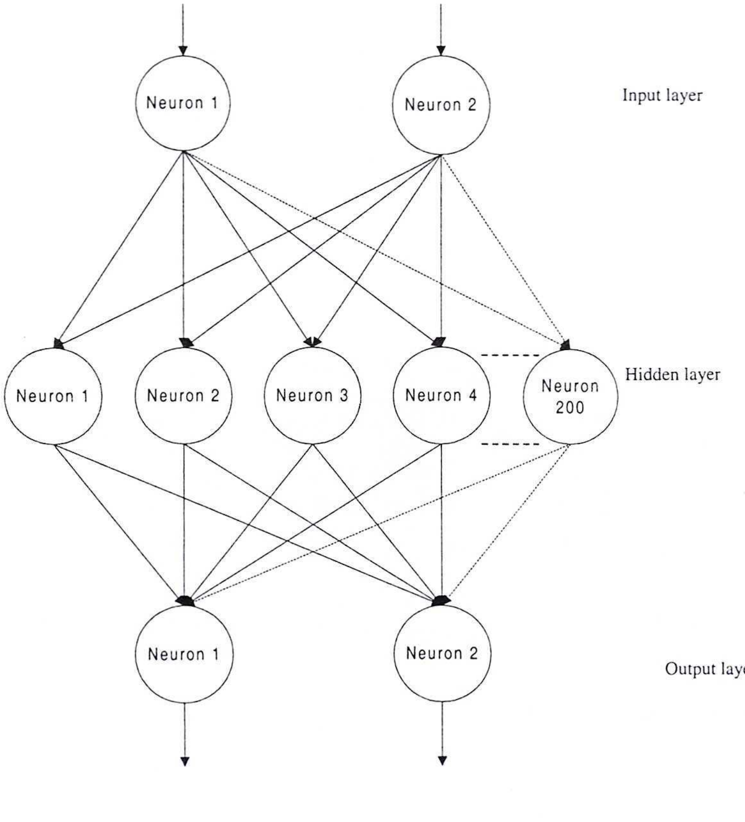


Fig. 1. Diagram of the perceptron applied in the neural state estimator

The number of elements of the input layer (two neurons) corresponds to the number of input data (two components of the estimated speed vector). Its task is to receive the steering signal and transform it into information accessible to the neurons of next successive layers. The essence of this process is scaling the value of the input signal in the range <0.1>.

**The delta rule** was applied for teaching the network described above. Let us recall that it consists in every neuron determining its output signal on receiving signals at their entrances, making use of previously determined values of the amplification coefficient (of weights) of all the entrances and (possibly) of the threshold. The value of the output signal, determined by the neuron in the particular stage of the teaching process, is compared with the pattern answer given by the teacher (intermediate estimation method) in the teaching sequence. If there are discrepancies, the neuron determines the difference between its own output signal, and the value of the signal correct according to the teacher.

In the case of one-layer network the situation is simple and self-evident: the output signal of every neuron is compared with the correct value given by the teacher, which provides a sufficient basis for weight correction.

In a multilayer network (a network like this was applied in the state estimator neural model discussed), the situation is more complicated. The neurons of the output layer can have their errors estimated in a relatively simple and certain way – as previously, by comparing the signal produced by every neuron with the pattern signal given by the teacher. On the other hand, neural errors of earlier layers must be estimated mathematically, as they cannot be directly measured because of lack of information what the values of respective signals should be (the teacher does not define these intermediate values, concentrating exclusively on the final effect).

A method commonly applied for estimating errors of hidden layers is the **backpropagation** method. It consists in reproducing supposed errors of deeper layers of the network on the basis of a back-projection of errors detected in the output layer. That means, while considering a neuron of the hidden layer there are considered all the errors of all these neurons the output signal was sent to, and they are summed up, considering the coefficient values of weight connections between the neuron under consideration with the neurons, the errors of which are summed up. Acting in this way and proceeding from the output to the input of the network, supposed errors of all the neurons are designated, thus gaining a basis for defining corrections of all the weight coefficients of these neurons.

As results from the above considerations, in the model of a neural state estimator there was applied the variant of teaching the network by means of a "teacher" (it being in this case the intermediate method) based on rules described above. The process of teaching the network consists in the presenting to the network a set of exemplary input signals (input pictures) and the corresponding output signals (output pictures). A set of these examples (otherwise called the teaching sequence) is presented to the network as long as the network does not work out a correct output signal. The accuracy to be attained by the network in the teaching process is determined by the size of the error. This error is defined as the difference between the output signal determined by the network at a given stage of the teaching process and the pattern answer given by the teacher in the teaching sequence.

The teaching sequence is completed while the numerical estimator is working. This sequence contains twenty of so-called teaching pairs, constituted by: the vector calculated on the basis of running measurements and the corresponding estimated vector obtained from the Kalman-type numerical estimator.

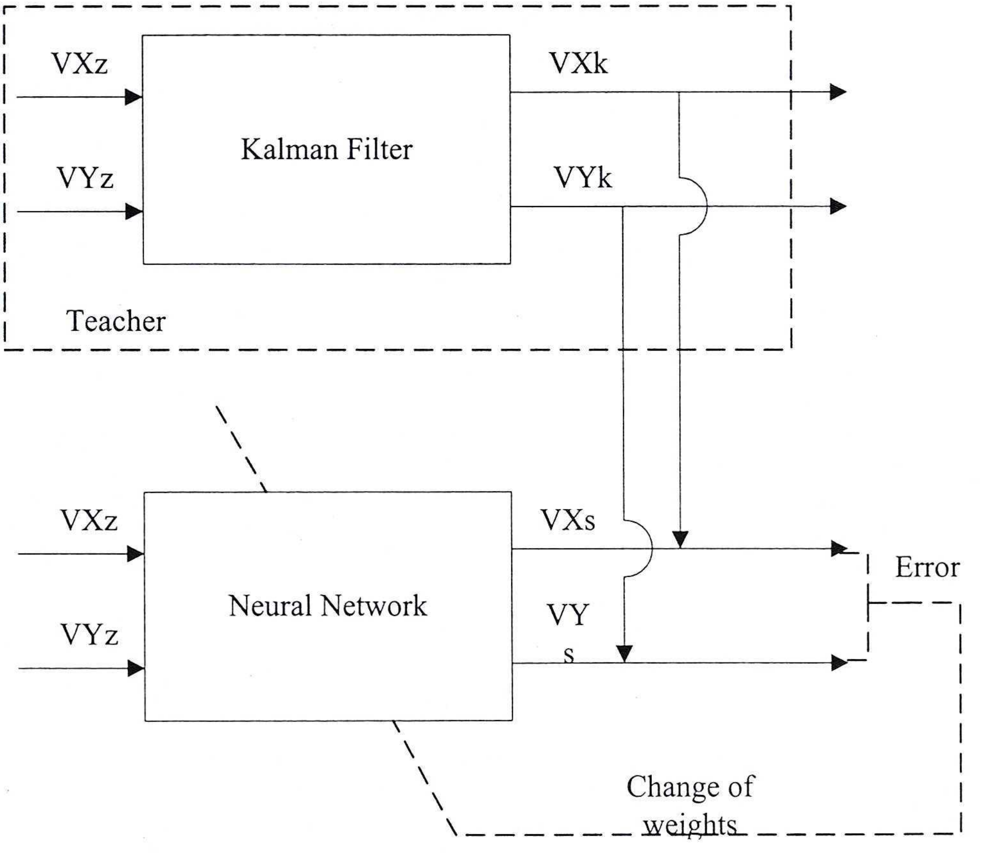


Fig. 2. A diagram of numerical-neural system estimating the vector of the object state

Fig.2 presents a schematic diagram of the activity of a constructed neural object state estimator.

The cycle of the numerical – neural system can be divided into four stages:

1. The functioning of the numerical filter up to the moment of the estimation process stabilising. It was assumed that the stage would last one minute, i.e. twenty estimation stages.

2. The second stage is the stage for gathering information for the teaching sequence. This stage lasts from the first to the second estimation minute (20-40$^{th}$ step). The teaching sequence in this stage is the collection of two speed vectors, i.e. the components of the estimated speed vector and the components of the speed vector obtained from the numerical filter.

3. Immediately on the teaching sequence having been gathered, the filter goes over to the next stage of work, which is the teaching of the network. The network receives from the teacher a collected teaching sequence as many times as the (delta) error is smaller than the assumed one, i.e. $\Delta$ = 0.001. The initial moment of teaching is the most essential in the working of the filter, as the initial weights among particular neurons are chosen at random, which has

decisively influences the final effect of the teaching process. In the model constructed teaching the net with a teacher was applied.

The selection of the kind of teacher is the basic problem. The numerical filter, which is assumed to fulfil this function, has itself big errors in the initial estimation stage. For lack of a better pattern, it was necessary to "supplement" the education of the network while the experiment lasts. This is a kind of feedback with a continual exchange of the teaching sequence with new, more precise elements. Thus, the network education is supplemented in a cyclic way.

A variant was also examined, where the teaching sequence is exchanged with network outputs. This variant was called network self-learning.

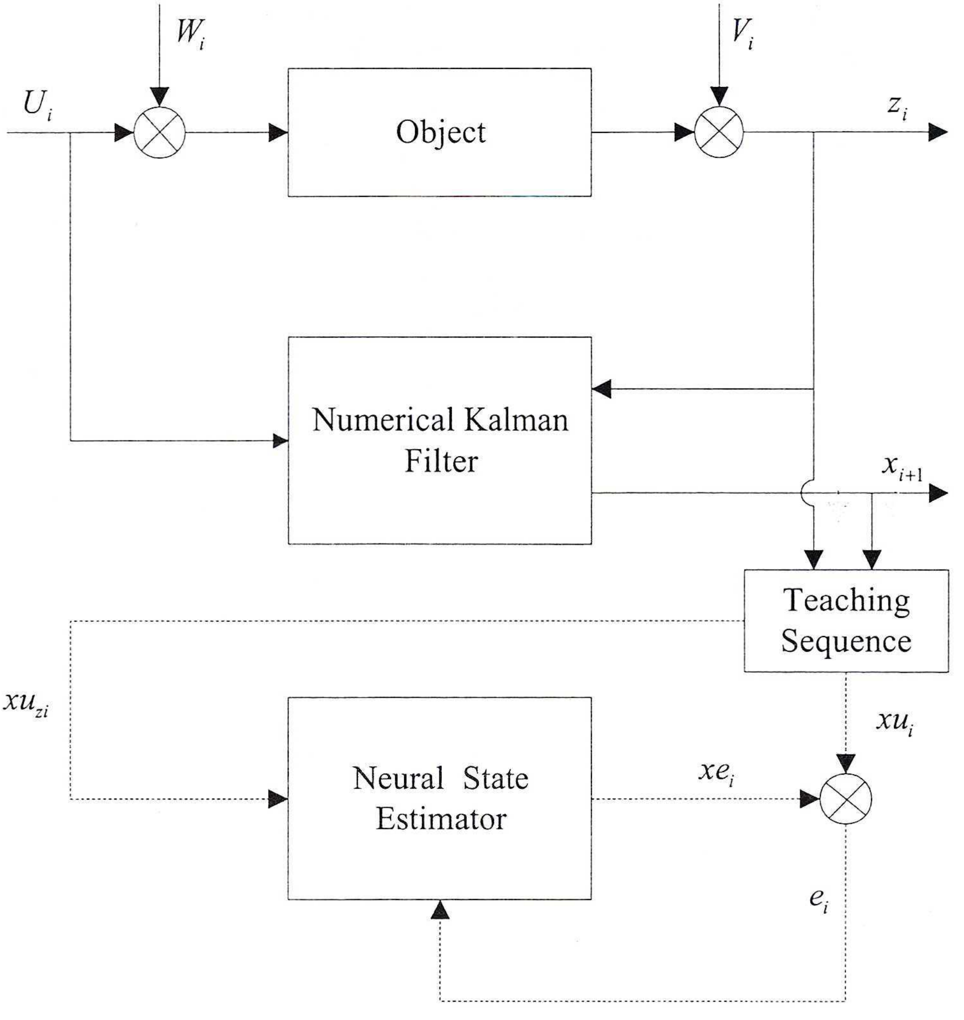


Fig. 3. Diagram of network teaching

The intermediate estimation method [2], which is a combination of Kalman filter and the generalized method of least squares showed the best convergence of the estimation process. This method became the "teacher" of the neural network.

4. The next period of estimator work is the so-called examination, which commences immediately after the network has been taught. The numerical part continues working, and the neural network output is given the values of the estimation vector components. At the network output we obtain state vector values filtered off by the network. The diagram below presents the process of this stage of filter work.

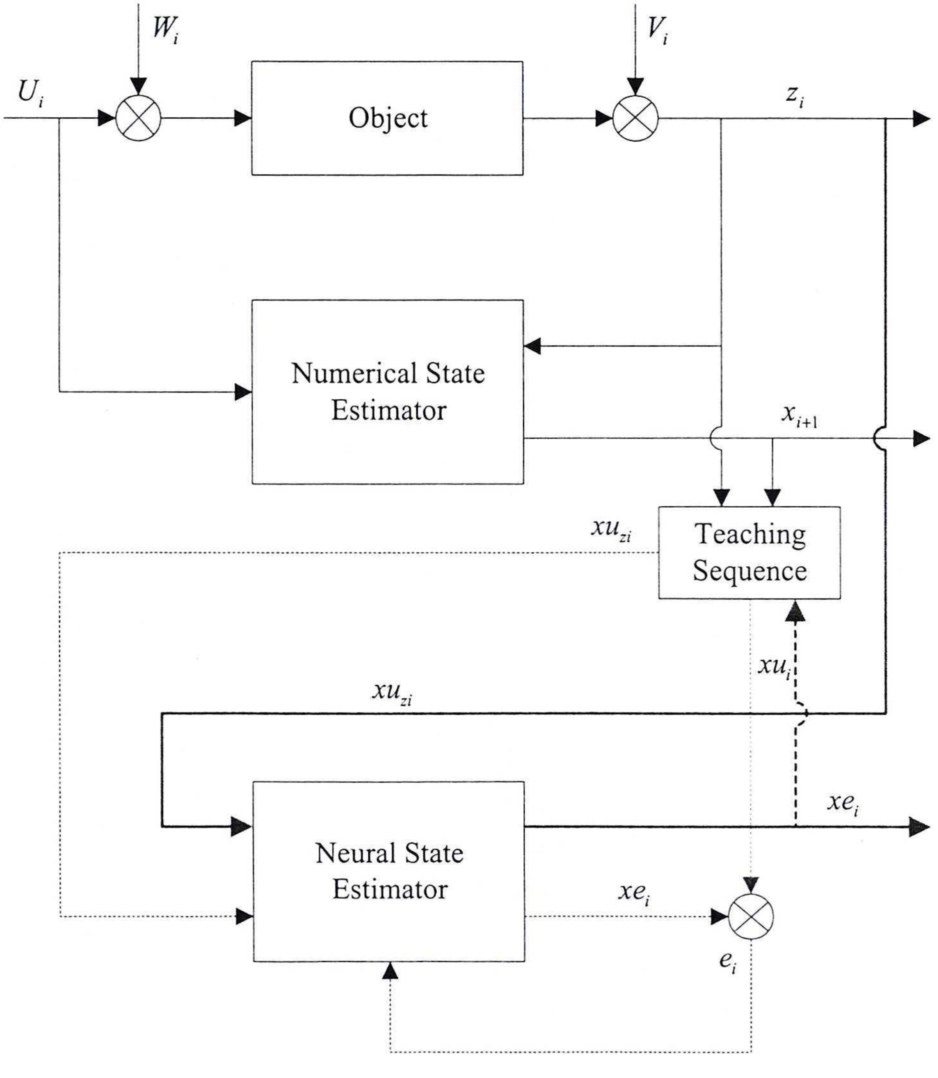


Fig. 4. A stage of network work, so-called examination

The algorithm of determining the state vector based on a neural state estimator has been presented on the diagram below.

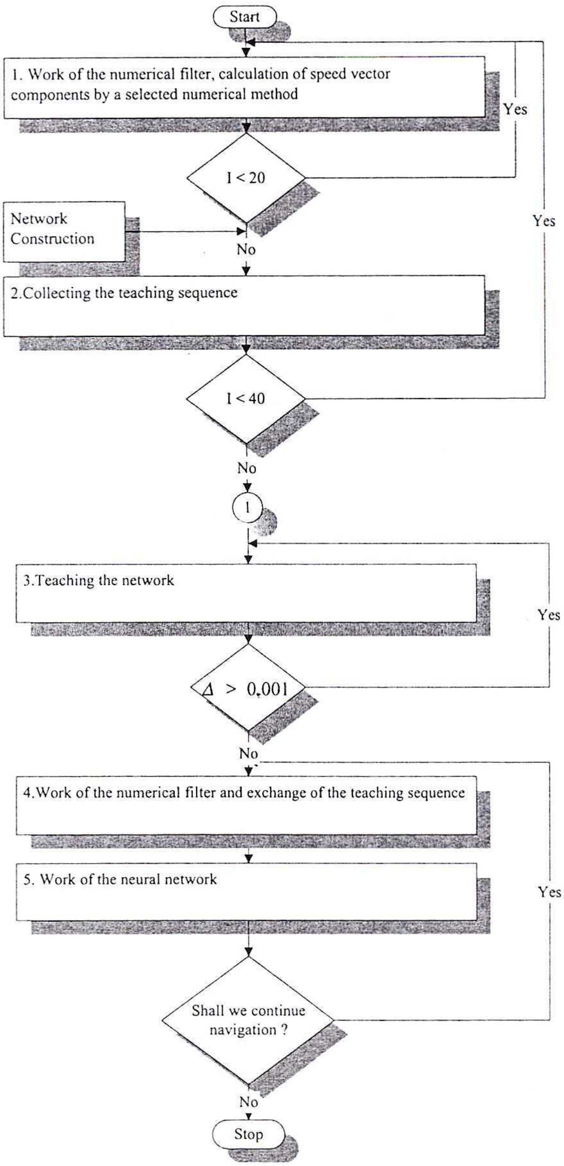


Fig. 5. Working algorithm of a neural state estimator

What comes below presents examples of neural filter estimation and its teacher, the numerical filter (intermediate estimation method) and process of estimation without filtering.

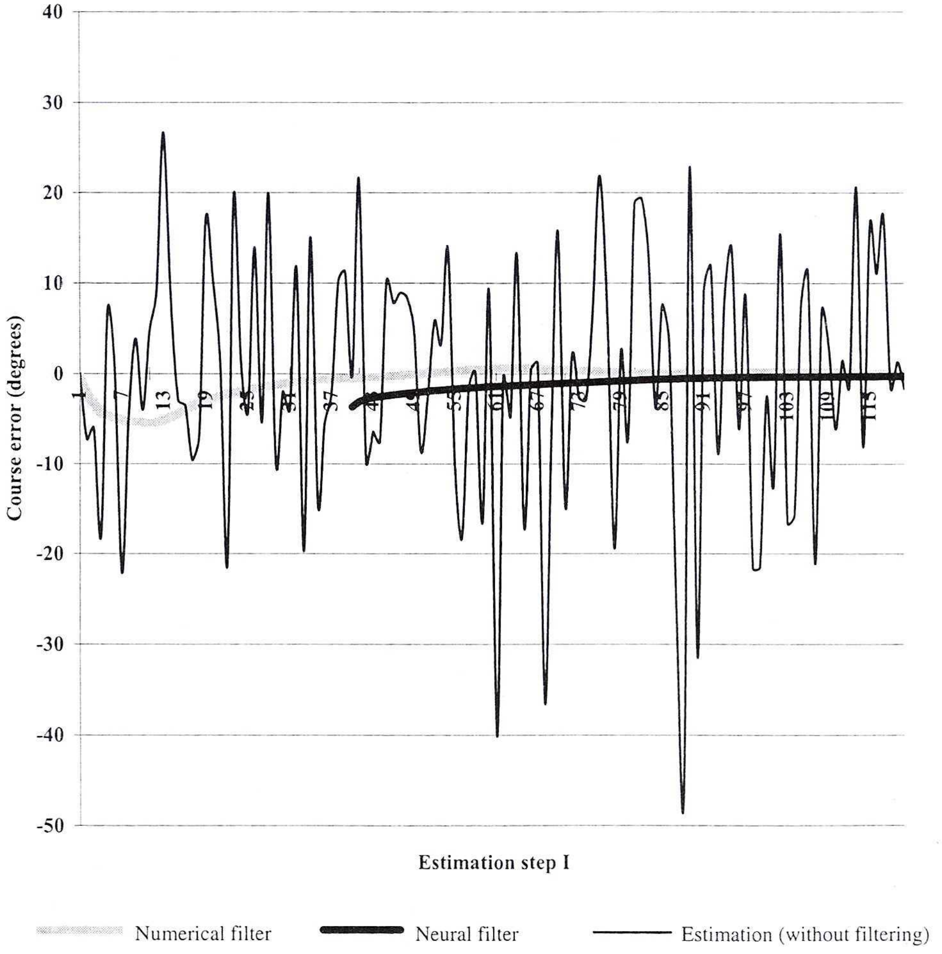


Fig. 6. Examples of estimation results

## 5. *Final remarks*

The treatise presents the neural realisation of the Kalman filter. Numerous numerical experiments carried out have provided no basis for rejecting the hypothesis of the correctness of the constructed neural mathematical model of state estimation. In the scope of numerical methods, decidedly the best results have been obtained by applying the intermediate estimation method [2]. In most cases the neural filter gave better results than the numerical filter. A multilayer perceptron has been applied for network construction, consisting of the input layer, one hidden layer (100 neurons) and the output layer.

The experiments carried out gave reasons for suppositions that in real conditions a hybrid adaptive system would work best, where, depending on the characteristics of the process the most suitable method of state estimation would be selected (numerical or neural) and the network "teacher" (supplementary teaching with a numerical filter or the self-education of the network).

REFERENCES

[1] Balicki J., Stateczny A.: *The Application of Artificial Neural Networks in Anti-Collision Systems*. Naval Academy Scientific Papers, No 4, Gdynia 1995, 5-22.

[2] Kantak T., Stateczny A., Urbański J.: *Basics of Navigational Automation*. A Naval Academy Manual, Gdynia 1988.

[3] Korbicz J., Obuchowicz A., Uciński D.: *Artificial Neural Networks*. Basics and Applications. AOW, Warszawa 1994.

*Received 26 April, 2000*
*Accepted 24 October, 2000*

*Andrzej Stateczny*

**Neuronowa estymacja stanu w nawigacji morskiej**

S t r e s z c z e n i e

W artykule przedstawiono realizację neuronowego estymatora stanu dla pomiarów nawigacyjnych. Rozpatruje się jedno z podstawowych zagadnień nawigacji morskiej jakim jest statyczne opracowanie pomiarów nawigacyjnych w celu wyznaczenia estymowanego wektora ruchu i pozycji okrętu.

*Анджей Статечны*

**Нейронная оценка состояния в морской навигации**

Р е з ю м е

В статье представлена реализация нейронной оценки состояния навигационных измерений. Рассмотрен один из основных вопросов морской навигации, каким является статическая обработка навигационных измерений с целю определения оцениваемого вектора движения и место судна.