Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 8, 2008, No. 2

# ATTEMPT OF NEURAL MODELLING OF CASTINGS CRYSTALLISATION CONTROL PROCESS

RYSZARD TADEUSIEWICZ, JOANNA GRABSKA-CHRZĄSTOWSKA, HENRYK POŁCIK

*Akademia Górniczo-Hutnicza, al Mickiewicza 30, 30-059 Kraków, Poland*
*Corresponding Author: rtad@agh.edu.pl (R. Tadeusiewicz)*

**Abstract**

The paper presents a successful attempt to create a neural model for a casting process. Neural network, following the training process, is able to predict selected effects of casting cooling (as a number of metal crystals in certain point of the casting), based on the data referring to the cooling method (the casting is cooled with water, the flow of which is controlled according to a set time algorithm). Various structures of neural networks used for solving the problem in question have been illustrated and the results obtained have been discussed. After creating several successful versions of a straight model, an attempt was made to set up a reverse model, it is such model in which expected control result is supplied to the input of the neural network (here, the number of crystals in a casting), and the network is expected to generate output signal stating how the cooling of the casting should be controlled to accomplish that objective. This has proven impossible, and the relevant reasons and circumstances have been presented.

**Key words**: casting, casting cooling, neural network, neural model, cooling control, reverse model

## 1. INTRODUCTION

Numerous examples of successful applications of neural networks for modelling and controlling various metallurgical processes mentioned in literature (Cho et al., 1997; Dixit & Chandra, 2003; Jansen et al., 1999; Larkiola et al., 1998; Lee & Choi, 2004; Lula et al., 2007; Morris & Martin, 1998; Pican et al., 1998; Pichler & Pfaffermayr, 1996; Zhao et al., 2005), as well as the authors' own experiences in that field, related to setting up neural models of many other (non-metallurgical) systems and processes (Lula et al., 2007; Szaleniec et al., 2006, Tadeusiewicz et al., 2007), are encouraging to make an attempt of neural modelling of castings crystallisation control process and to seek neural-supported algorithm for castings crystallisation control. Such attempt was made at AGH Automation Department and, although it was not a complete success, its findings were interesting enough to be described herein in order to give an account of the original results, and to present the formula that can encourage other scientists to undertake similar, hopefully more successful, research.

## 2. TECHNOLOGICAL ASPECT OF THE TASK

During the attempt to create a neural model of the casting crystallisation process, with various cooling strategies (water), the casting crystallisation process is discussed for which relevant values characterising the materials and the crystallisation process itself, given in table 1, are specified.

The major parameter in question is the mean value of crystal radius in a chosen point of a casting. Such parameter is closely linked with the average number of crystals. Such values are depended on the


ISSN 1641-8581

initial mould and metal temperatures and the assumed additional, dynamic, cooling system (in the task in question, tubing with circulating coolant is considered) of the casting during the crystallisation time.

***Table 1.*** *Selected parameters of the casting and moulds being modelled*

| Parameter | Simulation experiment | |
|---|---|---|
| | Casting | Mould |
| Heat conductivity, W/mK | 210 | 1 |
| Specific heat, J/(kg K) | 1180 | 1333 |
| Density, kg/m$^3$ | 2550 | 1500 |
| Crystallisation heat, J/kg | 3.73E+5 | --- |
| Eutectic temperature, ºC | 610 | --- |

Neural networks are applied in the task discussed for two purposes. On the one hand, the network is arranged in such a manner (as regards the number of input, output and hidden neurons) that it serves as a model in the process being discussed. This implies that the information, on how the coolant flow during casting crystallisation was controlled, is supplied to the network input, and the network is expected to predict the average number of crystals. The network is taught in such a manner that the data from physical measurements (collected while producing experimental castings) is supplied to its inputs and outputs, which are, yet, supplemented by the results obtained from the crystallisation simulation. This is because the physical experiment, that would supply such quantity of experimental data to allow effective network training, would be too costly and laborious, as hundreds or thousands, in some cases, training data sets are required.

Question can be posed as to the relevance of neural modelling (i.e. computer-aided) of data coming from simulation of the process being discussed, and consequently, from computer-aided calculations. This question can be replied to in two aspects.

Firstly, a neural network (which is able to approximate any well-defined empirical relationship) can be very fast in modelling the phenomenon being discussed. Contrary to the model being the source of data, treated herein as the training data, a neural model can also supply (following the training process) desired solutions very fast, with a minimum requirement for computational power. Such behavioural model, not a cause-effect model, may be easily solved even using industrial computers of low power, whereby the speed allows to check several control strategies for the process discussed in a short time and to select the one which most reliably guarantees meeting the practical objective. A model, the solutions of which have been the grounds for training the neural network in question, cannot serve as an "experimental field" allowing proper selection of object control. The model referred to a very precise simulation of physical and chemical processes during the cooling and crystallisation of castings, so it must have made use of models described by a complex system of partial differential equations. Such model can be solved with a supercomputer (and such machine was used for the research) having as much time available as necessary, but it is not practicable in the industrial environment.

There is another aspect encouraging the use of neural network as a model for the process being discussed. As is well-known, a neural network is capable of depicting any dependency as a relationship of certain input and output data, regardless of which data is of the cause or effect nature. For that reason, with the use of a neural network, a dependency of so-called reverse model can be found, which cannot be set up based on a typical scheme of a conventional simulation. The actual process involves the possibility to select control signals, hence desired control effects can be expected, such as, for example, proper inner structure of the casting, which is one of the effects of specified static and dynamic properties of the process being discussed and control structure adopted. The simulation model utilising the knowledge of the physical and chemical nature of the processes and cause-effect relations, also involves an imposed direction for the calculations to be made: after the input of an appropriate control action, information is generated as to what its effect was (average crystal radius in a selected casting point).

Meanwhile, the neural model is free from such limitation, as it is not based on the reproduction of cause relations, neither it refers to the physical sense of the phenomena, as all it can do is to reproduce the **coincidences** of signals arbitrarily chosen as the inputs and others indicated as the outputs. From the technical point of view, there are no objections not to select, in the neural model, the effect of a certain control action as the network **input** (in the example in question, the effect is the average crystal radius in a selected casting point, which is closely bound to the average crystal number), and to supply a control action as an **output**, which has led to such an effect (in the example discussed, the control action involved specified time scheme for cooling the casting with liquid). If only enough findings are in place, in which specified effects could have been linked to certain causes, the neural network is capable of learning how to associate its input (number of crystals) with its output (cooling scheme). This fact forms the grounds for setting up a **reverse model**, in which the desired effect appears in the input, and the suggestion as to what control action should be performed to achieve such effect, can be received in a form of network output signals (figure 1).

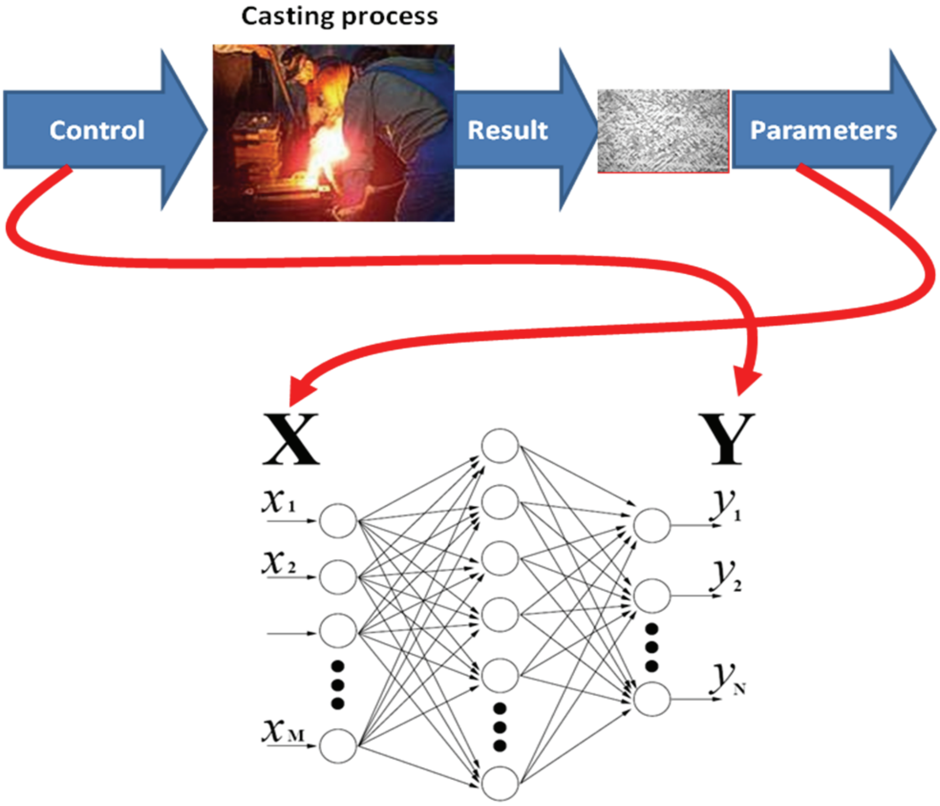


*Fig. 1. Principle for setting up a reverse neural model*

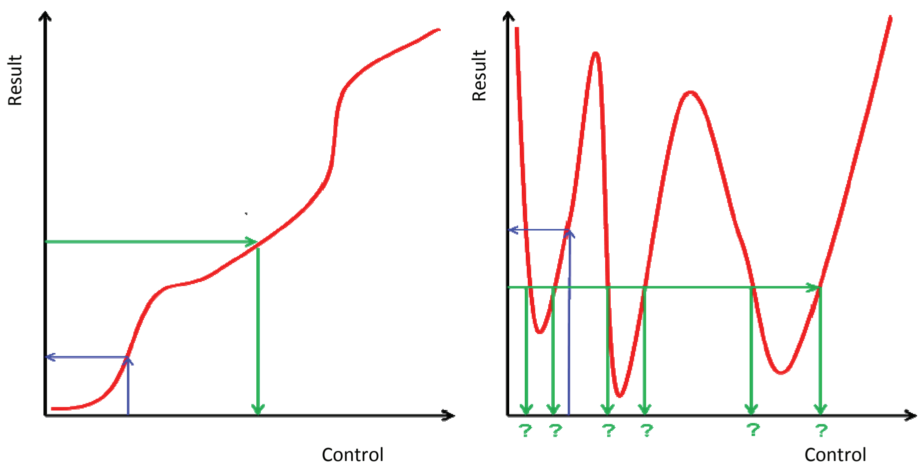


*Fig. 2. Possibility to create a reverse model with a monotonous (left) and non-monotonous dependency of the effects on the control action. Description in the text*

Further, the paper will focus on the attempt of creating a neural model to replace the simulation algorithm, and experiments will be described (unsuccessful unfortunately) related to setting up a reverse model for the casting process being discussed. The failure, however, was not attributable to the incorrectness of the method symbolically presented in figure 1, as it resulted from the fact that the dependence between the quantity of crystals and the cooling method has proven to be **non-monotonous**. Such dependency makes it impossible to find a reverse model (figure 2) as such model does not exist.

The difficulty encountered here is of fundamental nature, since while for the monotonous dependency (figure 2 on the left) there is always a straight reply to the questions "what will be the effect of certain control action?" and "what control action should be applied to achieve the desired effect?", for the non-monotonous dependency (figure 2 on the right), only the first question can be precisely answered.

In order to describe the technological process being subject of neural modelling, the mould-casting-cooling tubing arrangement shown in figure 3 has been adopted. For the simulations (which are the grounds for neural network training, with the aid of a precise model) the time in which the number of crystals stabilises, has been assumed to be 400 seconds.

"Bang-bang" cooling control system has been adopted (comprising two levels: 0 – no cooling; 1 – cooling on) with a tube through which the coolant at 20°C flows, positioned centrally inside the system being discussed.

The cooling control algorithm was as follows: Control time (400 seconds) divided into N equal sections, in which the control value is either 0 or 1. The simulation has been carried out for N = 1, 2, 4, 8 and 16. The control vector referred to as C, is a binary vector composed of N binary position, whereas occurrence of 1 values in a certain position of such vector means that in the given time interval the cooling was on (water flew through the tube), and filling the binary number with 0 means that the cooling was off in a given time interval. Calculations were made for all combinations of the control procedure, and their results were used for the neural network training process.

In physical simulations of the cooling and crystallisation process (not included herein, but utilised as network training data), the below results were obtained for a chosen casting point (the following designations were applied: ILK – average number of crystals, C – control vector as described above).

- N = 1: cooling off (C = 0) ILK= 252400; cooling on (C=1) ILK=65405;

Refer to Table 2 for sample results for N = 2. In that case the combinations of cooling and no-cooling periods were used as the control signals: C=00; C=01; C=10; C=11. The switching interval was 200 seconds. Since, to ensure comparability of results, all neural simulations were performed using the same neural network, taught with the data for all considered N values, the vector of constant length equal $2^N$ for N = $N_{max}$ = 16 was assumed for the structure of the model input, which is represented in the structure of the Table 2. The table is additionally provided with a column containing interpretation of the control vector C (considered as a binary number) in an equivalent decimal value. Analogical compilation of simulation results for N = 4 is show in the Table no. 3.

***Table 2.*** *Simulation results for a physical model, which form the grounds for setting up a neural network model. The result for N=2 is only presented (which corresponds to $2^N$ = 4 moments of switching the control signal during the entire crystallisation period.)*

| Control scheme | | | | | | | | | | | | | | | | Number of crystals | Decimal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 252400 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 434468 | 255 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 408272 | 65280 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 465405 | 65535 |

For higher N values, there is no point in referring to relevant tables, as for N = 8, the relevant table comprises 256 lines, and for N = 16 lines there are as much as over 65 thousand of these, hence, just to exemplify, it is worth a mention that for N = 16 for C = 1011001110110110 established value ILK = 458823.

***Table 3.*** *Simulation results for a physical model, which form the grounds for setting up a neural network model. The result for N=4 is only presented (which corresponds to $2^N$ = 16 moments of switching the control signal during the entire crystallisation period.)*

| Control scheme | | | | | | | | | | | | | | | | Number of crystals | Decimal |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 252400 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 306969 | 15 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 434060 | 240 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 434468 | 255 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 398759 | 3840 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 398852 | 3855 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 460833 | 4080 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 460833 | 4095 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 312427 | 61440 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 326543 | 61455 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 449252 | 61680 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 449252 | 61695 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 408272 | 65280 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 408272 | 65295 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 465405 | 65520 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 465405 | 65535 |

## 3. MODELS OF NEURAL NETWORKS – STRAIGHT MODEL

For the data described above (of which only part was quoted) attempts were made to set up neural models. The objective was accomplished as far as the model simulating the behaviour of a real process is concerned, while sustaining the same - as in the process (and its physical simulation) – order of outputs and inputs – i.e. the control action was applied to the neural model input and (of course, following the network training) the solution was expected giving the number of crystals achieved in the casting. In order to solve the task, at first the method was developed for representing the data describing the problem studied in the neural network in question.

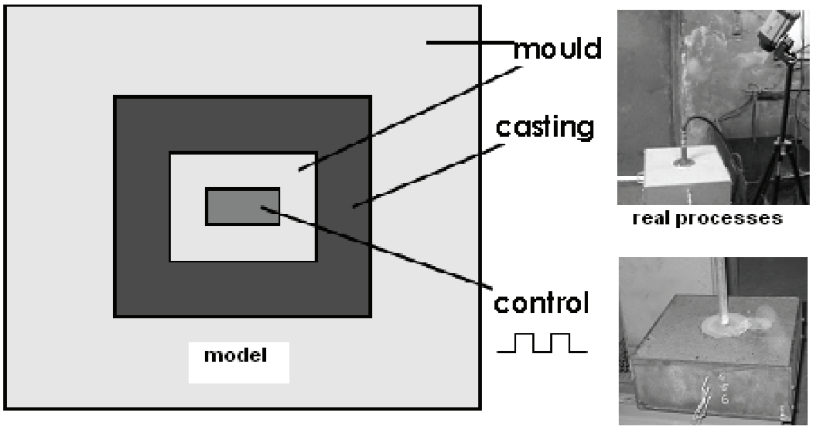


***Fig. 3.*** *Mould – casting – control with a cooling pipe*

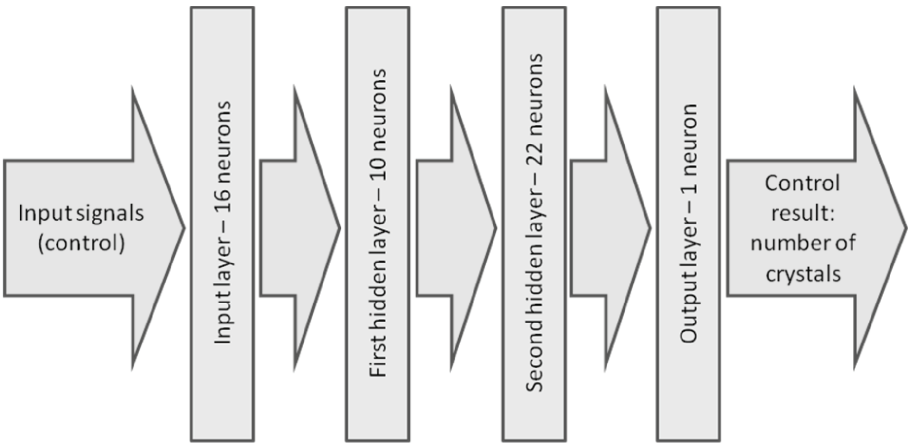


***Fig. 4.*** *Structure of neural network modelling the process described*

At the beginning, 16-component binary vector was applied to the network input (corresponding to the above-mentioned C control vector), which meant that all studied networks had the same number of 16 inputs, and numerical value was expected in the output, which – after appropriate re-scaling – could have been regarded as the number of crystals. Since during most studies, networks were used which had neurons of sigmoidal characteristic in the input layer, which – as is well-known – accepts input values from the range with both ends open (0, 1), during most experiments the fractional part was removed from the results (not specified in tables 2 and 3, but made available by the researches carrying out the simulation studies using a precise physical model) and, from all results, the value specified as a minimum number of crystals observed was subtracted (namely 252 399).

After determining the number and nature of the input signals to the neural model (16 inputs of binary values) and after selecting the method for representing the solution (one real number, scalable for the variability range corresponding to the number of crystals resulting from simulations and measurements), the last thing to do was to select the type and structure of the network. It was decided that MLP type network structure would be sought (known as well performing in other studies of similar nature (Szaleniec et al., 2006)) with two hidden layers.

To benefit from the functions offered by the software used (basically Statistica Neural Networks by Statsoft, a part of Statistica package version 7.0, was used), so called "automatic network designer" was used, it is a network structure optimisation algorithm incorporated in the package, based on the evolutional designing technique.

As a result of the optimisation process, network structure was achieved (indicated by the program as optimal, yet with no such guarantee) which modelled the required dependency of the number of crystals in a selected moulding point, on the course of the control signals. The structure is shown in figure 4. For the figure clarity, instead of drawing separate neurons and their links, the network structure is illustrated only with an indication of the layers occurring there and giving the number of neurons in individual layers. The links between the neurons of individual layers were effected in accordance with the commonly applicable "each to each" principle.

Following the training, which had been effected through the reverse error propagation method, making division of the data coming from the simulation, according to the scheme:

– training set - 40 thousand cases
– validation set - approx. 12 thousand cases
– test - approx. 12 thousand cases

results were accomplished that can be regarded satisfactory. The mean approximation error of the real process with its neural model yielded (after converting to the number of crystals, set by the neural network) **2149.7**. In order to evaluate that error, it is worth a mention, that after subtracting the minimum value, the mean value of signals determining the approximated number of crystals was **187 544,** which means that the average relative error of neural approximation was kept within 1%. Such accuracy can be regarded satisfactory for technology-level research. Attention should also be drawn to the figure 5 which describes the progress of specific values of errors made by the network for various control signals.

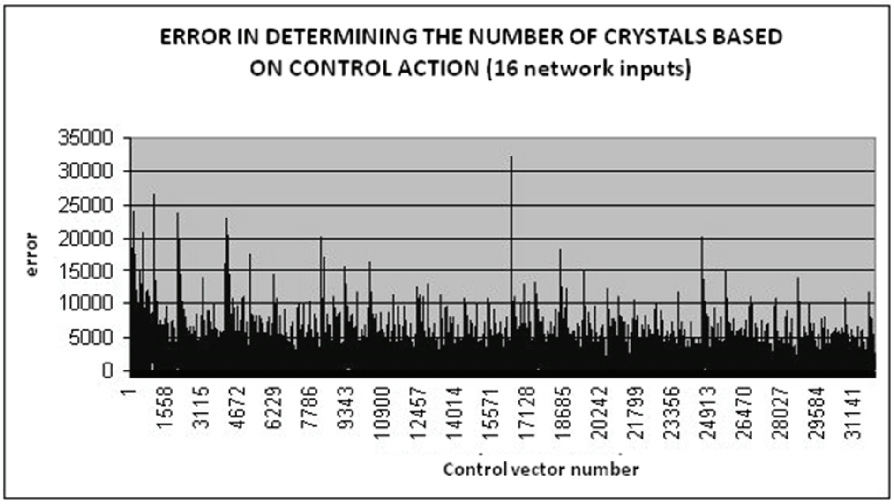


*Fig. 5. The errors made by a neural network while determining the number of crystals forming in a casting depending on cooling system control adopted*

Analysing figure 5, an argument may be submitted that the network had performed relatively well in the task, as the errors are small and quite uniformly distributed all around the input signals (control space). It can also be noted that the neural network exhibits very effective computational quality, as it was noticed that, after training the network, the time to receive neural model reply is over a hundred times shorter, than that for computer-aided simulation based on a precise process model, using differential equations. Another point it that, for a specific task, as it is the case while looking for the answer (number of crystals) with definite (but very large) number of known control schemes, the fastest way to find the solution is to use RAM large enough to accommodate all control strategies (expressed by C vectors) and known from simulations effects of such strategies (expressed by ILK values), and using the *look-up-table* method, commonly used in signal processing or automation. For example, using such table, it can be found that for the ILK value sought in a selected casting point, of 358900, the control should have the form C= 1111110000001000 (then, real ILK = 358923 is obtained, i.e. error produced in that method is of around 0.01 %). The control variant found means that, for the first 150 seconds, the flow should be on, and off for another 150 seconds, then on for 25 seconds and off for 75 seconds. This proves that such a solution can be effective, however, the idea will not be discussed further in the paper, as irrelevant to the subject hereof.

## 4. MODELS OF NEURAL NETWORKS – STRAIGHT MODIFIED MODEL

While studying the function of the neural model of the casting process discussed, of the structure shown in figure 4, it was concluded that individual input signals are of varied influence on the final effect achieved. In particular, it has been found that the input signals located at the start or end of the binary vector creating the C signal, are of relatively low effect on the network output signal. It can, for example, be concluded that the casting cooling strategy in the initial solidification phase (when the metal being cooled remains still in liquid state), and towards the end (when the crystallisation process has been mostly completed), does not considerably affect the studied output parameter as the number of crystals is. On the other hand, halfway the cooling process, a phase occurs in which changes of the cooling strategy can adversely affect the number of crystals obtained, and that is what was revealed while surveying the neural model sensitivity to individual components of the input control vector. Figure 6 shows which components of the C vectors were classified as significant for the cooling process outcome (marked red), and those found to be of minor effect on the crystallisation result (marked blue).

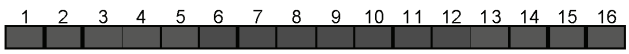


*Fig. 6. Sensitive and insensitive components of the control vector. Description in the text*

It needs to be stressed that the distribution of the control vector, shown in figure 6, was established solely based on the neural model sensitivity survey and it is of phenomenological character (not the cause-effect), hence, its physical interpretation as indicated above has to be considered only in hypothetic terms.

Having a distinguished subset of control signals, to which the model was particularly sensitive, an attempt was made to set up a neural model based only on seven "significant" inputs, separated during the above-mentioned studies (marked red in figure 6). Again, a program seeking for optimum neural network structure was used, utilising all data acquired as a result of the physical simulation, divided into subsets:

- training set - 40 thousand cases
- validation set - approx. 12 thousand cases
- test - approx. 12 thousand cases

The result of such structural optimisation is shown in figure 7, which deserves careful attention while compared with figure 4. What appears to be a distinguishing feature of that new neural model (figure 7) is the considerably larger number of hidden neurons which turned out to be indispensable in that case. A conclusion can be drawn here that the network which is supposed to come to proper conclusions, based on smaller number of references (less input signals in this case), has to be more "intelligent". Further on, referring to reasonable inference that the number of engaged (and subjected to training!) neurons is the measure of the neural network intelligence, significantly different distribution of such intelligence is striking in the new model (figure 7), as compared with the model utilising broader resources of input data (figure 4). Namely, in the new model, the greatest increase of the "demand for intelligence" was noted in the first hidden layer, in which the input data is analysed. The growth of demands as to the second hidden layer, although noticeable, was considerably weaker. For that reason, it can be concluded that the process of genetic optimisation of the network structure has demonstrated that, for a restricted number of the input vector components, the analysis of the input data becomes far more challenging and, to some degree (but less significantly), it becomes more difficult to correctly determine the final result.

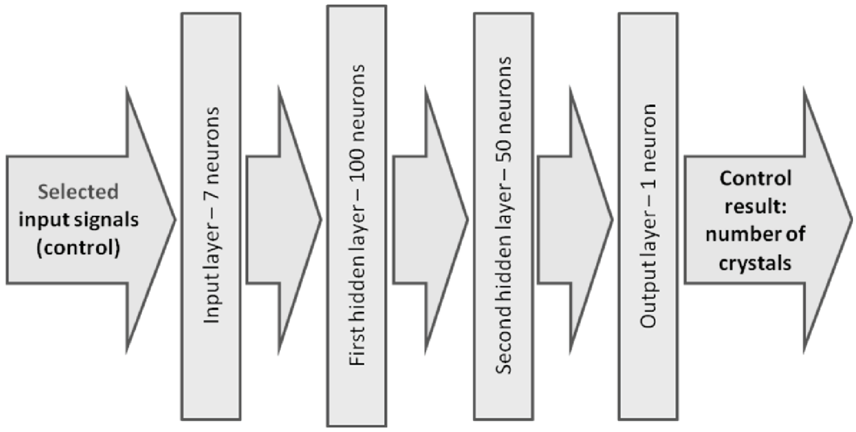


***Fig. 7.** Structure of a modified neural model*

The model shown in figure 7 following the training process assumed the result (number of casting crystals) with an average error of **2653**, it is not much worse than that of the model with full input vector, for which the error was **2149.7**. This means that, regardless of what the result may be of the physical interpretation of the input vector restricted to the data of only the middle cooling phase, the neural model has expressly proven the effectiveness of such restriction.

## 5. NEURAL NETWORK MODELS – MODEL UTILISING CONTROL FEATURES

Another attempt to enhance the neural model of the process studied, was to replace the precise control vector (showing in what moments coolant flew through the casting) with a vector referring to the features of such control process. Such features were no longer binary values, but comprised real numbers which can be regarded as various **characteristics** of the control process adopted.

The first component of the input feature vector was the value S, denoting the summarised number of ones in the control signal C being studied. That feature is important for obvious reasons related to the physical nature, as it is a measure of the summarised degree of cooling intensity with the use of liquefied coolant. The more ones, the longer (altogether) the coolant had been removing heat from the solidifying casting. Of course, the parameter does not completely cover the cooling control procedure, as it does not specify whether the heat removal took place in the beginning, halfway (critical period!) or at the end of the crystallisation process, yet, it is still important. Examples of training data summarised in figure 8 indicate that the S parameter affects the output value of the model relatively frequently (ILK parameter), although the distribution of the results is obviously studied, which is attributable to the fact that the same summarized number of ones (especially for small S values) can be obtained in many ways, by varying the cooling intervals in time.

Another component of the input feature vector was the Z value, regarded as the code of the characteristic maximum compound of ones. It was the measure of the time-dependent control variability.

The numerical value of the code mentioned depended on how large groups (clusters) of adjacent ones occurred in C vector describing the control action discussed. The numerical value Z corresponds precisely to the number of ones occurring in one of the longest noticeable series. Z = 1 refers to the situation when ones are only single (the coolant flow was started in short pulses only), *Z* = 2 when at least once (or more often), after switching on the coolant flow, such flow was maintained for two time quanta and so on. Generally, it can be asserted that small values of Z refer to strong control variability, and large Z refers to a "preservative" control – after switching on, the cooling is not switched off too soon.

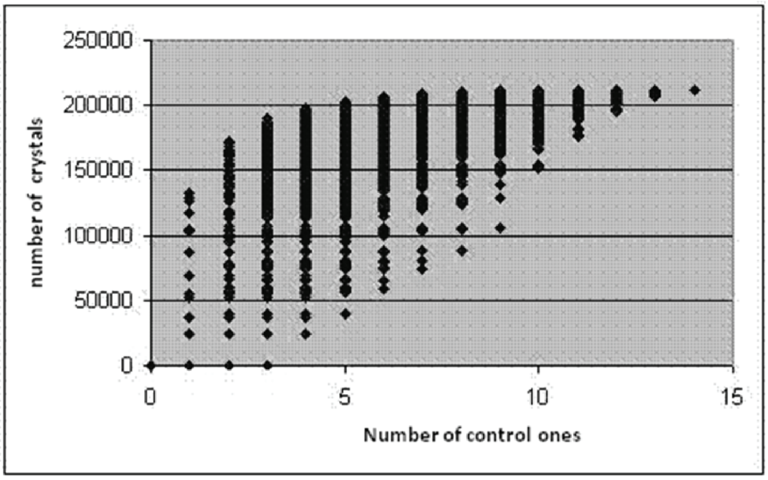


***Fig. 8.*** *Dependence of the crystal number on the S parameter (number of ones in a control action)*

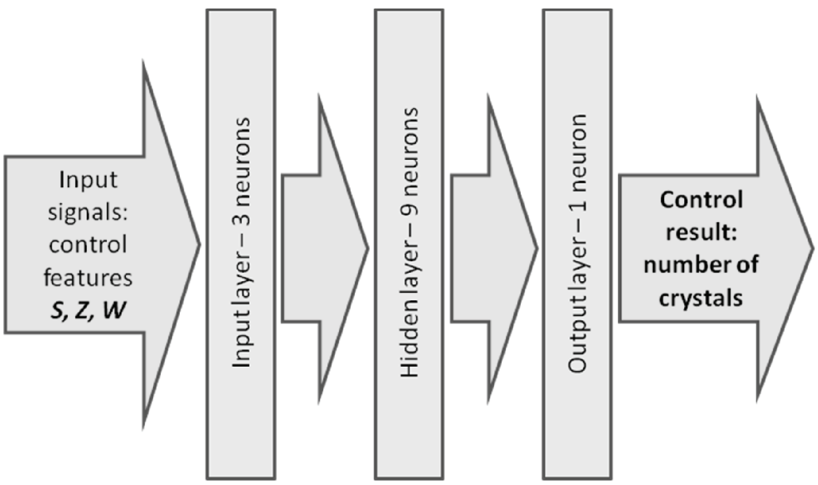


***Fig. 9.*** *Network based on control signal features. Description in the text*

The third and last value of such input vector of features was the W value, determined as a numerical (decimal) equivalent of the "value" encoded in the control vector C in a binary form. Such equivalent has already been referred to in tables 2 and 3 in their last columns. Such feature is obviously of no substantial relevance (attempt to set up a model in which, instead of 16 inputs as shown in the figure, only one input was present, supplied by a binary representation of the binary C vector, had proven to be clearly ineffective), yet, it is useful for differentiating vectors for which the other values of the remaining two features are identical.

For the assumptions given as to the neural network input, a system for automatic creation of an optimal network has been launched, and a network of a structure as shown in figure 9 was obtained.

It has to be noted that such network is much more economic than both variants discussed before, as it has only one hidden layer containing only 9 neurons. Such neural model learns very easily and can be efficiently implemented. Unfortunately, the quality of results obtained in such model is worse than that of the complete model, i.e. using the entire control vector C, and also in the modified model in which only 7 out of 16 control signals are taken into consideration, as the mean error value that was achieved for the network shown in figure 9 was **10 914** which stands for a relative error of some 6%, for an average number of crystals **187 544**.

Unsatisfactory value obtained for the input vector *<S, W, Z>* in the MLP network (*Multi-Layer Perceptron*) shown in figure 9, encouraged one more attempt of modelling the dependency being studied with the aid of RBF class network (*Radial Basis Functions*). Automatic optimisation of network structure led to the structure layout as shown in figure 10. Attention is drawn to a very large number of neurons of the radial layer present in that network after its optimisation, which was tried to be stressed by the form of figure 10, without maintaining the proportion.

Unfortunately, despite utilising so great radial layer, the prediction of the control result obtained with that network still indicated a relative error of some 6%, which implies that the result (by no means unsatisfactory) appears to have connections with rather limited quantity of information as regards the control run, which is successfully included in the limited vector of features <S, W, Z>, rather than the quality of neural networks aspiring to perform the role of models in the discussed metallurgical process.

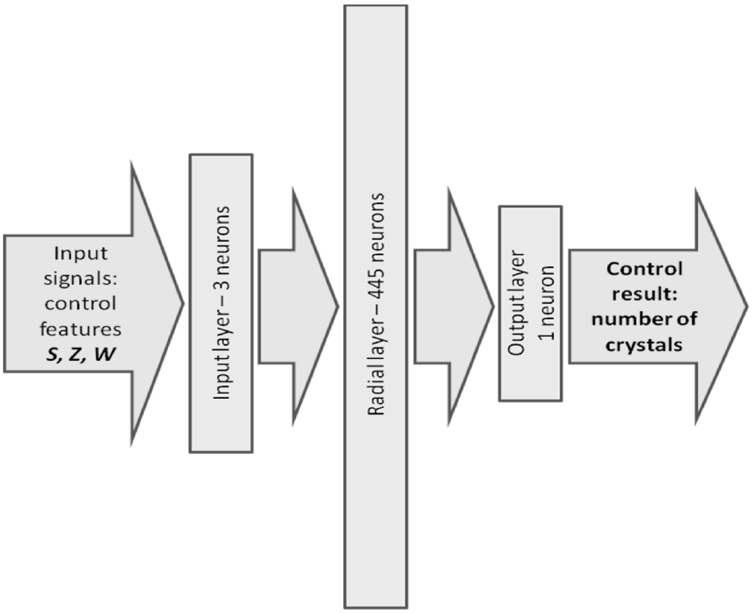


*Fig. 10. RFB network utilised for the studies*

## 6. NEURAL NETWORK MODELS – REVERSE MODEL

The above-mentioned models show that the neural network is able, with satisfactory accuracy (as regards majority of applications), to replace a precise physical simulation model and can determine the control effect (in a form the discussed number of crystals) for any combination of control signals (of the cooling process), whereas the outcome from a neural network can be obtained very fast and with minimum calculation effort. However, since the very beginning of the studies described herein, it had been obvious that the reverse model would be most desirable here, to guarantee, for a given number of crystals, to produce a casting with such required number of crystals (achievable with some assumed accuracy, of course).

Encouraged by good results of "straight" process modelling, we made attempts to set up a model that would be able to solve the reverse task and we ended up in defeat.

Of course, we succeeded in setting up the neural model, and the attempt to optimise its structure was even successful, as it has to be admitted that it was a difficult model: with one input and 16 outputs (figure 11). By overcoming the difficulties created by a professional modelling application with such an odd network structure (*Statistica Neural Networks*, version 7.0), the network was successfully trained (with the aid of the *Backpropagation method*). The result obtained was, however, completely unsatisfactory, as the mean output error was at the level of 0.5 (in outputs for which – let us remind – signal 0 or 1 had been expected, following the control principles adopted in the process studied). That meant that the results referring to the control, given by a trained neural network, were as irrelevant as those supplied by a random-number generator!

Superficial analysis of the results produced by the network for individual cases proved the model to be completely impracticable. Observing any network output (for example number 1 output) while supplying the input values corresponding to all data from the training set, over 32 thousand cases were counted, in which the output signal (after being appropriately rounded) was 0, while it should have been 1 or vice versa.

Complete failure to set up the reverse model for the control process, in case of acceptable quality of

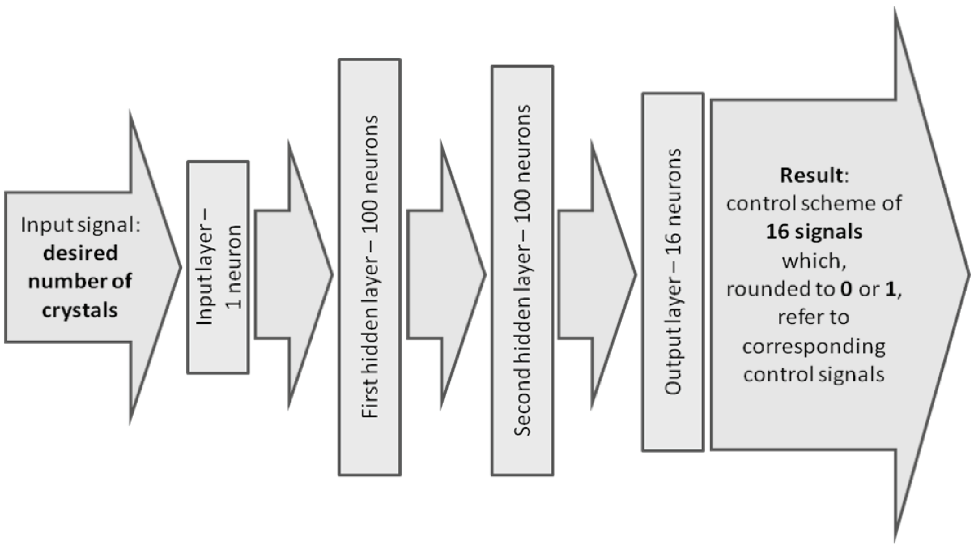


*Fig. 11. Structure of the model that was to determine control signals for the cooling process for a given control objective in a form of the desired number of crystals in a selected casting point*

straight neural models obtained, required the causes to be found. Initially the cause of the failure was sought in inappropriate problem formulation for the neural network model. A striking anomaly in the model being set (in relation to all neural network referred to in the literature) was a large size of the output vector C (16 components!), that the network had to generate based on one only ILK input signal.

Attempt was made to make up for that disproportion by replacing the network output signal in a form of C control vector, with a single numerical value, being the decimal equivalent of the vector value considered as a binary number. Appropriate numerical values have been already utilised in section 5, in which they are marked W. They are also shown in the last column of table 2 and 3.

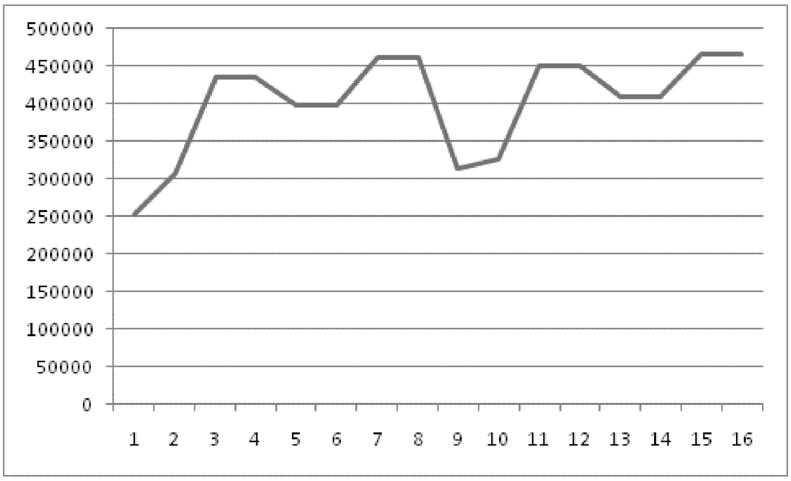


*Fig. 13. The dependence of the number of crystals (vertical axis) on the number of control vector (on the horizontal axis) for N = 4 indicates non-monotonous character of the corresponding dependence*

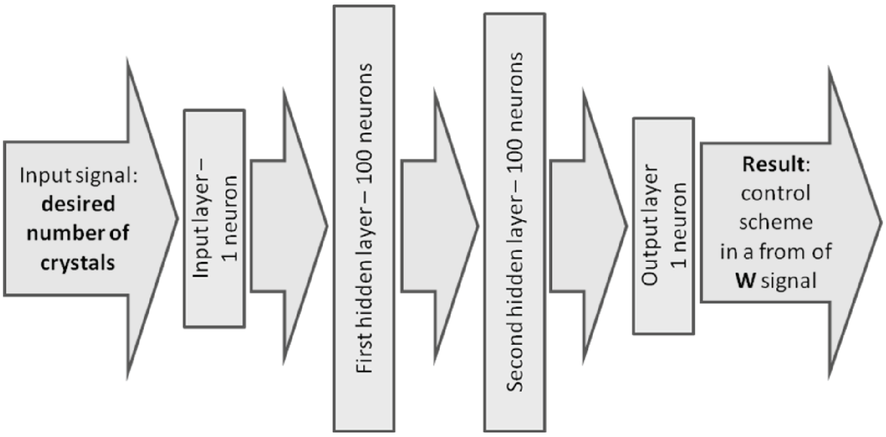


***Fig. 12.** Reverse model scheme with other output definition*

A reverse model of the process studied was formulated accordingly, in a form of a network with a single input, to which IKL values were supplied, and with one output, in which correct W values were expected (figure 12).

Unfortunately, the training of the neural network shown in figure 12 was unsuccessful either: the mean error of W signal determination was **16 321** which, for the signal values within the range of [0, 65535] indicates an error of around 25%.

## 7. ANALYSING CAUSES FOR WHICH, IN THE TASK IN QUESTION, NO REVERSE NEURAL MODEL EXISTS

Another failure induced the causes of the breakdown to be searched not in the neural network structure, but in the nature of the problem being solved. Careful investigation of the data contained in table 3 (plotted in figure 13) allows a conclusion that the problem discussed is of non-monotonous nature. As the figure shows, the same value of the ILK function (marked as a plot ordinate) can be obtained for several various values of control vectors (represented as the following numbers in the axis of abscissa).

The same is confirmed by plots in figures 14 and 15, showing the situation for N = 8 and N = 16, whereby the latter has been presented with low resolution deliberately, since for the number of points on the axis of abscissa above 32 thousand, the precise graph is impossible to be reproduced in printing, and is difficult from editorial point of view. In figure 15, the minimum value (252 399) has been additionally subtracted from the ILK values shown in the vertical axis, to better depict the variability of the characteristic being studied. Moreover, the studied phenomenon was presented only for half of the control vectors possible in that control variant, as the graphic tool used (MS Excel) refuses to create graphs in which the number of points on the axis of abscissa is above 65 thousand.

## 8. APPROXIMATE NEURAL MODEL–REVERSE MODEL

The comparison of plots in figures 13 – 15 with the figure 2 demonstrates that, in the problem studied reverse model **cannot** be created, as such model simply does not exist in the case being studied.

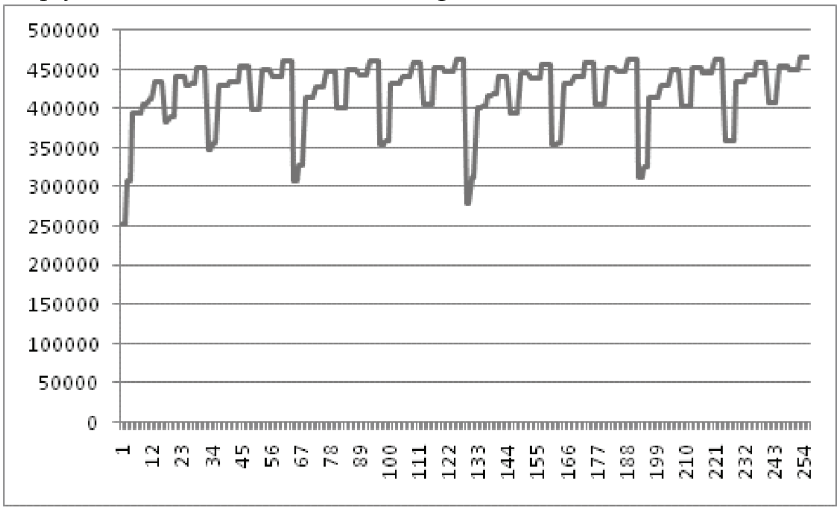


***Fig. 14.*** *Dependence of the number of crystals on the number of control vector for N = 8*

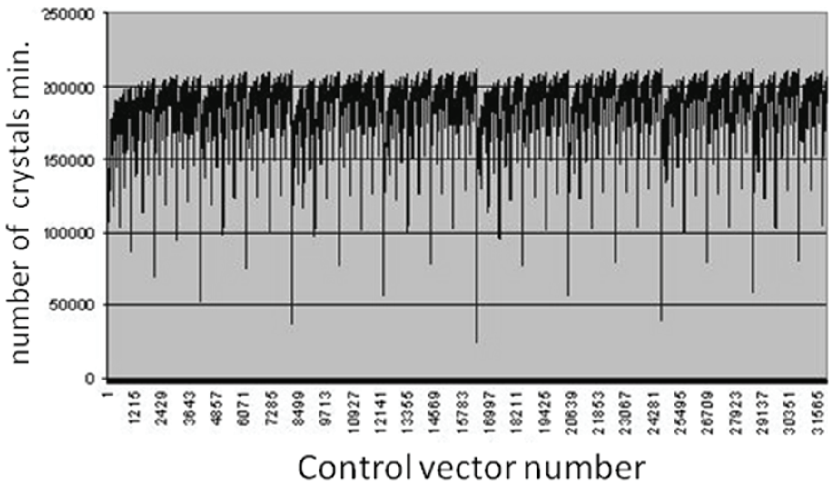


***Fig. 15.*** *Dependence of the number of crystals on the number of control vector for N = 16. Differences of that presentation in relation to figures 13 and 14 are described in the text.*

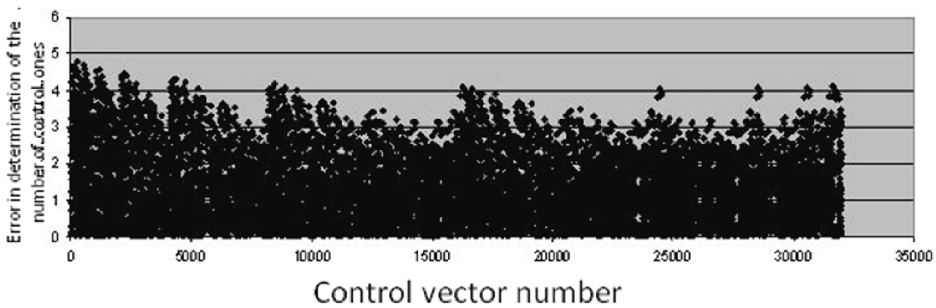


***Fig. 16.*** *Distribution of error valued for a neural network forecasting the number of ones in the control vector*

The prospect of neural support to the casting process is, however, not completely unfeasible. For the reasons discussed above, it is impossible to precisely determine the required control actions C for the casting cooling process, in cases when certain predetermined number of crystals is desired in the distinct point of casting ILK. Analysis of the figure 8, however, suggests that, if only selected (substantial!) control feature (as is the summarised number of ones in the C control vector) is taken into consideration (to remind, the feature marked S in section 5), then the dependency

$$ILK = f(S),$$

which is **not** unequivocal (as illustrated in figure 8), can be **approximately** reversed, to receive **rough** estimation of the reverse dependency

$$S = f^{-1}(ILK).$$

This means that it is possible to create a neural network which, after supplying to its input the number of crystals desired in the casting discussed, gives an indicative S value, i.e. the number of ones in the C control vector, which is linked in section 5 with a specified physical interpretation, which can be a useful hint for a process engineer. During the studies described herein, such neural network was built and tested with average error value of **1.18**. Such result is not fully satisfactory, since for the variability range of the S parameter being approximated of [0, 16], it means that the network performs with an average error of 7 %, yet, taking into consideration the **difficulties** in finding a reverse model for the process studied, and in view of the fact that an almost uniform distribution of error values had been successfully achieved for the forecasted number of ones in the control signal for various control vectors (figure 16), such result can be regarded satisfactory.

## 9. SUMMARY

The results described herein have proven that it is possible to build a model of casting process using the neural network technology (precisely, casting cooling process), and such model can be fast and effective. However, it has occurred that due to the nature of the process being modelled (as revealed in the data provided for network training), it is impossible to set up a reverse model.

The subject studied has proven to be difficult in the neural network technology. The most time-consuming calculations and most complicated training tasks for the neural network were required, for obvious reasons, for N = 16 (number of combinations of control signals is as much as 65536). So abundant training set was initially refused by commonly used applications for creating, training and using neural networks (basically Statistica Neural Networks by Statsoft – a part of Statistica 7.0 package – was used).Yet, appropriate neural network was successfully implemented and suitable studies were performed, and their results presented herein.

## REFERENCES


Cho, S., Cho, Y., Yoon, S., 1997, Reliable Roll Force Prediction in Cold Mill Using Multiple Neural Networks. *IEEE Transactions on Neural Networks*, 8 (4), 874-882.

Dixit, U. S., Chandra, S. A., 2003, Neural Network Based Methodology for the Prediction of Roll Force and Roll Torque In Fuzzy Form For Cold Flat Rolling Process. *The International Journal of Advanced Manufacturing Technology*, 22 (11-12), 883-889.

Jansen, M., Broese, E., Feldkeller, B., Poppe, T., 1999, How Neural Networks Are Proving Themselves in Rolling Mill Process Control, *Siemens Newsletter: Metals, Mining and More*. [Online].

Larkiola, J., Myllykoski, P., Korhonen, A. S., Cser, L., 1998, The Role of Neural Networks in the Optimization of Rolling Process, *Journal of Materials Processing Technology*, 80-81, 16-23.

Lee, D.M., Choi, S.G., 2004, Application of On-line Adaptable Neural Network for the Force Set-up of a Plate Mill. *Engineering Applications of Artificial Intelligence*, 17, 557-565.

Lula, P., Paliwoda-Pękosz, G., Tadeusiewicz, R., 2007, Methodology for artificial intelligence and applications in economy and management, University of Economics, Kraków.

Morris, A.J., Martin, E.B., 1998, Neural Networks: Panacea or Pragmatic Solution? In: *Applications of Artificial Neural Network Systems in the Steel Industry,* Brussels, Luxembourg: European Communities, 9-42.

Pican, N., Alexandre, F., Bresson, P., 1998, Artificial Neural Networks for the Presetting of a Steel Temper Mill. In: *Applications of Artificial Neural Network Systems in the Steel Industry*, Brussels, Luxembourg, European Communities, 253-259.

Pichler, R., Pfaffermayr, M., 1996, On-Line Optimisation of the Rolling Process: A Case of Neural Networks, *Steel Times*, 224, 310-311.

Połcik H., Kluska-Nawarecka S., Bieniasz S.: Selected Problems of Computer Aided Control of Solidification Casting Process, Naukowyj Żurnal 4.2007 "BICHNIK", Chemiczeskogo Nacjonalnogo Univeristeta, Chmielnickij, 2007. No4, T.2 (94).

Szaleniec, M., Goclon, J., Witko, M., Tadeusiewicz, R., 2006, Application of artificial neural networks and DFT-based parameters for prediction of reaction kinetics of ethylbenzene dehydrogenase. *Journal of Computer-Aided Molecular Design*, 20 (3), 145-157

Tadeusiewicz, R., Gąciarz, T., Borowik, B., Leper, B., 2007, Discovering properties of neural networks using applications in C# language, Polish Academy of Arts and Science, Kraków.

Zhao, Q.-L., Liu, X.-H., Wang, G.-D., Xu, J., An, H.-W., 2005, Neural Network Friction Model for Cold Strip Rolling, *International Journal of Iron and Steel Research*, 12 (4), 35-39.


**PRÓBA NEURONOWEGO MODELOWANIA PROCESU STEROWANIA KRYSTALIZACJĄ ODLEWÓW**


Streszczenie

W artykule zaprezentowano udaną próbę stworzenia neuronowego modelu procesu odlewniczego. Sieć neuronowa po procesie uczenia jest w stanie przewidzieć wybrane efekty chłodzenia odlewu (w postaci liczby kryształów metalu występujących w określonym punkcie odlewu) na podstawie danych dotyczących sposobu chłodzenia odlewu (odlew jest chłodzony wodą, której przepływ sterowany jest według zadawanego wzorca czasowego). W pracy pokazano różne struktury sieci neuronowych używanych do rozwiązywania postawionego problemu oraz przedyskutowano uzyskiwane przy ich pomocy wyniki. Po stworzeniu kilku udanych wersji modelu prostego podjęto próbę stworzenia modelu odwrotnego, to znaczy takiego, w którym na wejście sieci neuronowej podawany jest oczekiwany wynik sterowania (tu – ilość kryształów w odlewie) i oczekuje się, ze sieć na swoim wyjściu wyprodukuje sygnał określający, jak należy sterować chłodzeniem odlewu, żeby ten cel osiągnąć. To zadanie okazało się niemożliwe do zrealizowania, przy czym w artykule dokładnie pokazano przyczyny i okoliczności tej niemożliwości.



*Submitted: March 25, 2008*
*Submitted in a revised form: July 12, 2008*
*Accepted: July 12, 2008*