ANN-based Model of Continuous Casting of Steel

Detailed Study

Igor Grešovnik
Tadej Kodelja

Centre of Excellence for Biosensors, Instrumentation and Process Control

Revision 7, September 2013.
(revision 0: Nov. 2012)

Warning: This is a long document; **please don't send it to a printer**!

Contents:

1 INTRODUCTION

This report provides detailed description of results of the artificial neural network-based modeling of the continuous casting process in the Štore Steel [1] company. This is currently the only part of the process for which the company has access to numerical simulator, and therefore the only part of the process for which a reliable ANN model can be constructed.

The model was constructed on basis of the Robert Vertnik's casting simulator [2]-[4], which was used to produce training data. Construction and analysis of the model was performed by the *NeuralShell*, a flexible multilayered interpreter-centered ANN modeling software *Investigative Generic Library* (*IGLib*). Concepts used in this software have evolved in several years of research work on optimization, inverse analysis and approximation models, which finally culminated in the *IGLib* [5]-[13] library that is used as code base for demanding technical applications. The code depicts on several third party libraries used for the solution of different tasks, which are mentioned in the IGLib web page. Among these, the Aforge.Net [14] library is used as the library for neural networks-related tasks.

The process modeled is briefly described in Section 2. Section 3 contains a short description of the software that was used for generation and analysis of the model. In Section 4 we describe the search for the best ANN model for this example. In the scope of this, we have tested several combinations of ANN architectures and training parameters, which was done by using the IGLib's generic training modules. Section 5 is dedicated to validation of the results and estimation of errors. Section 6 contains a detailed depiction of results of the developed ANN model. This part is the most interesting for practical use and for industrial people involved in the modeled process.

2 PROCESS PARAMETERS AND FINAL MATERIAL PROPERTIES

Process parameters are shown in

Table 1 and output values are shown in Table 2 for the current example.

Table 1: Process parameters (input parameters).

VALUES	NUMBER
Casting temperature (Tcast)	1
Casting speed (v)	1
Temperature difference of cooling water in the mould (DTmould)	1
Cooling flow rate in the mould (Qmould)	1
Cooling flow rate in wreath spray system (Qwreath)	1
Cooling flow rate in 1st spray system (Qsistem1)	1
TOTAL	**6**

Table 2: Output values.

VALUES	NUMBER
Metallurgical length (ML)	1
Shell thickness at the end of the mould (DS)	1
Billet surface temperature at straightening start position (T)	1
TOTAL	**3**

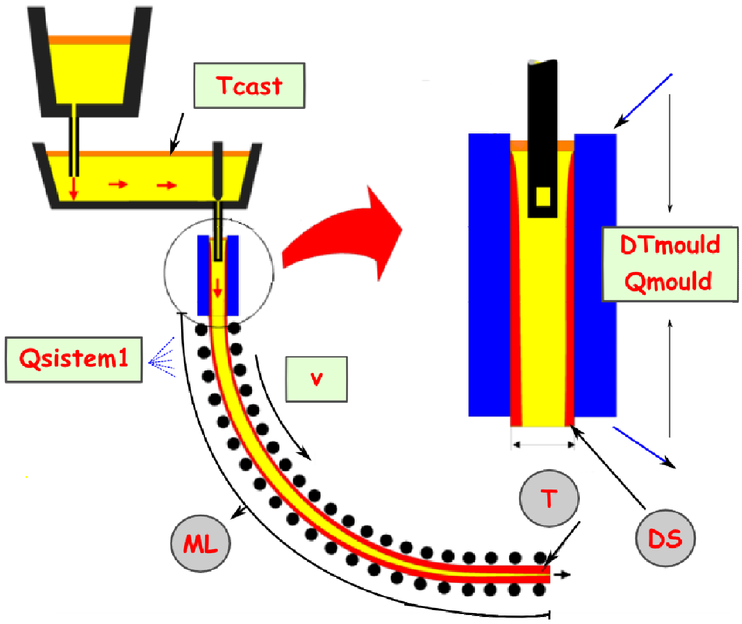

Figure 1: Location of process parameters and output values in the continuous casting process.

3 ANN MODELING SOFTWARE

A software for construction and use of ANN-based models has been developed in the scope of this work. The software was designed as to match the challenges and requirements met when solving this kind of problems. In particular, it must provide good flexibility in designing training strategies, filtering training data, verification of results, testing different network layouts, integration with other software, etc. This is crucial when approximating behavior of steel processing systems with large number of processing parameters. Data obtained from such systems is often inaccurate or even corrupted due to practical limitations in acquisition procedures. Response sampling can not be planned in advance but is accommodated to production schedules in the factory, therefore information available may be deficient in some regions of parameter space in order to obtain good response approximation and therefore verification of results plays an important role.

The software is based on *IGLib* [13], a specialized framework for efficient development of demanding technical applications, and inherits the majority of design paradigms from this framework. Some of these concepts historically follow from the experience gained in development of general purpose optimization software [8]-[12], which is subjected to similar requirements regarding the modularity of software and sufficient flexibility of its user interfaces to accommodate to dynamic environment of multidisciplinary research and development. Among the others, this historical concept is the origin of postulation that a suitable interface for defining and solving complex optimization problems should include a full interpreter capability, through which the built-in functionality, packed in modular libraries, is accessed. Two-level interpreter system developed in IGLib on basis of this experience is utilized in our software and enables rapid reactions to changing demands as new knowledge is gained form the results. IGLib also incorporates design of approximation-based modeling utilities emerging from the development of optimization libraries [10]-[12]. In the scope of the framework, the elaborate object oriented design was supplemented by a number of advanced concepts directed towards rapid application development. These include adherence the concepts of .NET frameworks and extensive use of generic programming. Yet another design feature brought by IGLib is typical arrangement of the framework libraries in a number of levels according to criteria such as platform dependency or license restrictions. The software relies on a number of third party libraries that are mainly distributed under permissive open source licenses.

The Aforge.Net library is used as ANN framework [14]. A convenient characteristics of neural networks is that approximation can be performed in two separate stages (Figure 2). In the training stage, the network is trained by using the sampled response (either measured or calculated by a numerical model). In the approximation stage, trained network is used for all subsequent calculations of approximated response at arbitrary values of input parameters. This gives neural networks an important advantage over other modeling techniques, since the second stage if very fast as compared to the first stage. The software takes full advantage of this feature by separating these stages. This is especially important when performing extensive analyses of the considered process on basis of the developed ANN models, or when incorporating the models in automatic optimization procedures [16],[17].

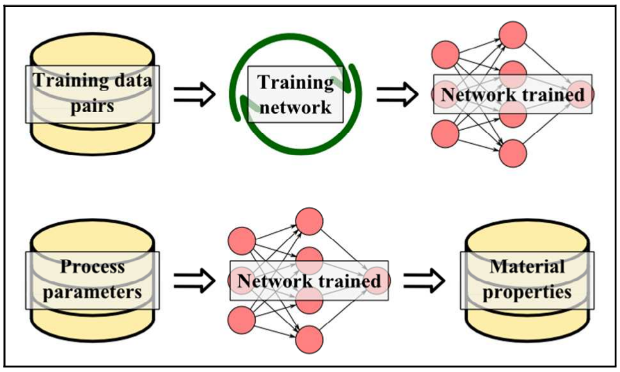

Figure 2: Approximation with neural networks: training a network with presented data pairs (top) and calculation of approximated response with trained network (bottom).