NEURAL NETWORKS ARCHITECTURES FOR MODELING AND SIMULATION OF THE ECONOMY SYSTEM DYNAMICS

NICOLAE TUDOROIU

Spiru Haret University, Romania
Email: tnicolae@excite.com

CLAUDIU CHIRU

Spiru Haret University, Romania
Email: claudiu_chiru@yahoo.com

MANUELA GRIGORE

Spiru Haret University, Romania
Email: grgrmanuela@yahoo.com

Abstract:

This research work investigates the possibility to apply several neural network architectures for simulation and prediction of the dynamic behavior of the complex economic processes. Therefore we will explore different neural networks architectures to build several neural models of the complex dynamic economy system. In future work we will use these architectures to be trained by well-known training algorithms, such as Levenberg-Marquardt back-propagation error and Radial Basic Function (RBF), to compare their results and to decide at the end, which one is the best among the different applications from the economy field. The results presented in this work are based on the experience accumulated by the authors in the field of identification, modeling and control of the industrial and economic processes, namely chemical, HVAC, automotive industry, and satellites constellation. The neural networks are strongly recommended for the highly nonlinear processes for which an analytic description is almost impossible. It is well known that the single-index economic models and the selection of leading indicator variables are normally based on linear regression methods. Moreover, in statistical modeling of the business cycle, it has been well established that cycles are asymmetric; therefore it is doubtful that linear models can adequately describe them. With recent developments in nonlinear time series analysis, several authors have begun to examine the forecasting properties of nonlinear models in economics. Probably the largest share of economic applications of nonlinear models can be found in the field of prediction of time series in *capital markets. Furthermore recently, the neural network architectures use financial variables to forecast industrial production by estimating a nonlinear, non-parametric nearest-neighbor regression model, and are very useful for fault detection, diagnosis and isolation (FDDI) of the modes faults in the control systems. The simulation results reveal a high capability of the neural networks to capture more accurate the nonlinear dynamics behavior of the process and to yield high performance, comparable to the Kalman filters techniques and all other control strategies developed in literature. The nonlinear mapping and self-learning abilities of neural networks have been motivating factors for development of intelligent control strategies. The neural networks approach is very interesting because don't need the linear model of the process that means time consuming and increasing the risk to reduce the accuracy in capturing the appropriate dynamics of the process.*

***Key words**: Dynamic systems, Kalman filters, Neural networks architectures, ARMA models, Estimation, Neural-models, Neural-control strategy, Inverse Neural-control strategy, MIMO control strategies.*

1. INTRODUCTION

Macroeconomic growth simulation has great importance in economic developments and policies, but there have not been any effective methods of simulation. For this purpose could be used several architectures of neural network (NN) to construct the neural models simulating the complex dynamics of the overall economy system, in general, or particularly the relations of GDP (Growth Domestic Product), financial revenue, work force and price on industry structures of the economy.

However, detecting trends and patterns in financial time-series has been of great interest to the finance world for decades. So far, the primary means of detecting trends and patterns has involved statistical methods such as statistical clustering and regression analysis and more recently the Autoregressive Conditional Heteroscedastic (a condition where a stochastic process has non-constant second moments) (ARCH) model, a category of conditionally heteroskedastic stochastic processes, and Generalized ARCH (GARCH) model, which are considered today the most often applied time-varying models (Dias, F.C., 1994; Burns, A.F., and Mitchell, W., 1946; Fritsch, U., and Stephan, S., 2000; Hymans, S., 1973; Timotej Jagric, 2009). The mathematical models associated with these methods for economical forecasting, however, are linear and may fail to forecast the turning points in economic cycles, because in many cases the data set for the model may be highly nonlinear. Time series analysis is the fitting of stochastic processes to time series. Any associative array of times and numbers can be viewed as a time series. In most cases the time might be of a regular interval length. Energy traders will often attempt to forecast power consumption based upon both weather normal and short term weather forecasts. The purpose of time series analysis is to study the more interesting case in which terms corresponding to different points in time have interdependencies (Dias, F.C., 1994; Burns, A.F., and Mitchell, W., 1946; Fritsch, U., and Stephan, S., 2000; Hymans, S., 1973; Timotej Jagric, 2009).

Nowadays the stock market prediction is an area of financial forecasting that attracts the attention of the worldwide scientists from economic field. In financial theory, the efficient market hypothesis (EMH), in its weak form, predicts that analysis of time series data alone will provide no excess return over a simple buy and hold strategy and the data contained in the time-series has no economic value unless the data leads to a transaction. However, it does not deny that such prediction is possible from inside information. Predictive success with neural networks and univariate time series would be contrary to this form of the EMH (Timotej Jagric, 2009). Research on using neural networks has been carried out to retrieve trends and patterns of stock markets. Application of neural networks in time series forecasting is based on the ability of neural networks to approximate, in real-time, nonlinear functions very quickly, the both off-line and on-line modes, if they are implemented correctly.

The literature on time series analysis covers numerous standard models for stationary processes. The simplest of these are white noise processes. From white noise processes can be constructed the well known moving average (MA), autoregressive (AR) and autoregressive-moving-average (ARMA) models of the economic processes, which are generally used to model conditionally homoskedastic auto-correlated processes (Dias, F.C., 1994; Burns, A.F., and Mitchell, W., 1946; Fritsch, U., and Stephan, S., 2000; Hymans, S., 1973; Timotej Jagric, 2009; Tudoroiu, N., Yurkevich, V., Khorasani,K., 2003; Tudoroiu, N., Khorasani, K., Yurkevich, V., 2003). Other processes are used to model conditionally heteroskedastic processes. Techniques for fitting these processes to actual time series tend to be specific to the particular models.

Noticeably, the economic processes of both the national and the global economies could be multivariable (Multi Input-Multi Output, MIMO) (Tudoroiu N., Khorasani K., Patel R.V., May 2000; Tudoroiu N., Khorasani K., Patel R. V., March 2000; Tudoroiu N., Zaheeruddin M., 2002; Tudoroiu N., Zaheeruddin M., 2003) and nonlinear, with variable gain and variable time delays. It is very well known that in industry to control such processes, self-adaptive, model-based artificial neural networks (ANN), is very complex task, using multi-variable control strategies, such as Proportional Integral Derivative (PID) (Tudoroiu N., Zaheeruddin M., 2003), state feedback control law, sliding mode control, etc. Self-teaching ANN algorithms (Bishop, C.M., 1995; Tudoroiu, N., Manuela Grigore, Jeflea, V., Roxana-Elena Tudoroiu, 2008; Zaheer-uddin, M., and Tudoroiu, N., 2004), are trained off-line on past history of the economic process (the measured data set of the last decades of operation of the process). In our research we will consider a realistic approach based on the fact that any economy is a multivariable dynamic economic process. The block scheme of the overall economy system is presented in *Figure 1* (Béla Lipták, 2005). The controlled variables (the set points) of this process could be the GDP and the leading economic indicators (LEIs) of the economy. Different nation's economies might distribute the resulting production differently among the population, but the manipulated and disturbance variables in all economic processes are basically the same. The only difference is that inside of the free societies, most of the variables are allowed to float freely, while in totalitarian societies some of the variables are arbitrarily constrained. The goals of both economies (the set points in the block scheme shown in *figure 1*) are to keep their Leading economic indicators, LEIs, and their GDP reference variable at some desirable values or to keep raising these targets (Béla Lipták, 2005). LEIs could be several economic indicators, namely the indicators of the standing of the stock and housing markets, percent unemployment, percent inflation, increase in wages, quality of health, education, social security services, etc (Béla Lipták, 2005). As well, LEIs also consider the value of the currency and the price to earnings ratio of securities. Several LEIs also consist of the housing "bubble" (market value of the real estate in units of years of rental income) (Béla Lipták, 2005). As shown in the block scheme presented in *figure 1*, the relationship between manipulated variables, for instance the interest rate, taxation, trade or energy policy and controlled variables (GDP or LEIs) are functions of the "gains" at the nodes in the "hidden" layers of this self-learning ANN model of the economic process (Béla Lipták, 2005). The neural controller looks at the discrepancy between the desired (set point value, Y_{sp}) and actual controlled variable values (Y) of the GDP or LEIs. Moreover if an error exists the neural controller adjusts the manipulated variable at the desired value. Also the neural controller uses the inverse of the model of the economy to constantly update it (Tudoroiu, N., Khorasani, K., 2007; Tudoroiu N., Zaheeruddin M., 2002; Béla Lipták, 2005). It does that by comparing its predicted output with that of the actual economy and, if a difference exists, the neural model will be corrected. Also, both the manipulated variable and the disturbance variables have a tight effect on the real process, and therefore they could be considered inputs to the neural model architecture. Before one can use an neural model of an economic process as a feed-forward predictor, it must be trained on past experimental data set of the economic process performance. In case of the economy, the model can be trained on the experimental data set of the past decades, just as it would be trained on the historical performance data of a distillation column (Béla Lipták, 2005). From the neural model's viewpoint it makes little difference between industrial processes or economic process, i.e., for example, if the energy source to a process is the steam supply to a rebuilder is similar if the money supply of the economy (Béla Lipták, 2005). This is a reason of this research to use these similarities of the industrial processes and to adapt all the concepts and the control strategies to the economic processes. Therefore the industrial process could be whenever replaced by an economic process with the same dynamics.

In both processes, there is a "gain" relationship between the input and the output, the change in the steam (or money) flow and the resulting increase in production of distillate (or GDP) (Béla Lipták, 2005). Naturally, the response to the flow of steam (or money) is not instantaneous, but is determined by the time constants and delay times of the processes. In addition, all measurement signals contain some noise. The filter in the simplified feedback closed loop block scheme serves to remove noise. In case of the process of economy, the filter might serve to remove the effects of the arbitrary acts of fund managers, politics (Béla

Lipták, 2005). In this research we will explore different approaches to derive the relations among the input nodes and output nodes in the neural models of the Economy system and Economy neural controller, based on the statistical method (Béla Lipták, 2005).

Furthermore, we *investigate* how to use the statistical method to explain the simulation results and the effectiveness of the simulation. Several algorithms to train the multilayer feed-forward neural network models, among the back-propagation, Levenberg-Marquardt (LM), Radial Basic Function Neural Network (RBF) algorithms will be considered. Overall this research will provide a new perspective of economic growth.

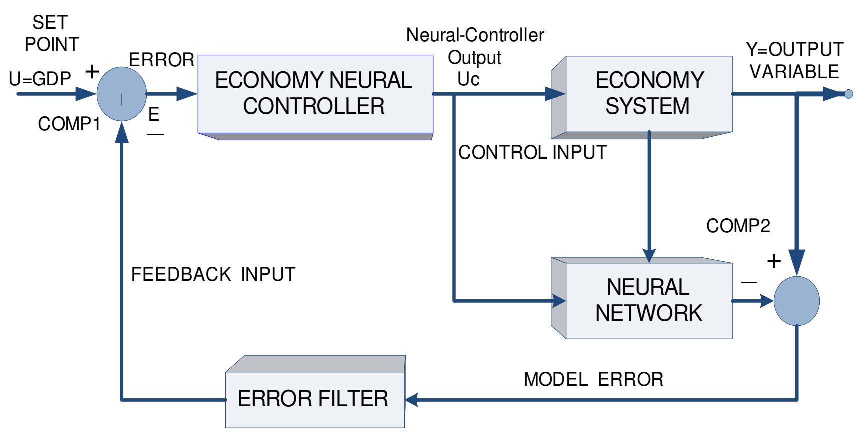

Figure 1: *The dynamic complex economy system-Block scheme*

Neural networks the and genetic algorithms represent today a big challenge in Computational Intelligence and capture the attention from analyst and quant's of trends and patterns. In particular, the neural networks architectures are used extensively for modeling and simulation of financial forecasting with stock markets, foreign exchange trading, commodity future trading and bond yields.

To overcome the deficiencies of the linear economic processes modeling, in the recent economic research are adopted neural networks models to forecast business cycles. The decision to focus on neural networks arises directly from the features of these models. First, neural networks are data-driven, can "learn" from experience during the training process, and are well adapted to, underlying relationships. This property makes the neural networks an ideal modeling method whenever exists a little prior knowledge about the appropriate functional representation of the input-output economic process relationship. Second, once the neural network architectures are suitably specified they are capable to approximate every functional representation to any given degree of accuracy. Finally, neural networks are nonlinear representations and predict, with high accuracy degree, the future value of the reference series, and consequently they are very useful to model a broad variety of macroeconomic time series.

The classical economic process model of leading indicators is capable only to provide a sign for a turning point in aggregate economic activity, and is almost impossible to exactly define the turning point occurrence, or how strong could be the following contraction or expansion. To avoid these deficiencies, a reliable composite index of leading indicators should possess the following properties (Timotej Jagric, 2009):

a) similar movements in the index to those in the business cycle reference series
b) strong statistically relationship between the reference series and the indicator,
c) stability over time of the forecasting performance

According to these requirements we could develop a multivariate neural network forecasting model. Also it is worth to notice that these neural models could be adapted easily for monthly forecasts and could be tested on data for a small, open, transition economy, giving the opportunity for the developer to test the properties of the models under extreme conditions, namely short time period coverage, deep transformation depression of the economy, and market-oriented economy. In the process of the major economy restructuring, in time series wild swings could occur, and it may have many strong impacts on investment and consumption behavior. The development of a broad database to cover all crucial fields of economic activity it is most important step to build the neural model, during the process modeling. For example, the database used in (Timotej Jagric, 2009) includes 365 time series classified into the categories, among main of these industrial production, construction, trade, tourism, transport, exports, imports, balance of payments, employment, wages, unemployment, labor costs and productivity, bank claims, bank liabilities, interest rates, exchange rates, international liquidity, government expenditures, prices, consumption and foreign activity indicators, etc. In the final version of the database, all these time series were transformed into growth rates. Overall the purpose of the model is to forecast a reference variable that is selected to indicate fluctuations in economic activity. The proposed neural models of the economy system could be easily adopted by many countries with the similarities in their economy. Also in our research we will try to apply this modeling strategy for the Romanian economy. It is necessary that the exogenous variable, Y, to be a monthly reported variable, and must measure the real sector of the economy. In (Timotej Jagric, 2009) are given the following two alternative strategies for obtaining a time series of current monthly business activity:

a) adopting a single series as the variable of interest

b) using a function of several variables.

Both approaches have long traditions in empirical macroeconomics (Hymans, S., 1973), namely the monthly money-income relationship focused on the predictability of monthly industrial production or constructed a reference series by averaging several different major aggregate time series, used later to date their reference cycles.

Considering the economy system such as MIMO system, than the construction of leading economic indicators (normally the composite index includes between five and twelve leading indicators), that cover different fields of economic activity and able to include a large amount of information about the economy in the model, requires a monthly and up-to-date series. Therefore, could be selected a monthly index of total industrial production, that has the same cyclical characteristics as GDP (the input U in the economy system) from many countries. To construct several forecasting neural models, first it should be selected the input variables (U), using the extension of criteria employed by NBER (Burns, A.F., 1946), by adding several elements, similar to the approach presented in (Dias, F.C., 1994; Burns, A.F., 1946; Fritsch, U., and Stephan, S., 2000; Timotej Jagric, 2009) in the scoring system. The scoring of each series helps the developer of the neural models to make as explicit as possible the criteria for selecting the input variables, U, and to provide information to evaluate their impact on the system output, Y. The scoring system assures that the selected leading indicators possess the best characteristics among all time series in the database.

In (Dias, F.C., 1994; Burns, A.F., 1946; Fritsch, U., and Stephan, S., 2000; Timotej Jagric, 2009) the scoring system includes the following five major elements:

a) economic significance
b) statistical adequacy
c) promptness of publication
d) smoothness and conformity
e) timing

Economic significance is high scored when the series that succeed in measuring a variable has an important role in the analysis of business cycle movements. First the entire variables that represent a strategic process more largely are rated higher than those more scarcely defined. These largely defined variables are also less likely to shift as a result of different factors such as technological developments, changing consumer tastes or other similar. Second statistical adequacy is the requirement that the selected variable's series should continue to measure the same economic process during future business cycle fluctuations (stationarity, coverage of time unit, measure of revisions, compatibility throughout the period), when the selected indicators are put to the hard test of current usage (Timotej Jagric, 2009). Series that are released quickly are highly scored than those that delay in publication. Third, according to the smoothness criterion, smoother series get higher ratings compared to one which is asymmetrical. When the developer is using only monthly series, suitably for smoothness measure will be the MCD (months of cyclical dominance) value. Finally the conformity of an indicator to past business cycles and timing of its turning points relative to those in aggregate economic activity are crucial merits in this indicator.

To find series that have information about the cyclical component of the reference series, additional econometric procedures are needed. More recent work has focused on the second moment of the joint distribution of the series of interest, summarizing the cyclical timing by estimating phases in the frequency domain at business cycle frequencies (Hymans, S., 1973). The stability test of the results could be performed on both original and seasonally adjusted series, and in the final forecasting model, only original time series will be used.

Recently the neural networks represent a big challenge to be used extensively for financial forecasting with stock markets, foreign exchange trading, and commodity future trading and bond yields.

Stock market prediction is an area of financial forecasting which attracts a huge attention of the scientists from the economic community. It is worth to mention that in financial theory field, the efficient market hypothesis (EMH), in its weak form, predicts that analysis of time series data alone will provide no excess return over a simple buy and hold strategy, and the data contained inside of the time-series have no economic value except the data lead to a transaction. On the other hand, it does not deny that such prediction is possible from inside information. Predictive success with neural networks would be contrary to this form of the EMH. Almost in every stock market prediction that use neural networks models has been carried out to retrieve trends and patterns of stock markets. Application of neural networks in time series forecasting is based on the ability of neural networks to approximate nonlinear functions very quickly, possibility in real-time, if they are implemented correctly.

2. NEURAL NETWORKS ARCHITECTURES

2.1. NEURAL NETWORKS DESCRIPTION

The Neural networks, called also Artificial Neural Networks (ANN), were introduced by McCulloch and Pitts in 1943, and complex dynamical systems by Forrester in the 1950s. The ANN is capable of learning the relationships between inputs (manipulated and disturbance variables) and outputs of the economic process based on its past history. The early work on neural networks considered binary nodes only: each neuron had only two states, on and off. By the 1980s, under the influence of new information from studies on biological neurons, analog neurons came to dominate the field. These neural networks usually involve very simple dynamical schemes as nodes, and very complex networks of connections, an approach known as connectionism. Recent work in computer science has shown that these networks are more capable than digital computers. Once the neural model of the economic process is build, off-line trained, it can be continuously on-line updated, in real-time, by minimizing the difference between its predicted output value, *Y*, and its target value $\hat{Y}$. Their architectures are presented in *figure 2-3* and represent computational models that consist of a number of simple processing units (neurons), placed in layers (input, hidden, output) that communicate by sending signals to each other over a large number of weighted connections. The simplified network diagram shown in *Figure 2* is a full-connected, four layers (input, output and two hidden) feed-forward, perceptron neural network. Fully connected means that the output from each input and hidden neuron is distributed to all of the neurons in the following layer and feed-forward means that the values only move from input to hidden to output layers; no values are fed back to earlier layers as a recurrent network does (Bishop, C.M., 1995; Timotej Jagric, 2009; Tudoroiu, N., Patel, R.V., Khorasani, K., 2006; Tudoroiu N., Khorasani K., Patel R.V., 2000; Tudoroiu, N., Manuela Grigore, Jeflea, V., Roxana-Elena Tudoroiu, 2008; Zaheer-uddin, M., and Tudoroiu, N., 2004; Zaheer-uddin, M., and Tudoroiu, N., sept 2004). The detailed full-connected, feed-forward, multilayer perceptron network diagram is shown in *Figure 3*.

All neural networks have an input layer and an output layer, but the number of hidden layers may vary such in *figures 2-3*. When there is more than one hidden layer, the output from one hidden layer is fed into the next hidden layer and separate weights (matrix W) are applied to the sum going into each layer.

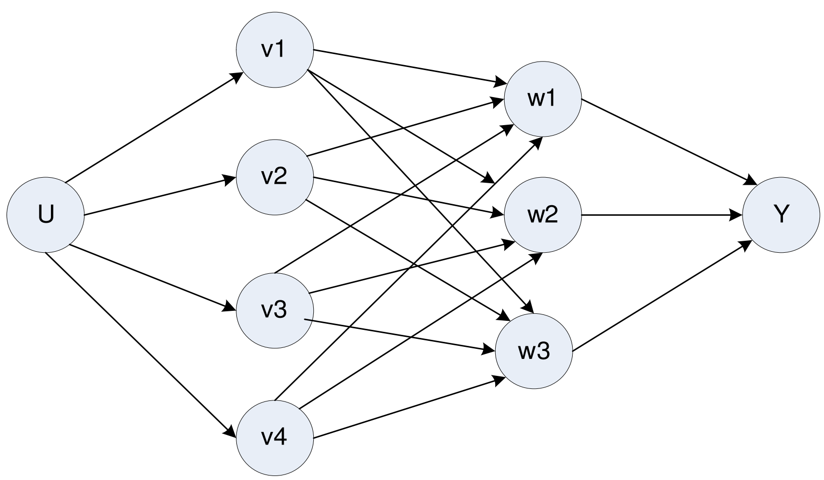

***Figure 2**: 1-4-3-1 neurons simplified multilayer perceptron architecture*

The processing units transport incoming information on their outgoing connections to other units. The signal information is simulated with specific values stored in those weights w_{ij} that make these networks have the capacity to learn, memorize, and create relationships among experimental input-output data set. Each unit j can have one or more inputs, but only one output. An input to a unit is either the data from outside of the network, or the output of another unit, or its own output. The total input to unit j is simply the weighted sum of the separate outputs from the connected units plus a threshold or bias term $(\theta_1, \theta_2, ..., \theta_n)$. Most units in neural network transform their net inputs by using a scalar-to-scalar function called an activation function σ (sigmoid function), given by:

$$\sigma = \frac{e^{\lambda x}}{1 + e^{\lambda x}}$$

where λ represents the learning rate of the neural network scheme.

Many natural processes and complex system learning curves display a history dependent progression from small values of the learning rate λ that accelerates and approaches a peak over time. For lack of complex descriptions a sigmoid function is often used. A sigmoid curve is produced by a mathematical function having an "S" shape, represented in *Figure 4*.

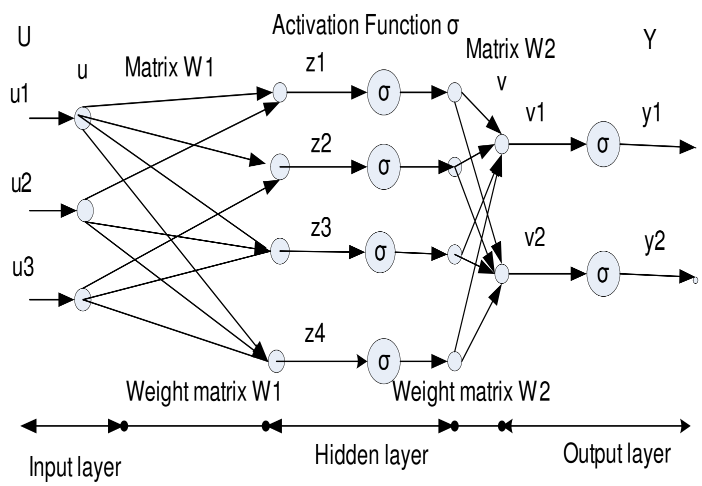

***Figure 3**: 3-4-2 neurons detailed multilayer perceptron architecture*

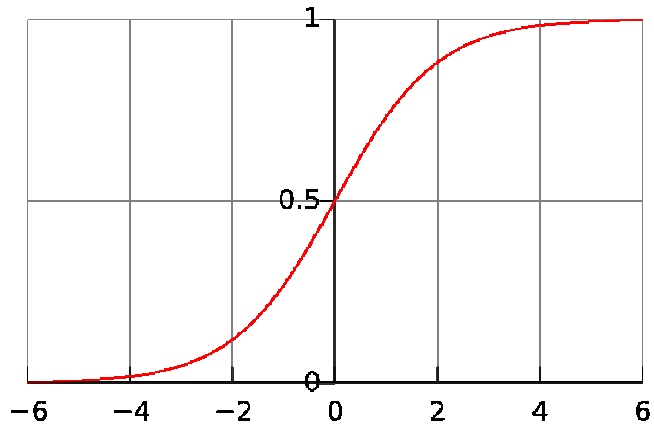

Figure.4: *The graph of the sigmoid activation function*

Recently, empirical economic models derived from neural networks have been shown to offer advantages in both accuracy and robustness over more traditional statistical approaches (regression methods).

Furthermore, these neural-models are used to develop other advanced structures and algorithms, such as adaptive neural-controllers and exponentially weighted moving average (EWMA) neural-controllers. These neural-controllers are then integrated in the feedback control structures of the dynamical complex economic processes to maintain process targets over extended periods, such as those represented in *figures 6-10*. Proper choice of neural-controller parameters (weights) is critical to the performance of these systems. The neural-controllers eliminate the need for an experienced people to tune the system parameters and can be more easily applied to control the dynamical complex economic processes.

2.2. THE NEURAL NETWORKS LEARNING SCHEMES

BACK-PROPAGATION (BP) TRAINING SCHEME

Our goal in this research is to determine an appropriate neural-model for the highly complex dynamical economic process. The experimental input-output data set will be used to train feed-forward neural networks using an error back-propagation algorithm. We will focus our attention on matching model predictions with measurements for network learning and generalization. For this purpose we investigate the simulation with neural networks consisting of three layers (input, hidden, output) or more, configured in different architectures trained by the Levenberg-Marquardt back-propagation error algorithm (Bishop, C.M., 1995; Timotej Jagric, 2009; Tudoroiu, N., Patel, R.V., Khorasani, K., 2006; Tudoroiu N., Khorasani K., Patel R.V., 2000; Tudoroiu, N., Manuela Grigore, Jeflea, V., Roxana-Elena Tudoroiu, 2008; Zaheer-uddin, M., and Tudoroiu, N., 2004; Zaheer-uddin, M., and Tudoroiu, N., sept 2004).

Even though a simple steepest descent gradient algorithm can be efficient, there are situations when moving the weights within a simple learning step along the negative gradient vector by a fixed proportion will yield a minor reduction of error. For flat error surfaces for instance, too many steps may be required to compensate for small gradient values. Furthermore, the error contours may not be circular and the gradient vector may not point toward the minimum. To avoid these situations one may we replace the gradient descent method by the Gauss-Newton optimization method, which uses the second derivative of the error function *E,* namely its Hessian matrix $H\ (w)\ =\ \Delta E\ = \nabla(\nabla_w E)$. To update the weights, a recursive Gauss-Newton optimization algorithm may be expressed in the following matrix form:

$$W^i = W^{i-1} - \gamma [H(w^{i-1})]^{-1} (\nabla_W E)$$

Because the Hessian matrix may be singular, it can be made invertible by using the Levenberg-Marquardt relaxation as follows:

$$\hat{H}(w^i) = H(w^i) + \mu I_n$$

where μ is a relaxation parameter and I_n is an identity matrix. The Levenberg-Marquardt algorithm is preferred for a small number of weights because the computation speed of the inverse Hessian matrix decreases when the number of the weights increases. Otherwise, the steepest descent optimization algorithm is preferred.

The number of hidden neurons and layers are varied to provide optimal network performance. The development of an optimal neural network structure is complicated by the fact that back-propagation networks contain several adjustable parameters for which the optimal values are initially unknown. These include structural parameters (such as the number of hidden layer neurons, initial weights and biases) as well as learning parameters (such as the learning rate, momentum, and error goal). The learning rate determines the speed of convergence by regulating the step size. However, the network may settle far away from the global minimum of the error surface if the learning rate is too large. On the other hand, smaller rates can ensure stability of the network by diminishing the gradient of noise in the weights, but result in longer training times. For this reason the algorithm could be improved by introducing an adaptive learning scheme which decreases considerably the training time.

A smaller training tolerance usually increases learning accuracy, but can also result in less generalization capability as well as longer training time. Conversely, a larger tolerance enhances convergence speed at the expense of accuracy in learning. It is shown in the literature that a single hidden layer is sufficient for learning any function, but the number of hidden neurons can grow without a bound. This of course, may result in a network with a large number of connections which defeats the main purpose of having an accurate prediction. By increasing the number of hidden layers, each consisting of sigmoid nodes, the complexity of the network can increase more rapidly than the number of connections. The optimum network architecture should have a minimum number of connections and produces a low cross-validation error. Development of neural network models typically consists of considerable training and testing. The objective is to find a network that will perform well on the test data. For the training set it is recommended to select the first half of the experimental input-output data set to update the weight matrices and to use the other half as the test set. Network performance is measured by the root mean squared error [RMSE), which is given by:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{i=n}(y_i - \hat{y}_i)^2}$$

where n is the size of the test set, y_i is the measured value of the output, and $\hat{y}$ is the estimated response provided by the neural networks. All these neural network architectures have to be trained on the experimental data set to earn the mapping from inputs to outputs of the process model. However, sometimes a network with small number of hidden neurons could be sufficient to generate this mapping. To avoid over fitting it is recommended to limit the number of neurons to the fewest as possible, as long as the network converges to the desired error level, and cut off the training once that error was met, i.e., for example, such in *figure 5*.

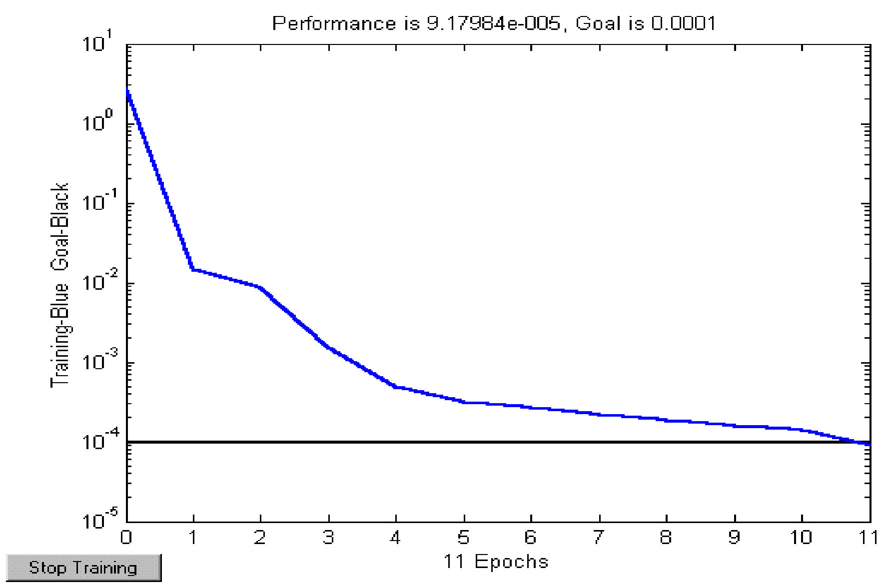

***Figure 5**: Neural Network –Back-propagation training phase*
(Neural network MATLAB Toolbox, version 7.1)

One of the most problems that occur in many applications could be the over-fitting. The error on the training set is driven to a very small value, but when new data are presented to the network, the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. One method for improving network generalization is to use a network that is just large enough to provide an adequate fit. If we use a small enough network, it will not have enough power to over-fit the data. Since we do not know how large a network should be in our application, we will select regularization. This technique of regularization encourages smoother network mappings by adding a penalty to the error function to give.

LEARNING BY LEAST SQUARE APPROXIMATION

Given a complex dynamical system, one may fix initial states at each node, and initial values for any free control parameters, and begin iterating the entire system on successive ticks of a master clock. There results a time series of values for the states at the nodes, which may be compared with experimental data set from an original economy system (Timotej Jagric, 2009). If the comparison is a good approximation, we have a mathematical model and computer simulation for the given data set. Unexpectedly, good approximations may be obtained by developing an automatic process of learning. One such process is the well-known least squares approximation scheme, in which a problem in the calculus of variations is solved by a numerical algorithm to provide the neural controller parameter values to best fit the computed data set to the given one, the experimental data set. This iterative process could be implemented using specific software found in the toolbox of several computational math programming environments, namely, MATLAB, Maple, Mathematica, etc.

2.3. APPLICATION TO THE WORLD ECONOMIC SYSTEM

These approaches, in a particular case, could be used for simulation and prediction of global economic data from national accounts of Romania. We could use quadratic polynomials only for our maps, and linear functions for links, chosen for their richness of the bifurcation diagrams. Applying the both approaches to such complex dynamic economy system, consisting from several sectors, it is possible to get a very large total space of control parameters for this system of high complexity and nonlinearity. Least squares approach to find the best choice of all these parameters for the available time series from national accounts could result in a rather encouraging simulation. However, this solution is computationally intensive, but running the model with fixed values of the controls the recursive process is very fast. Also the rapid evolution of the computer hardware and software, the massively parallel model of the least square approximation scheme and the neural network structure, trained in batch mode by back-propagation algorithm, are feasible for economic systems from a small business to the world economy (Bishop, C.M., 1995; Dias, F.C., 1994; Burns, A.F., and Mitchell, W., 1946; Fritsch, U., and Stephan, S., 2000; Hymans, S., 1973; Timotej Jagric, 2009;

Tudoroiu, N., Manuela Grigore, Jeflea, V., Roxana-Elena Tudoroiu, 2008). What we need is only a good data set (big challenge) and time to process the software and the data set. Since the neural networks structures can perform essentially arbitrary nonlinear functional mappings between sets of variables, a single neural network structure could, in principle, be used to map the rough input data directly onto the target output values. It is worth to mention that in practice, such an approach will generally give poor results. Moreover for many practical applications, the choice of pre-processing could be one of the most significant factors in determining the performance of such complex control system. In many situations we could use the scoring system to select leading indicators for the input variables. All these input variables cover different fields of economic activity and are reported monthly. One of the target forecasted output variable could be, for example, the monthly index of industrial production. The input and target variables of the feedback economy system are not seasonally adjusted because the seasonal adjustment procedure applied to the data may cleanse them of any underlying nonlinearities. A key assumption, implicit used in this approach, is that the statistical properties of the generator of the data are time-independent. Therefore the time series problem will be mapped onto a static function approximation problem to which a feed-forward neural network can be applied. In many cases, almost all of the selected input variables and the target variable show an underlying trend. However, there is no universal trend function that could be applied to all economy system input variables, and so we could select to be used one-year and monthly growth rates. Also it is worth to mention the well known phenomenon specific of the economic processes, called "course of dimensionality". According to this phenomenon if we are forced to work with a limited quantity of experimental data set, such is the case in practice, then the dimensionality of the input variables space can rapidly lead to the point where the data set is very sparse, providing a very poor representation of the mapping. To achieve this goal, it is recommended to use an unsupervised linear transformation technique, such as the principal component analysis (PCA), where a set of data are summarized as a linear combination of an orthonormal (orthogonal and the norm equals to one) set of vectors. The estimated principal components provide a linear approximation that represents the maximum variance of the original experimental data set in a low-dimensional projection. Also these principal components give the best low-dimensional linear representation in the sense that the total sum of squared distances from data points to their projections in the space are minimized. More challenging is how it could be determined the size and topology (design) of the neural network. The design changes could fundamentally alter the forecasts (estimates) produced by the neural network, even when no changes are made to the inputs, outputs, or sample size. Nevertheless, the literature provides various techniques for optimizing the design. It is worth to note that it is important to distinguish between two distinct aspects of the design selection problem. First, we need a systematic procedure to define possible designs. Second, we need several selection criteria of deciding which of the designs considered should be selected. This is determined by the requirement of achieving the best possible generalization (trial and error procedure). The following most important basic design requirements for the neural network architecture are reported:

a) Feed-forward back-propagation networks are mostly used in forecasting applications.

b) Due to the type of input and output variables experimental data set it is useful one combination of pure linear and tan-sigmoid activation functions because they are continuous and easily differentiated.

c) For time-series forecasting the output-layers of the neural networks have only one neuron since they predict the future value of one reference series.

d) The neural network has no more than three layers of neurons (input, hidden and output) since such a network is a universal approximator.

The above requirements greatly reduced the space of possible designs, simplifying significantly the design algorithm. We will start to build several architectures of neural networks in different successive layers with each layer having fewer units than the previous layer. Every time a new layer will be build, a single unit, called the master unit, will be added. Then step-by-step additional units will be added. At every step the neural network will be trained and the forecasting performance will be estimated. The whole process will be repeated until a larger network did not sufficiently contribute to the forecasting performance.

In the process of network design selection, we will test every neural network structure for different forecast horizons, all calculations will be performed by using MATLAB 7.5 (R2007b) software, Neural Network TOOLBOX. We will use the back-propagation algorithm that automatically supervise the testing and record the neural network performance.

After extensive testing of possible structures of neural network designs we will select the best neural network architecture that performs very well.

The layers of a multilayer network play different roles. Multiple-layer networks are quite powerful. For instance, a network of two layers, where the first layer is sigmoid and the second layer is pure linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well. A single layer neural network with linear activation function has the same properties as the linear regression model.

The sensitivity analysis suggested that there are some nonlinear relationships between the reference variable and selected leading indicators. This explained why we were able to improve the forecast performance of the original model.

We underlined one important contribution of neural networks--namely their elegant ability to approximate arbitrary nonlinear functions. This property is valuable in time series processing, especially in the forecasting field. However, it should be noted that nonlinear models are not without problems, both with respect to their requirement for a large database and careful evaluation, and with respect to limitations of learning or estimation algorithms.

2.4. NEURAL NETWORK ARCHITECTURES

All the neural network architectures proposed in this section will be trained on the experimental data set to learn the mapping from inputs to outputs of the economic process model (Tudoroiu N., Khorasani K., Patel R.V., 2000; Tudoroiu N., Zaheeruddin M., 2002; Tudoroiu N., Zaheeruddin M., 2003; Zaheer-uddin, M., and Tudoroiu, N., 2004; Zaheer-uddin, M., and Tudoroiu, N., sept 2004). To avoid over fitting we will limit the number of neurons to the fewest as possible as long as the network converges to the desired error level, and cut off the training once the error was met, such in *figure 5*. Our proposed four neural models structures for the economy system integrated with the neural controller are described in detail below:

a. Nonlinear static model integrated in the first neural control strategy, represented in *figure 6*:

- The neural network objective is to represent a static model of the economic process which is assumed to be expressed as a nonlinear function g:

$$Y(k) = \hat{g}(U_c(k)) \qquad (1)$$

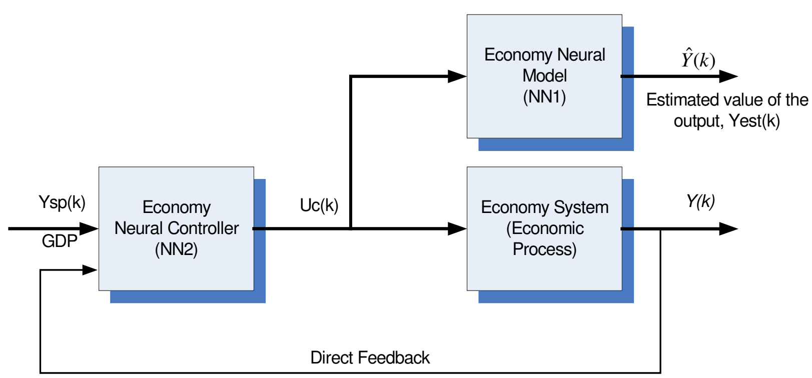

***Figure 6**: The first neural control strategy*

b. Nonlinear first-order model integrated in the second neural control strategy, represented in *figure 7*.

- In this case the delayed economic process (economy system) output $\hat{Y}(k-1)$ is used in addition to the present system input $U_c(k)$ as input variables i.e., the dynamic input-output neural model is assumed to be expressed as:

$$\hat{Y}(k) = g(\hat{Y}(k-1), U_c(k)) \tag{2}$$

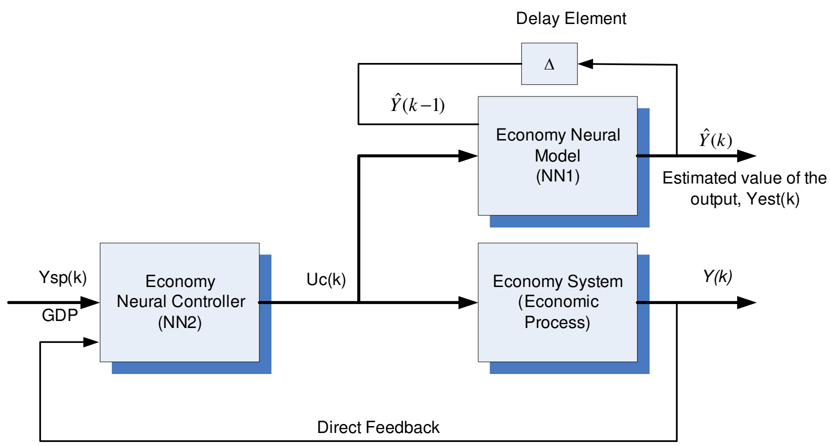

***Figure 7**: The second neural control strategy*

c. Nonlinear first-order model with delayed input vector $U_c(k-1)$ integrated in the fourth neural control strategy, represented in *figure 8*:

- In this case the delayed economy system output $\hat{Y}(k-1)$ and the delayed neural controller output $U_c(k-1)$ are used in addition to the neural controller output $U_c(k)$ as input vectors, i.e.,

$$\hat{Y}(k) = g(\hat{Y}(k-1), U_c(k), U_c(k-1)) \tag{3}$$

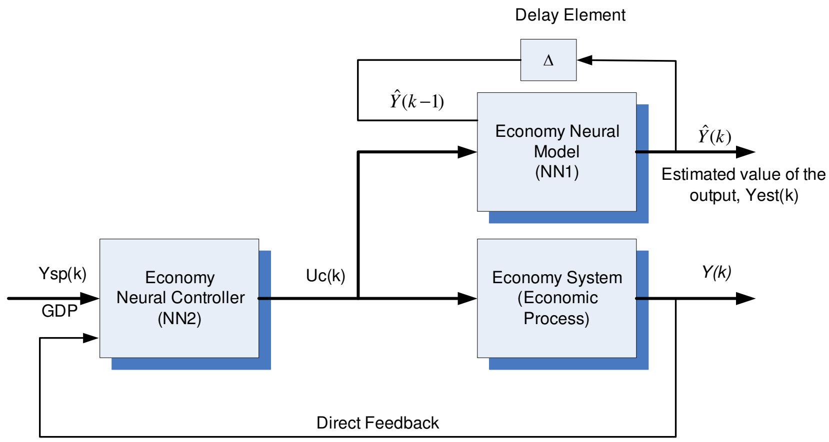

Figure 8*: The third neural control strategy*

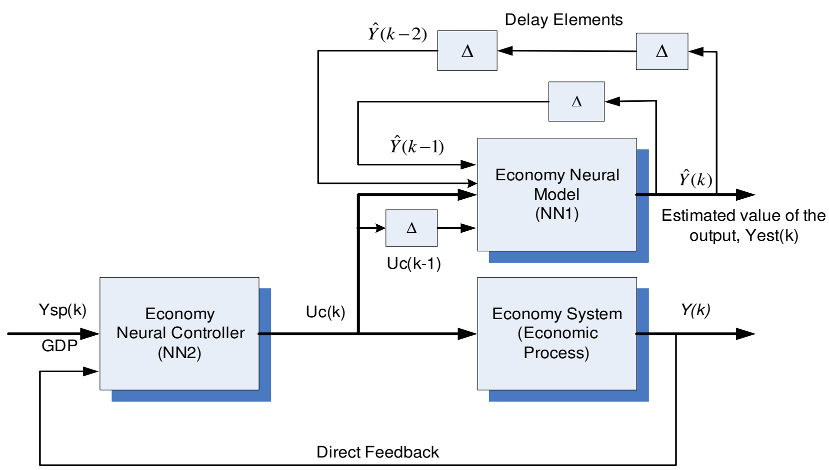

Figure 9*: The fourth neural control strategy*

d. Nonlinear second-order model integrated in the third neural control strategy, represented in *figure 9*:

- In this case the delayed estimated economy system output $\hat{Y}(k-1), \hat{Y}(k-2)$ are used in addition to the economy neural controller output, $U_c(k-1), U_c(k)$, as input vectors, i.e., the dynamic input-output neural model is assumed to be expressed as:

$$\hat{Y}(k) = g(\hat{Y}(k-1), \hat{Y}(k-2), U_c(k-1), U_c(k)) \quad (4)$$

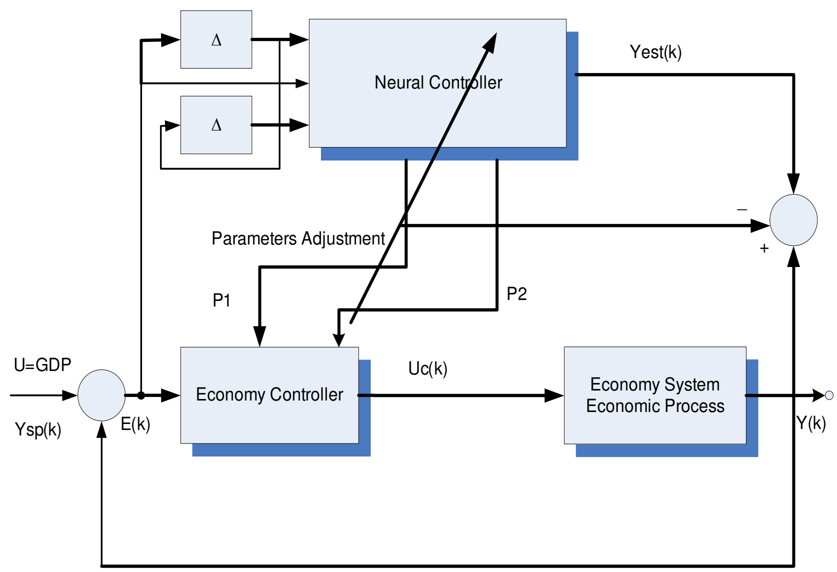

Figure 10*: Neural control strategy with parameters adjustment*

In *figure 10* we represent one of the neural control strategies with parameters adjustment, applied successfully for the industrial applications.

Before training the network weights and biases off-line they are initialized using Nguyen-Widrow (Dias, F.C., 1994; Burns, A.F., and Mitchell, W., 1946; Kim, B., May, G.S., 1993; May, G.S., Huang, J., Spanos, C., 1991), initial conditions (small random values). From the preliminary simulation results in our research, the network parameters could be very well initialized, for example, to the following values: error goal = 0.01, learning rate = 0.02 and momentum = 0.95. The number of epochs to reach the error goal depends on the initial conditions for the weights and biases and the number of hidden neurons. The back-propagation algorithm attempts to minimize the error between the output of the network and the target or desired response in weight space, using the method of gradient descent in conjunction with Levenberg-Marquardt relaxation. Neural network architecture with few hidden neurons is sufficient to generate this mapping and to give an indication about the evolution of the neural controller parameters.

Also we observed that the last two neural models representations yield the best performance due to the presence of their internal feedback and the delayed input signals as input vectors to the neural networks.

Once the neural network-based models of the economic processes are built we are able to develop neural controllers that must meet the following performance objectives for the closed-loop system: tracking the reference target without delay, preventing disturbances from influencing the output, and rejecting noise, i.e., not responding to spurious fluctuations.

It is well known that the first and the third objectives are sometimes mutually exclusive. In other words, a neural controller that improves both the speed of the response and rejects noise is, in general, very challenging to build.

3. INVERSE DYNAMICS NEURAL MODELS FOR THE ECONOMIC PROCESSES

In recent years, there has been a number of neural control learning schemes proposed in the literature. Among these, the inverse model neural control approach, developed by Widrow and et al. (Timotej Jagric, 2009; Tudoroiu N., Khorasani K., Patel R. V., 2000; Tudoroiu N., Zaheeruddin M., 2002; Tudoroiu N., Zaheeruddin M., 2003), has been one of the most viable techniques for implementation of neural networks in control. One reason for its utility is its simplicity. Once the network has learned the inverse model of the economy system, it is configured as a direct controller for the economy system. It is worth to investigate the inverse dynamics control techniques because of its ease of implementation. The objective of a nonlinear dynamic inversion is to invert the dynamic equations of the economy system directly in order to find the control necessary to yield the given output. In *figure 11* is represented a simplified block scheme of the inverse neural control strategy capable of learning the highly nonlinear inverse dynamics of the economic process, similar to the industrial processes.

We could describe, similar to the representations from *figures 6-10*, same architectures for the inverse dynamics of the economy system or economic process. To learn the inverse dynamics of the economy system, we will train the neural controllers off-line. By applying the desired range of inputs to the economy system, its corresponding outputs can be obtained and a set of training patterns can be selected. Once trained, the neural networks could be used to produce the appropriate control input as a function of the desired economy system output. The performance of the neural networks based on these input vectors is observed by configuring it directly to control the economic process. Based on these observations, the neural network structures that give the best performances are then used in the neural control structures of the economy system.

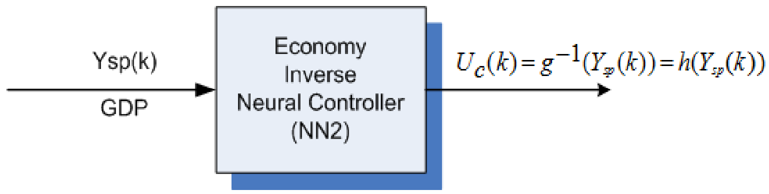

***Figure 11**: The inverse neural control strategy-simplified block scheme*

4. CONCLUSION

This research work is dedicated to investigate the possibility of applying several neural network architectures for simulation and prediction of the dynamic performance of the complex economic processes. Therefore we explore different neural networks architectures to build several neural models of the complex dynamic economy system.

It is our opinion that the lack of economic structure is the main weakness of neural networks in forecasting applications. Due to the black box nature of neural networks, users of forecasts may feel some discomfort if they are unable to give proper economic interpretation to the estimated relationships (Timotej Jagric, 2009). Also it is very difficult to determine which of the explanatory variables are driving the bulk of the forecasts, as comparative statistics are difficult to perform.

On the other hand, economic theory does not always yield a specific functional form that is to be used for empirical verification of the theory. In such cases, neural networks have a great advantage over traditional methods (Timotej Jagric, 2009). A researcher can start with a large network and prune it to the most efficient form.

In future work, we hope to further refine the best neural network models from this research (by considering additional types of networks, different training methods, etc.) for use as forecasting tools to exploit readily available data in order to gauge future economic activity.

Forecast comparisons with other models, such as vector error-correction models may also be considered.

Future work may also investigate the development of new neural network models to forecast other important macroeconomic variables. In these projects we will also examine the possibility of combining forecasts from different models (linear and nonlinear neural network models for economic process simulations).

5. REFERENCES

Béla Lipták (2005), *"Can an automation engineer control the economy?"*, available on-line on site: http://www.controlglobal.com/articles/2005/479.html.

Bishop, C.M. (1995) "*Neural Networks for Pattern Recognition*", New York, NY: Oxford University Press.

Dias, F.C. (1994) "*A Composite Indicator for the Portuguese Economy*", Estudios e Documentos de Trabalho, Working Papers, 18-19, pp. 1-20.

Burns, A.F., and Mitchell, W. (1946) *"Measuring Business Cycles*", New York: NBER.

Fritsch, U., and Stephan, S. (2000) "*Leading Indicators of German Business Cycles: An Assessment of Properties*", DIW Discussion Paper 207.

Hymans, S. (1973) "*On the Use of Leading Indicators to Predict Cyclical Turning Points*", Brookings Papers on Economic Activity. 2. Pp. 339-284.

Khalid, M., Omatu, S. (1990) "*A neural controller for a temperature control system*", IEEE Contr. System. Mag., vol. 12, no. 3, pp. 58-64.

Kim, B., May, G.S. (1993) " *An optimal neural network process model for plasma etching*", IEEE Trans. Semiconduct. Manufact., vol. 7, no. 1, pp. 12-21, Feb.

May, G.S., Huang, J., Spanos, C. (1991) "*Statistical experimental design in plasma etch modeling*", IEEE Trans. Semiconduct. Manufact., vol. 4, pp. 83-98, May.

Smith, T.H., Boning, D. S. (1997) "*A Self-Tuning EWMA Controller Utilizing Artificial Neural Network Function Approximation Techniques*", IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part C, Vol. 20, No. 2, April, pp. 121-132.

Stock, J.H., and Watson, M.W. (1989) "*New Indexes of Coincident and Leading Economic Indicators*" In NBER Macroeconomics Annual, Blanchard, O., and Fischer, S. (eds)- MIT Press, Cambridge, MA.

Timotej Jagric (2009) "*Forecasting with leading economic indicators-a neural network approach*", Business Economics, March, available on-line on the site: http://findarticles.com/p/articles/mi_m1094/is_4_38/ai_111856311

Tudoroiu, N., Khorasani, K. (2007) "*Satellite Fault Diagnosis using a Bank of Interacting Kalman Filters*", IEEE Transactions on Aerospace and Electronic Systems, Vol. 43, No.4, October, ISSN 0018-9251, pp.1334-1350.

Tudoroiu, N., Patel, R.V., Khorasani, K. (2006) "*Neural Network based control strategies for improving plasma characteristics in reactive ion etching*", Elsevier, Neurocomputing, Volume 69, Issues 7-9, March, pp. 786-802, available on line at www.sciencedirect.com, or www.elsevier.com/locate/neucom.

Tudoroiu, N., Tehrani, E.S., Khorasani, K. (2006) "*Interactive Bank of Unscented Kalman Filters for Fault Detection and Isolation in Reaction Wheel Actuators of Satellite Attitude Control System*", IECON'2006, IEEE, Paris, The 32nd Annual

Conference of the IEEE Industrial Electronics Society, Proceedings CD-ROM, ISBN: 1-4244-0136-4, ISSN: 1553-572X, 6 pages.

Tudoroiu N., Khorasani K., Patel R.V. (2000) "*MIMO RIE Neurocontrol Strategies", International Symposium on Systems Theory* "SINTES 10", Craiova, May 25-26, pp.143-147.

Tudoroiu N., Khorasani K., Patel R. V. (2000) "*MIMO RIE Neurocontrol Strategies*", IEE Proceedings-Computerized Machines and Systems, New-York, Vol.147, Issue 2, March. pp.143-147.

Tudoroiu N., Zaheeruddin M. (2002) " *SISO Neurocontrol Strategy Based On Inverse Neuromodel for Improving Air Discharge Temperature (DAT) Performance*", The 27th Annual Congress of the American Romanian Academy of Arts and Sciences (ARA), Proceedings, Oradea, pp.918-922.

Tudoroiu N., Zaheeruddin M. (2003) " *SISO On-Line Training PI Neurocontrol Strategy for Improving Air Discharge Temperature (DAT) Performance*", The 28th Annual Congress of the American Romanian Academy of Arts and Sciences (ARA), Proceedings, Tg-Jiu.

Tudoroiu, N., Yurkevich, V., Khorasani,K. (2003) "*Discrete-Time Control system Design for a Reactive Ion Etching (RIE) System*", 2003 Physics and Control International Conference Proceedings, St. Petersburg, Russia, August, CD-ROM Congress Publication, pp.329-334.

Tudoroiu, N., Khorasani, K., Yurkevich, V. (2003) "*Robust two-scale control system design for Reactive Ion Etching*", European Control Conference 2003, Cambridge, England, September, CD-ROM Conference Publication.

Tudoroiu, N., Manuela Grigore, Jeflea, V., Roxana-Elena Tudoroiu (2008) "*The State Dynamics Estimation of the Nonlinear Processes using Neural Networks as an Alternative to the Kalman Filters Estimation Techniques*", Analele Universitatii "Spiru Haret"- Seria Economie, ISSN 1582-8336, Vol III, Anul 8, nr.8, pp.223-234.

Zaheer-uddin, M., and Tudoroiu, N. (2004) "*Neuromodels for discharge air temperature system*", Elsevier, Energy Conversion and Management, Vol.45, Issue 6, pp. 901-910.

Zaheer-uddin, M., and Tudoroiu, N. (2004) "*Neuro-PID tracking control of a discharge air temperature system*", Elsevier, Energy Conversion and Management, September , Vol.45, Issues 15-16, pp. 2405-2415.