Model Update and Real-Time Steering of Tunnel Boring Machines using Simulation-Based Meta Models

Jelena Ninić, Günther Meschke∗
Institute for Structural Mechanics, Ruhr-University Bochum, Germany

Abstract
A method for simulation-based steering of the mechanized tunneling process in real time during construction is proposed. To enable real-time predictions of tunneling-induced surface settlements, meta models trained a priori from a comprehensive process-oriented computational simulation model for mechanized tunneling for a certain project section of interest are introduced. For the generation of meta models, Artificial Neural Networks (ANN) are employed in conjunction with Particle Swarm Optimization (PSO) for the model update during construction and for the optimization of machine parameters to keep surface settlements below a given tolerance. To provide a rich data base for the training of the meta model, the finite element simulation model for tunneling is integrated in an automatic data generator, for setting up, running and postprocessing the numerical simulations for a prescribed range of parameters. Using the PSO-ANN for the inverse analysis, i.e. identification of model parameters according to monitoring results obtained during tunnel advance, allows the update of the model to the actual geological conditions in real time. The same ANN in conjunction with the PSO is also used for the determination of optimal steering parameters based on target values for settlements in the forthcoming excavation steps. The paper shows the performance of the proposed simulation-based model update and computational steering procedure by means of a prototype application to a straight tunnel advance in a non-homogeneous soil with two soil layers separated by an inclined boundary.

Keywords: Finite Element Method, Neural Networks, Particle Swarm Optimization, real-time prediction, mechanized tunneling, optimization, parameter identification

1. Introduction
Increasing urbanization and the need for environmentally friendly urban transportation and national and transnational high-speed mobility has increased the need for underground transportation systems and safe and efficient construction technologies. Mechanized tunneling is a well-established construction technology, that allows for tunnel advances in a wide range of geological environments, high ground water pressures or small cover depths often met in urban tunneling. It is characterized by a construction process involving complex interactions between the surrounding underground, the groundwater, the support measures at the face and the tail void and the Tunnel Boring Machine (TBM) [1]. Although the continuous increase of efficiency of modern computer technology and the considerable progress in computational structural mechanics have stimulated the development of numerical simulation models in tunnelling since the early 1980s, compared to the larger number of models developed in the context of New Austrian Tunnelling

∗Corresponding author, phone: +49 234 32 29051, fax: +49 234 32 14149
Email address: guenther.meschke@rub.de (Günther Meschke)
Method (NATM) (see, e.g. [2] and references therein), only a relatively small number of fully three-dimensional
numerical models exist for shield tunnelling due to its considerably more complex nature, see, e.g. [3, 4, 5, 6]. These
models are based upon a pre-defined excavation path determined by the mesh discretization and do not allow for the
simulation of the actual steering process accomplished by the elongation of hydraulic jacks and its effect on the actual
movement of the tunnel boring machine in the soil. However, this capability is required if computational methods
should be employed to support the steering process in machine driven tunneling.

A prototype for a process-oriented, three-dimensional, finite-element (FE) model for simulations of shield-driven tun-
nels in soft, water-saturated soil has been proposed by Kasper and Meschke [7] and successfully used for systematic
numerical studies of interactions in mechanized tunneling [8]. Based on the simulation model presented by Kasper
and Meschke (2004) a model for mechanized tunneling has been developed by Nagel et al. (2008), using a more
advanced and flexible software architecture [9]. The model incorporates realistic models for all relevant components
involved in mechanized tunneling (fully and partially saturated soft soils, the shield machine, the segmented lining,
the face support, and the tail void grouting) and closely follows the construction stages on site. Furthermore, it is
characterized by an automatic model generator [10] and an adaptive technique for re-meshing in the vicinity of the
tunnel face [11].

However, although modern programming concepts such as object-oriented modeling and parallelization techniques
[12] are used, the enhanced capabilities and accuracy enabled by such sophisticated 3D FE models have to be paid by
a larger effort for the model generation and a higher computational cost for the actual simulation. If numerical simul-
tion models are to be used directly for the support of the tunneling process during construction, real-time predictions
are required, which evidently is only possible either using massive parallelization or, alternatively, computationally
cheaper meta (surrogate) models. This strategy is adopted in the present paper by proposing Artificial Neural Net-
works (ANNs) to be trained by means of the full-scale simulation model for a certain project section a priori, i.e. in
the design stage to replace the far more expensive 3D numerical simulations for the purpose of real-time predictions
of surface settlements, parameter identification and process optimization.

Due to evident need for rapid computation, robustness and adaptability in all areas of process simulations, in the last
20 years the simulation-based meta models are often used to substitute numerical simulations. Meta models [13, 14]
are used to represent the dependency between the simulation input and output, learning the multiple relations between
different parameters and mapping them to a given output. Different methods are used to provide compact represen-
tations for the simulation model, e.g. regression models [15], ANNs [16], radial basis functions [17], Kriging or
spatial correlation models [13, 18], response surfaces [19], game theory models [20] and proper orthogonal decom-
position [21]. Those models are used for simplifying and interpreting simulation models [14], conducting sensitivity
and what-if analyses as well as optimizing the simulation output [22]. In geotechnical problems, ANNs have been
applied as meta models trained with numerical simulations and used for the prediction of the deformations induced
by geotechnical interventions [23, 24, 25].

Although ANNs have shown excellent performance for mapping given tunneling conditions and parameters to a spec-
ified output parameter (such as settlements), there is still the question of the reliability of data used for the training.
In geotechnical problems, and in particular in mechanized tunneling, information is usually generated in the design
phase based on explorations from a limited number of locations. Consequently, during the construction, the geotech-
nical conditions may differ considerably from the a priori assumptions in the design stage.

This fact raised an interest in the application of inverse parameter identification strategies and optimization algorithms
to geotechnical modeling in order to update the actual material and model parameters at certain stages of the construc-
tion. Since such analyses require a large number of simulations and an automated procedure for execution, sufficiently
fast computer hardware is required. Therefore, automated techniques of inverse analysis for geotechnical processes
have been investigated e.g. in [26, 27, 28, 29]. The difficulty related to methods of inverse analysis like Particle
Swarm Optimization (PSO) is that they need a large number of realizations. This problem can be solved by using
substitute numerical models for evaluation [30].

In the paper, a procedure for a simulation-supported steering of tunnel boring machines is proposed, using ANN as
a meta (or surrogate) model to substitute the full process-oriented model for predictions made in real time during
construction. More specifically, the meta model is used for the update of soil parameters according to monitoring
data by means of inverse analysis using PSO. After the model and/or material parameters are updated according to
the measurements during the tunnel advance, the combined ANN-PSO model is also used for the optimization of the
steering parameters, such as the grouting or the face pressure, according to pre-defined target values for the tolerable
The remainder of this paper is organized as follows: Section 2 motivates the need for numerical steering of the tunneling process and describes the concept, methods and potential applications of the proposed model. Section 3 describes a numerical experiment for mechanized tunneling based on the process-oriented simulation model eKate [9]. In Section 4, methods used for establishing meta model and performing a back-analyses of measurement data are described. Finally, in Section 5 and Section 6, a numerical example of the application of the proposed method characterized by a straight tunnel advance in a heterogeneous soil is presented.

2. Simulation-Supported Real-Time Steering in Mechanized Tunneling

Mechanized tunneling is an industrial processes with a high degree of mechanization, which is characterized by the repeated sequence of individual steps of the process: the thrust of the TBM through hydraulic jacks, the excavation of the soil at the face by the cutterhead and the installation of the next segment ring while the machine is stopped. In order to ensure the stability of the surrounding soil, minimize settlement and prevent the ingress of groundwater into the tunnel, the boring process is accompanied by a range of support measures: the face is supported in the excavation chamber by a pressurized support medium, the soil stands in frictional contact with the shield skin and the annular gap, which is created between the segment lining and ground, is filled with grouting mortar at the same time as the machine is thrust forward (Figure 2). The control of TBMs is supported by a vast amount of monitoring data and machine data, gathered more or less continuously during construction. Decisions on TBM operation depend on the experience of the engineering staff.

Having process-oriented computational models for mechanized tunneling in soft soils [9, 31] available by now, the question arises if such advanced prognosis tools can be applied to support the steering process during construction. As a prerequisite, two problems have to be solved: firstly, the prognosis obtained from the computational model must be available in real time, i.e. in the range of minutes, and secondly, the parameters used within the model must reflect the actual conditions of the project (i.e. geological situation, existing buildings etc.). During the design phase of a tunneling project, geotechnical information is only available from point-wise exploration at discrete boreholes. This information is generally used as the source for determining model parameters for numerical analysis models. Consequently, due to the limited spatial information on the ground properties, the material parameters adopted for numerical analyses in the design stage often do not fully reflect the in situ situation which will be later met during tunnel construction. Since in contrast to the design stage, abundant information from continuous monitoring is available during tunnel construction, these data can be used to update the computational model during tunnel construction. For the model update, i.e. the identification of model parameters according to in situ measurements, a large number of procedures is available (see e.g. the benchmarking of different methods in parameter identification in [32]). In complex geotechnical problems, in general global optimization algorithms are preferred due to their ability to find global optima. However, these algorithms require a large number of realizations, which, due to the relatively complex nature of process-oriented finite element models for tunnel advance, is connected with large computing times - even if massive parallelization is employed - in the range of hours. In order to perform such back analysis in real time, the computing time should be in the range of minutes. We propose to use computationally cheap meta (or surrogate) models instead of the original finite element model. Such meta models substitute the original computational model by providing, for a certain range of parameters, identical results. Project specific surrogate models are a priori (i.e. already in the design phase of a tunnel project) generated in a training phase for a certain set of parameters. The concept of using meta models for back analysis of process oriented numerical model for mechanized tunneling according to the in situ state in real time is depicted in Figure 1. For the meta model described in this paper, an advanced FE simulation model for mechanized tunneling [9] was used as the basis for the training procedure, which requires to compute a large number of FE simulations for a given range of parameter (“numerical experiment”). To minimize the required manual intervention, the generation of numerical experiment is performed using a Data generator, which automates the complete process of setting the (large number of) simulations, pre- and postprocessing of simulation data and creating and validating the meta model. A large set of data, obtained from the numerical models of consistently created hypothetical cases of the tunneling process, is used for the training of the meta model. These numerical experiments provide the model response for a certain combination of material and model parameters that have a relevant influence on the predicted quantities, e.g.
on settlements. This set of parameters is determined a priori in the design stage according to sensitivity studies. The meta model is trained using a back-propagation neural network, whereas the architecture and the network-related parameters are optimized using a global optimization approach. During tunnel construction the meta model is used for the identification of model parameters based on monitoring data and for real-time predictions of surface settlements. Having a reliable meta model with parameters updated according to monitoring data, the ANN-PSO model is used again to support the steering of the tunnel machine. To this end, the process parameters of the tunneling machine (e.g. grouting and tail void pressure) are optimized according to defined steering targets, e.g. minimal surface settlements.

3. Numerical Experiment

A number of numerical simulations of the tunnel excavation is performed a priori in the design stage for a certain portion of the tunnel project to provide a data base for the training of the meta model. This set of FE simulations is called “numerical experiment” in the following. As output, the numerical experiment provides a realistic, process-oriented and holistically conceived representation of the consequences of the tunneling process (e.g. the surface settlements) considering all relevant parameters and with the variable geological boundary conditions in parallel to the actual tunneling work, for a certain range of (predefined) parameters. To this end, a process-oriented numerical model for mechanized tunneling (ekate) [10] was employed to simulate the shield tunneling process. A Python script called Data generator was used for automatically setting up the numerical experiment and performing pre- and postprocessing of the data. These two components (ekate and the data generator) are described in more detail in the following subsections.
3.1. Simulation Model for Mechanized Tunneling in Soft Soils

\textit{ekate} is an advanced simulation model for mechanized tunneling built in the object-oriented software framework KRATOS [33]. The model considers all relevant components of the tunnel construction process (TBM, soil and groundwater conditions, lining, steering via the hydraulic jacks, tail void grouting and various types of face support) and their (time-dependent) interactions (Figure 2). The soil is modeled as a three (two) phase material for partially (fully) saturated soils [34]. Two elastoplastic models are available for the modeling of the inelastic response of soft soils: the Clay and Sand model [35] and the Drucker-Prager model.

![Figure 2. a) Tunnel Boring Machine; b) numerical model for shield tunneling (\textit{ekate})](image)

The TBM is modeled as a distinct deformable body advancing through the soil by means of hydraulic jacks, represented by truss elements. The elongation of the hydraulic jacks is controlled by an automatic steering algorithm to advance the shield along the prescribed designed alignment similar to [7]. The TBM is interacting with the surrounding soil through a surface-to-surface contact algorithm [36], which allows for a smooth advancement of the machine and interactive steering also for curved tunnel alignments. After each TBM advance step, modeled by means of deactivation of soil elements in the excavation zone and pushing the shield forward, a lining ring is installed as a permanent support of the soil. The annular gap between the segmented lining tube and the excavation boundary is refilled with grout, modeled as a fully saturated two-phase material with a hydrating matrix phase, considering the evolution of stiffness and permeability of the cementitious grout [37]. To ensure stability of the tunnel face due to distortions caused by the excavation process and to reduce ground loss behind the tapered shield, face support pressure and grouting pressures are applied, respectively.

3.2. Automatic Data Processing

An automated \textit{Data Generator} for the training phase of the meta model (i.e. for the “numerical experiment”) was created for setting up a suitable set of simulation models covering a certain range of relevant parameters (e.g. material and topological parameters related to the soil, operational parameters related to the temporary support of the soil at the face and in the steering gap of tunnel boring machine) and for providing the desired data set with uninterrupted workflow. The \textit{Data Generator} is formed by a number of Python routines, reading all parameters of the numerical experiment from a single input file and automatically creating the geometry and the FE mesh of desired tunnel models, writing simulation scripts, applying material properties, distributing the execution of simulations to a prescribed number of computers and finally, post-processing the data of the investigated tunneling cases in a suitable form for the subsequent evaluation.

To enable automated generation and execution of numerical analyses involved in the training of the meta model, the simulation of the tunnel advancement is divided into three basic objects: the geometry and the finite element discretization, the material properties and the process parameters. (Figure 3). The geometry and the finite element
The mesh of the tunnel is created using the *Scaled Model Generator*, taking geometrical attributes like diameter of the tunnel, overburden, thickness of the lining and grouting, shape of the machine, etc. as input. The material parameters of the different soil layers are defined in a separate script and subsequently applied to the previously created FE mesh using the *Material Properties Utility*. This script performs a global search over the complete domain and applies the material properties at the integration points of elements belonging to prescribed polygons. The material properties script is created according to user-defined variables such as the topology of the soil layers, type of the constitutive model and the respective model and material parameters. Finally, the process parameters (face and grouting pressure, advance speed, etc.) are defined in a simulation script, which reads all previously defined data and invokes KRATOS to solve the resulting simulation problem.

Figure 4 outlines the workflow of the generation of the meta models, the execution of the simulation and the postprocessing performed by the *Data Generator*. Reading an input file, the *Data Generator* generates a set of simulations with the prescribed parameters. The input file for the numerical experiment contains the following information:

- Geometry, material and operational parameters
- Number of variable parameters ($N_p$)
- Minimum and maximum values of the variable parameters
- Number of intervals between minimum and maximum values ($N_i$)
- Number of simulations executed in parallel ($n$)
- Portion of data used for training, testing, validation (and synthetic measurement)

Reading the input file containing geometry, material and process parameters, the *Data Generator* automatically creates $N_i^{N_p}$ simulation sets and executes $n$ simulations in parallel using a shared memory system based on openMP. For each executed simulation, an output file is created using the *Output Utility* [38]. The output file contains selected information about the model in the observation points ($S_1$, $S_2$ and $S_3$). In the *Data Processing* algorithm, the resulting data is filtered for the meta-modeling process so that noise from the data is eliminated. Considering the different dimensions of different parameters, all input-output pairs are mapped to the
interval (0.1, 0.9) using a Data Normalization algorithm. For a parameter $V$, the normalized value $V_{\text{norm}}$ is obtained from

$$V_{\text{norm}} = \frac{V - V_{\text{min}}}{V_{\text{max}} - V_{\text{min}}} (\bar{V}_{\text{max}} - \bar{V}_{\text{min}}) + \bar{V}_{\text{min}}$$ (1)

$V_{\text{max}}$ and $V_{\text{min}}$ are the maximal and minimal value of the variable $V$, and $\bar{V}_{\text{max}}$ and $\bar{V}_{\text{min}}$ are the maximal and minimal values of the variable $V$ after normalization, defined as 0.1 and 0.9. After the complete data set has been processed, the data is split into data for training, testing and validation of the meta model, according to prescribed portions (see Figure 4).

4. Meta Model for Mechanized Tunneling

For the purpose of real-time predictions of surface settlements a meta model is employed to substitute computationally demanding 3D numerical simulations. To this end, Artificial Neural Networks (ANN) are proposed, taking the variable parameters $i$ as input variables and the predicted tunnel-induced settlements as output data. For the training of the ANN, a Particle Swarm Optimization (PSO) is used. ANN and PSO are described in what follows.

4.1. Artificial Neural Networks

Artificial Neural Networks (ANNs), initially introduced by McCulloch and Pitts, are a form of artificial intelligence that attempts to mimic the behavior of the human brain and neural system, being capable of learning the pattern associated with a large body of data [39]. Following the initial proposal, numerous researchers provided significant contributions for the advancement of this method and introduced new learning techniques [40, 41]. The most popular ANN method, the Back Propagation Neural Network, with a typical architecture illustrated in Figure 5, was first
proposed by Rumelhart, Hinton, and Williams [42]. This method can be used to learn the weight values and yield values. During the learning process, the gradient descent method is used to change the weight values and yield values as quickly as possible to reduce the error.

\[
\begin{align*}
\text{Input layer} & \quad x_1, x_2, x_3, \ldots, x_n \\
\text{hidden layer} & \quad w_{11}, w_{12}, w_{22}, \ldots, w_{1n}, w_{2n} \\
\text{output layer} & \quad o_1, o_2, o_3, \ldots, o_m
\end{align*}
\]

Figure 5. Basic structure of a Backpropagation Neural Network (BP-ANN)

Considering all input nodes \( x_i \) (\( i = 1, \ldots, n \)) and output nodes \( o_k \) (\( k = 1, \ldots, m \)) for a given number of patterns \( p_n \), a basic back propagation algorithm consists of two steps:

- **Feed-forward step**
- **Back propagation step**

The first step is calculated according to Equation (2), where \( w_{ij} \) are connecting weights from the input to the hidden layer, \( w_{ij} \) are connecting weights from the hidden to the output layer and \( f(\bullet) \) is the activation function used to transform the incoming values and transfer them to the next layer:

\[
o_k(x, w) = f \left( \sum_{j=1}^{m} w_{jk} f \left( \sum_{i=1}^{n} w_{ij} x_i \right) \right).
\]

(2)

One of the most popular activation functions for back propagation networks is the sigmoid function, a real-valued function \( f : \mathbb{R} \rightarrow (0, 1) \) defined as:

\[
f(x) = \frac{1}{1 + e^{-cx}}.
\]

(3)

All output values are compared with the respective target values \( t_k \). A measure of the error between the output and target values is defined as:

\[
E = \sum_{k=1}^{m} (o_k - t_k)^2.
\]

(4)

The objective of the second step – the learning process – is to adjust the free parameters (the synaptic weights of the network) to minimize the error given in Equation (4). This problem can be solved with a gradient descent method, where the gradient of \( E \) with respect to the input quantities is calculated and weights are adjusted incrementally:

\[
\Delta w_{ij} = -\gamma \frac{\partial E}{\partial w_{ij}} \quad \text{and} \quad \Delta w_{jk} = -\gamma \frac{\partial E}{\partial w_{jk}}
\]

(5)

\( \gamma \) is the learning rate, which is explained in Subsection 4.3. Note that the learning process can also be accomplished using PSO as a learning method, where the synaptic weights of the network are unknown particle positions, and the fitness is a function of the learning error \( E \).
4.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an evolutionary computational algorithm suggested by Kennedy and Eberhart in 1995 [43]. The method was developed to simulate social behavior associated with the movement of organisms in a bird flock or fish school. The system is initialized with a population of random solutions and searches for optimima by updating subsequent generations. The potential solutions, called particles, “fly” through the problem space by following the current optimum particles. Each particle belongs to a swarm and has two properties: velocity ($v_{ij}$) and position ($x_{ij}$). The particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) and achieves the particle-best value $p_{best} (x_{ij}^{p_{best}})$. If a particle takes the complete population as its topological neighbors, the best value is a global best $g_{best} (x_{ij}^{g_{best}})$. Kennedy and Spears [44] improved PSO by introducing the inertia weight ($w$) into the basic update rule to reduce the particle speed when particle swarms are searching a large area. The new velocity and position of the particles is updated in each iteration using the following equations:

$$
\begin{align*}
    v_{ij,t+1} &= w_{ij} + \phi_1 r_1 (x_{ij}^{p_{best}} - x_{ij}) + \phi_2 r_2 (x_{ij}^{g_{best}} - x_{ij}) \\
    x_{ij,t+1} &= x_{ij} + v_{ij,t+1}.
\end{align*}
$$

In Equation (6) $r_1$ and $r_2$ represent random numbers uniformly distributed in [0, 1], and $\phi_1$ and $\phi_2$ are so-called cognition and social learning factors.

BP-ANNs described in Subsection 4.1 can be combined with PSO and used to evaluate a Fitness function. In this case, the Fitness relates to the learning error $E$ (Equation 4). It is defined as:

$$
    \text{Fitness} = \frac{1}{\sum_0^{n} E + TOL}
$$

where TOL is a tolerance to avoid singularity of Equation (7). Particles are moving towards the positions leading to the maximum possible values of the Fitness function.

4.3. Optimized Learning

In the present work, the PSO was successfully combined with the backpropagation algorithm for the training of ANNs in order to optimize the architecture of the network (number of nodes $j$ in the hidden layer) and the learning rate $\gamma$ in Equation (5), which controls the speed of the learning process. Figure 6 describes the algorithm for the optimization of the ANN architecture and the speed of learning. As the variables to be optimized, the number of nodes in the hidden layer ($j$) and the learning rate ($\gamma$) are chosen. Therefore, the dimension of the swarm $d$ is set to 2.

The Fitness of the PSO was evaluated using the success of ANN training, i.e. the total error for all trained patterns.

It should be noted, that the PSO could also be used as a learning algorithm for ANNs if the synaptic weights $w_i$ of the network are chosen to be unknown particle positions [45]. However, for the large number of unknowns involved in the presented example in this paper, this algorithm turned out to be inefficient and therefore was not used.

4.4. ANN-PSO based Meta Model for Simulation-Based Steering of the Tunneling Process

Figure 7 shows the flow chart for the interactive steering support of the tunneling process based on the proposed ANN-PSO based meta model. A prototype application is be demonstrated in Section 5, where, however, actual in situ monitoring data and measurements are replaced by synthetic project data and synthetic measurements generated from a forward analysis using the finite element model for mechanized tunneling.

In order to enable simulation-based steering of the tunneling process, the meta model providing settlement prediction for a certain given set of input parameters has to be established first. This procedure is described in the first two steps of Algorithm 7, where the simulation model for mechanized tunneling (Section 3.1) is used to create a master shield tunneling case (Section 5.1) which is then integrated into the automated workflow for a systematic data generation and postprocessing (Section 3.2). The data obtained from the numerical experiment containing information on the model geometry, material parameters and TBM operational parameters is used to train BP-ANNs (Section 4.1) with different architectures. The optimal model obtained from the previous step is the new meta model which is able
Figure 6. Algorithm for optimized learning using a combination of BP-ANN and PSO to determine optimized values of the learning rate \( \gamma \) and the number of nodes \( j \) in the hidden layer.

This fast meta model can now be used during construction. However, if the predicted settlements, which are based upon geotechnical data from the design stage, show a mismatch with the in situ measurements, the model parameters are apparently not representing the real situation and must therefore be updated during tunnel advance. In the proposed ANN-based meta model, the PSO (Section 4.2) is used as an optimization algorithm for parameter identification. In the parameter identification process, the TBM operational parameters are kept fixed and material parameters of the soil are updated such that the predicted settlements match the field measurements for the given project data. The proposed ANN-PSO algorithm allows to flexibly chose the number and the type of the parameters which need to be updated and to keep the parameters within a plausible range.

Having an updated meta model at hand, which suitably represents the in situ state of the soil conditions in the vicinity of the current location of the TBM, this model can now be used for the support of the steering of the TBM. To
For this purpose, the same ANN-PSO algorithm can be used to control surface settlements by searching for the values of the TBM operational parameters that reduce the settlements to a desired value. Therefore, in this phase of the optimization process, the material parameters are treated as known and fixed parameters, while an arbitrary number of TBM operational parameters is subject to modification during the simulation-based steering process.

5. Numerical Example 1: Mechanized Tunneling in Homogeneous Soil

As a first numerical example for the generation of meta models and the identification of soil parameters in mechanized tunneling, tunneling advance in soft, homogeneous soil along a straight path of 48 m is considered. First, the process-oriented finite element model used for the training of the meta model is described, followed by the generation of a suitable architecture of the ANN-based meta model and finally the description of the parameter identification procedure.

5.1. Numerical Simulation for the Training of the BP-ANN

The meta model is trained by means of 625 simulations created using the process-oriented finite element model eka. The finite element discretization and geometry of the investigated portion of the tunnel is shown in Figure 8a; Figure 8b shows the distribution of the surface settlements when the TBM face crosses the measurements section, which is characterized by three measurement points. The example is concerned with the excavation of a tunnel with...
a diameter $D$ of 8 m and an overburden of $1D$. The investigated portion of the tunnel is assumed to be constructed by a sequence of 32 steps with 1.5 m length each. The total dimensions of the model are 48 m length, 85 m width and 55 m height, taking the symmetry of the geometry, material properties and boundary conditions assigned to the model into account. The chosen constitutive laws and the respective parameters of the model components are summarized in Table 1.

| Parameters                      | Soil         | Lining      | Grout       | Machine     |
|---------------------------------|--------------|-------------|-------------|-------------|
| Young Modulus - $E$ [GPa]       | 0.01-0.1     | 30          | 5           | 210         |
| Poisson ratio - $\nu$ [-]      | 0.3          | 0.2         | 0.45        | 0.15        |
| Density - $\rho$ [Kg/m$^3$]    | 2000         | 2500        | 2000        | 7600        |
| Porosity [-]                   | 0.2          | -           | 0.2         | -           |
| Cohesion [KPa]                 | 75           | -           | -           | -           |
| Friction angle - $\varphi$ ['']| 36           | -           | -           | -           |
| $K_0$ - $\phi$ [-]             | 0.5 - 1.5    | -           | -           | -           |
| Permeability [m/s]             | 0.0044       | -           | 0.0001      | -           |

Table 1. Material parameters of the simulation model

For establishing the data set for the generation of the meta model, the effect of two material parameters – the Young’s Modulus $E$ and the ratio of the horizontal and vertical initial stresses in the soil $K_0$ (see Table 1) – on the induced settlements was investigated. In the previous subsection, we mentioned that the stability of the tunnel face and the induced settlements are controlled by means of the operational parameters of the TBM, such as the face support pressure $p_f$ and the grouting pressure $p_g$. The variation of these two parameters was also taken into account for the data generation to enable the process optimization in further studies. The investigated range of operational parameters is $p_f = 120$ KPa – 270 KPa and $p_g = 62.5$ KPa – 187.5 KPa.

Three monitoring points ($S_1$, $S_2$ and $S_3$) located at the monitoring section on the soil surface were chosen to track the vertical displacements in the course of the excavation in a range from 18 m ahead of the tunnel face to 18 m behind.
the tunnel face. $S_1$ is on the axis of symmetry and $S_2$ and $S_3$ have an offset of 1.5D and 3.5D in transverse direction, respectively. This allows to assess the influence of different geometrical, soil and process parameters on the surface settlements.

For the automated generation of the meta model and its training based on the finite element model according to Section 3.2, two parameters related to the soil properties and two parameters related to the advancement process of the TBM are taken as variable parameters to be updated during construction (i.e. $N_p = 4$): the modulus of elasticity $E$, the coefficient of earth pressure at rest $K_0$, the grouting pressure $p_g$ and the face pressure $p_f$. To provide a sufficient resolution for the interpolation quality of the ANN meta model, the number of intervals between minimum and maximum values is set to $N_i = 5$. A total number of 625 simulations was executed in sets of 5 simulations in parallel. One simulation required $\approx 8$ hours for computation, the complete numerical experiment took $\approx 1.5$ months, resulting in a set of 45,255 data values for the surface settlement at the measurement points $S_1$, $S_2$ and $S_3$ (Figure 5.1). This data was then divided into data sets for training (60%), testing (15%) and validation (15%) of the meta model. Since the presented numerical application is not related to a specific tunneling project, measurement data was substituted by computed settlements, for which 10% of the computed data was used to serve as “synthetic measurements”. This is sufficient to demonstrate the effectiveness of the proposed parameter identification and process optimization method for mechanized tunneling.

5.2. Selection of Meta Model

Since the data used for training ANNs usually contains a certain amount of noise, over-training of the data would result in a fitting error in the training and thus decrease the prediction capability. To avoid this overfitting problem, the meta model is established on the basis of the training and testing data set, where the training set is used to establish the value of connection weights and the testing set to check the optimality of the trained weights. As a measure of success of ANN training and testing, the relative root mean square error ($r_{RMSE}$) between the the “synthetic measurements” and the computed settlements,

$$r_{RMSE} = \sqrt{\frac{1}{p_n} \sum_{k=0}^{p_n} \left( o_k - t_k \right)^2},$$

is used.

![Structure of two different BP-ANN architectures used as meta model: a) Model M1 with 6 neurons in input layer and one neuron in output layer, b) Model M2 with 5 neurons in input layer and three neurons in output layer](image_url)
Two different architectures of BP-ANNs are compared (Figure 9). The model $M_1$ is characterized by 6 nodes in the input layer for $E$, $K_0$, $p_g$, $p_f$, $d_f$ and $d_i$ were $d_f$ denotes the position of the machine with respect to the monitoring point in the direction of the TBM advance, and $d_i$ ($i = 1, 2, 3$) denotes the transversal distance of the monitoring point from the tunneling axis, referring to the position of points $S_1$, $S_2$ and $S_3$. The model $M_1$ has one output node $o_i$, which, similar to $d_i$ refers to the monitoring points. The training set of this model contains 27,153 input patterns. The second model $M_2$ has 5 nodes in the input layer and 3 nodes in the output layer, because $d_i$ was omitted as input and the settlements at the monitoring points are provided in separate output nodes. Consequently, $M_2$ requires only 9,048 input patterns.

Both models were trained using the ANN-PSO algorithm as described in Subsection 4.3. The training procedure was performed in 20 iterations, where $M_1$ was initialized with 15 and $M_2$, due to the larger number of synaptic weights and the smaller number of input patterns for the $M_2$ model, with 25 swarms. The training was performed for 5,000 training cycles for each swarm and each iteration. The objective was to optimize the architecture of the network as well as the learning rate $\gamma$ such that training and testing errors are minimized. In this training algorithm, one hidden layer was used for all networks, a choice that has been proven as sufficient by Cybenko [46].

Figure 10a shows the evolution of the error measure $rRMSE$ of the models $M_1$ and $M_2$ during the iterative training procedure, in which the number of neurons in the hidden layers ($j$) and the learning rate $\gamma$ are optimized with regards to global best positions of particles ($g_{best}$).

According to this plot, model $M_1$ has a better performance compared to $M_2$ both for the training and the testing set. However, for both models, the solutions were achieved quickly after the first two iterations, and improved only slightly afterwards. The global best solution for model $M_1$ was obtained in the eighth iteration for 24 neurons in the hidden layer and a learning rate of $\gamma = 0.207$. The optimal model was validated using the validation set as shown in Figure 10b. The $rRMSE$ of the validation set was 2.34% which confirms that the model has very good predictive capability.

Figure 11 shows a comparison between the longitudinal and greenfield settlements predicted by the meta model with the synthetic measurements generated from the finite element model.

5.3. Parameter Identification

During the tunnel advance, the soil parameters determined in the design stage of the project (Young’s Modulus $E = 88.7$ MPa, and $K_0 = 1.25$) are available as initial values. In Figure 12a, the solid gray line shows, however, that
Figure 11. Comparison of virtual measurements with predicted settlements for: a) $S_1$, $S_2$ and $S_3$ monitoring points during TBM advance; b) Greenfield settlements during TBM advance; $M^{20}_1$ – predictions of the meta model, $N_M$ – results of the numerical model.

Figure 12. Parameter identification: a) Comparison between measurements, initial predictions according to design parameters ($E = 87.5\text{MPa}$) and parameters updated according to (synthetic) in situ measurements during tunnel construction ($E = 10.1\text{MPa}$ and $K_0 = 1.0$); b) Performance of the PSO-based parameter identification procedure for different initial values of $E$ and $K_0$; using this initial set of parameters leads to a significant mismatch with the (synthetic) measurements (Figure 12a, solid black line). Therefore, the meta model must be updated by means of the parameter identification procedure described in Section 4.4. In this example, the parameter identification of the meta model was performed using "synthetic measurements" mimicking real monitoring data gathered during tunnel advance in a tunneling project. Figure 12b shows the performance of the model update algorithm when identifying the parameters $E$ and $K_0$ such that predictions match the site measurements. Starting from different initial values of $E$ (88.7 and 55.0 MPa) and $K_0$ (0.5 and 1.45), after $\approx 25$ iterations, the algorithm converges to the correct solutions ($E=10\text{MPa}$, $K_0=1.0$). It should be noted that the
back analysis process requires only a few seconds to be executed on a standard PC, since the evaluation of the Fitness function is based on the previously trained ANN.

6. Numerical Example 2: Model Update and Steering Support for Mechanized Tunneling in Heterogeneous Soil

In this example, we demonstrate the ability of the proposed method for the prediction and parameter identification in a geologically more complex situation as compared to the first example in Section 5. Furthermore, the developed meta model is now also used to determine optimal process parameters with regards to limiting the expected tunneling-induced settlements. The present application is characterized by a straight tunnel passing through two layers of soil (Soil$_1$ and Soil$_2$), with unknown orientation of the boundary between the layers with respect to the tunnel axis (Figure 13). In this example, it is assumed that the angle of inclination $\Theta$ of the interface between the two layers with respect to the tunnel axis is not reliably determined in the design stage, and that this parameter therefore must be updated during tunnel construction by means of available in situ measurements. The two soil layers are assumed to have different Young’s moduli $E_1$ and $E_2$. To focus on the influence of the soil stiffness, the remaining soil geotechnical parameters are assumed to be identical for both layers: friction angle $\varphi = 36^\circ$, Poisson ratio $\nu = 0.3$, cohesion $c = 100$ KPa, porosity 0.2 and permeability 0.0044 m/s. The range of parameters investigated in this example is given in Table 2.

Figure 13. Numerical application to mechanized tunneling in heterogeneous soil: Longitudinal section and tunnel alignment; soil layers Soil$_1$ and Soil$_2$ with angle of inclination $\Theta$ of the interface between the two layers; $S_1$, $S_2$ and $S_3$ are measurement points

Figure 14 shows the finite element model used for the training of the meta model, exploiting axial symmetry. The diameter $D$ of the tunnel is 8 m, and an overburden of 1 $D$ is assumed. The tunnel analysis is performed within 48 exact steps. Three monitoring sections, where the maximum settlements are monitored in points $S_1$, $S_2$ and $S_3$ (Figure 13) are assumed in this example. The first and third monitoring sections have a distance of 21 m from the model boundaries to ensure that the boundary conditions have no influence on the analysis results. The properties of lining, grouting and machine are the same as in example 1 (see Table 1).

To provide the data for the training of the meta model, again the Data Generator was used to set up the numerical experiment, perform the pre and postprocessing of the simulations and to control the ANN training procedure. For this example, three material parameters (Young’s modulus of the two layers $E_1$ and $E_2$ and the slope of the interface
Figure 14. Numerical application to mechanized tunneling in heterogeneous soil: Finite element model used for training of the meta model including vertical tunneling-induced vertical displacement and three measurement points $S_1$, $S_2$ and $S_3$.

| Parameters                     | Range          |
|-------------------------------|----------------|
| Young’s Modulus of the Soil1 - $E_1$ [MPa] | 30 – 60        |
| Young’s Modulus of the Soil2 - $E_2$ [MPa] | 70 – 100       |
| Slope of the layers – $\Theta$ [°] | 0 – 90         |
| Grouting pressure – $p_g$ [KPa]   | 75 – 225       |

Table 2. Numerical application to mechanized tunneling in heterogeneous soil: Range of parameters for the training of the meta model.

between the two soil layers $\Theta$) are assumed to need an update during tunnel construction due to insufficient geological information available in the design stage. The investigated range of the parameters is given in Table 2. The resulting set of 500 simulations was then executed on a shared memory system, using four threads, running six simulations in parallel. This numerical experiment took 44 days in total, resulting in a total set of 61,260 data values for the surface settlements in the three measurement points. This data set was divided into training (50%), testing (15%) and validation (15%) set, while 10% was again used as synthetic measurements.

The meta model was trained using the algorithm in Figure 6. The architectures of the BP-ANN models $M_1$ and $M_2$, the initialization of the algorithms and number of training cycles are identical to the previous example in Section 5. Model $M_1$ has six input nodes: The Young’s Modulus of the two soil layers $E_1$ and $E_2$, the angle of inclination $\Theta$, the grouting pressure $p_g$, the time step $t$ and the position of the measurement w.r.t. the middle cross section. The model has one output node: the settlement at the respective point $S_1$. Model $M_2$ has five input nodes, coinciding with the first five input nodes of the model $M_1$, and three output nodes: the settlements for three measurement points $S_1$, $S_2$ and $S_3$.

Figure 15a shows the performance of models $M_1$ and $M_2$ during training. The PSO used for the global optimization of the architecture and the learning rate during the training of the meta model again demonstrated good performance with fast convergence within the first two iterations. The best solution was achieved in the 14th iteration for model $M_1$, where it has been found that the optimal number of hidden layers is 22 and the learning rate $\gamma = 0.1014$. 

This resulted in an $rRMSE$ of 4.6% in the training set and 5.9% in the testing set. It is noted that, for the same training conditions, the performance of the training in this geologically more complex scenario, which includes the identification of a topology variable (the angle $\Theta$) was slightly worse compared to the previous example, characterized by a homogeneous geological configuration. This is attributed to the more complex distribution of the settlements as functions of space, time and the variable parameters: The choice of measurement points with offset in longitudinal direction causes an offset of the peaks of the settlement functions, and the variable inclination of the boundary between the two soil layers additionally causes a shift of peaks for each settlement pattern. However, as the comparison between the output from the trained meta model and the original finite element model in Figure 15b shows, the predictive capability of the trained BP-ANN is satisfactory.

6.1. Parameter Identification and Determination of Optimal Process Parameters

In this subsection, we illustrate both steps of the proposed concept for monitoring- and simulation based steering in mechanized tunneling: the identification of unknown geological conditions for given in situ measurements during tunnel construction and the optimization of process-related parameters to reduce tunnel-induced settlements to an admissible value. For this example, we used numerical simulations to create a set of synthetic measurements, i.e., data that has not been used for training or validation of the model.

The ANN-PSO algorithm described in Section 4.4 was used for the back analysis of material properties of the layers ($E_1$, $E_2$) and the angle of inclination between the layers ($\Theta$) based on (synthetic) measurements available at the monitoring points $S_1$, $S_2$ and $S_3$. In Figure 16a, solid lines indicate the (synthetic) measurements, and the dashed lines represent the predictions from the updated meta model. This shows that the updated model predictions match the measurement plot qualitatively and quantitatively well, reaching an overall error of $rRMSE = 4.9\%$. Figure 16b shows that all material and topological parameters were identified within $\approx 30$ iterations. The identification was performed for different initializations of the problem, numbers of swarms (20–50) and numbers of iteration. Convergence to the same value within less than 50 iterations was observed for all initial conditions. The computations were carried out within few seconds on a standard PC. If the predicted settlements are larger than the values tolerated by the design requirements, we now use the same procedure to determine process parameters (in this case the grouting pressure $p_g$) which lead to a reduction of tunneling-induced settlements to an admissible limit. For the given numerical example, the acceptable maximum surface settlement was assumed as 1 cm. Having identified the conditions in the soil according to the (synthetic) monitoring data, the model parameters $E_1$, $E_2$ and $\Theta$ can now be used as reliable
Figure 16. Identification of geological conditions using ANN-PSO: a) Comparison between (synthetic) measurements and predictions from the meta model after parameter identification b) Performance of PSO during parameter identification (identified parameters $E_1$, $E_2$ and $\Theta$)

Figure 17. Process optimization using ANN-PSO: Determination of the level of the grouting pressure such that settlements do not exceed a limit value of 1 cm basis for the prediction of the settlements in the subsequent excavation steps, if the geological situation does not change abruptly. The update meta model can now be used for the back analysis of process parameters, with the the grouting pressure considered as the target function for the optimization with a maximum value of 10 mm (see Figure 7). Figure 17b illustrates the convergence of the grouting pressure from the initial value of 80 KPa to the value of 187 KPa, leading to settlements less than 10 mm as shown in Figure 17a.
7. Conclusions

In this paper, a prototype concept for the simulation- and monitoring-based support of the steering of tunnel boring machines in real time was proposed. It consists of two stages: firstly, an efficient method for updating a computational model for mechanized tunneling according to monitoring data obtained during construction, and secondly, the back analysis of optimal operational parameters according to prescribed limits for the target settlements. Due to the requirement for real-time response, meta models, providing response within a few seconds, are employed for the steering support during construction, substituting the computationally demanding, process oriented finite element model for mechanized tunneling. This finite element model is used, however, in the design stage for the training of the meta model based upon the available information on the geological conditions.

To this end, a fully automated computational procedure for the generation of simulation-based meta models with capabilities of prediction, parameter identification and process optimization was developed. Backpropagation Artificial Neural Networks (BP-ANNs) were employed to generate the meta model. The BP-ANN has been trained by means of a process oriented computational model for mechanized tunneling (ekate). The meta model was created using PSO-optimized ANN learning by means of the predictions of the finite element model ekate for a given set of material and model parameters as well as operational parameters related to the steering of the TBM.

Selected numerical applications have shown that it is possible to predict the spatio-temporal evolution of settlements with high reliability with the proposed simulation-based ANN model, even for a more complex geological conditions characterized by two soil layers with an inclined boundary. For the selected portion of the tunneling project and a fixed set of relevant in- and output parameters, the meta model fully substitutes the finite element model during construction.

The trained meta model was used for updating the model parameter during construction on the basis of information obtained from monitoring. For the inverse analysis, the Particle Swarm Optimization (PSO) method was used. In two numerical examples (using synthetic measurements instead of monitoring data) the performance of the PSO for the inverse identification of material parameters of the soil as well as topology related parameters such as the inclination angle of the boundary between two soil layers was demonstrated. On average, less than 30 iterations were required to identify the unknown parameters. Furthermore, the choice of PSO for performing the global optimization during the inverse analysis allows to easily consider physical limits for parameters to be identified.

Finally, a procedure for the simulation-based steering of the tunneling process was proposed based upon the updated meta models. Using inverse analyses, operational parameters related to the steering of the TBM (e.g. grouting or the face pressure), are determined such that certain target values (e.g. surface settlements) to be expected within the next few excavation steps are kept below an acceptable limit. For the inverse analyses of the selected operational parameters, again a combination of ANNs and the PSO method was used successfully. It was shown that only a few (10-15) iteration steps were necessary to identify the optimal steering parameters with sufficient accuracy.

The advantage of this method is that using the meta model for the evaluation of the fitness function in the PSO algorithm, the identification process is executed within a few seconds on a standard PC or possibly also on mobile devices and therefore is well suited to support decisions related to the selection of steering parameters during tunnel construction.

It should be noted, that the proposed procedure is based on the assumption that the soil condition does not change abruptly during further advancement of the tunnel boring machine from the current location to the next excavation step. However, due to the minimal computing time required for the determination of operational parameters, the proposed method is well suited for the consideration of uncertainties of the soil parameters ahead of the tunnel face. This allows to provide a range of suitable operational parameters, instead of a deterministic value obtained from the proposed (deterministic) approach. The extension to account for uncertain soil parameters will be the subject of future research.

8. Acknowledgements

Financial support was provided by the German Science Foundation (DFG) in the framework of project C1 of the Collaborative Research Center SFB 837 “Interaction modeling in mechanized tunneling”. This support is gratefully acknowledged.
References

[1] B. Maidl, M. Herrenknoechl, L. Anheuser, Mechanised Shield Tunneling. Ernst und Sohn, 1996.
[2] G. Beer (Ed.), Numerical Simulation in Tunneling. Springer, Wien - New York, 2003.
[3] S. Bernat, B. Cambou, Soil - structure interaction in shield tunneling in soft soil, Computers and Geotechnics 22 (3/4) (1998) 221–242.
[4] K. Komiyai, K. Soga, H. Akagi, T. Hagiwara, M. Bolton, Finite element modelling of excavation and advancement processes of a shield tunneling machine, Soils and Foundations 39 (3) (1999) 37–52.
[5] D. Dias, R. Kastner, M. Maghazi, Three dimensional simulation of slurry shield tunneling, in: O. Kusakabe, K. Fujita, Y. Miyazaki (Eds.), Geotechnical Aspects of Underground Construction in Soft Ground, Tokyo 1999, Balkema, Rotterdam, 2000, pp. 351–356.
[6] G. Swoboda, A. Abu-Krishna, Three-dimensional numerical modelling for tbm tunneling in consolidated clay, Tunneling and Underground Space Technology 14 (1999) 327–333.
[7] T. Kasper, G. Meschke, A 3D finite element model for TBM tunneling in soft ground, International Journal for Numerical and Analytical Methods in Geomechanics 28 (2004) 1441–1460.
[8] T. Kasper, G. Meschke, A numerical study on the effect of soil and grout material properties and cover depth in shield tunneling, Computers & Geotechnics 33 (4–5), 234–247.
[9] F. Nagel, J. Staschiet, G. Meschke, Process-oriented numerical simulation of shield tunneling in soft soils, Geomechanics and Tunneling 3 (3) (2010) 268–282.
[10] J. Staschiet, F. Nagel, G. Meschke, M. Stavropoulou, G. Exadaktylos, An automatic modeller for finite element simulations of shield tunneling, in: J. Eberhardtsteiner, G. Beer, C. Hellmich, H. Mang, G. Meschke, W. Schubert (Eds.), Computational Modelling in Tunneling (EURO:TUN 2007), Vienna, Austria, 2007, CD-ROM.
[11] A. Alsahl, J. Staschiet, G. Meschke, Computational framework for 3d adaptive simulation of excavation and advancement processes in mechanized tunneling, in: G. Meschke, J. Eberhardtsteiner, K. Soga, T. Schanz, M. Thewes (Eds.), Computational Methods in Tunneling and Subsurface Engineering (EURO:TUN 2013), 2013, pp. 85–96.
[12] G. Bui, J. Staschiet, G. Meschke, A parallel block preconditioner for coupled simulations of partially saturated soils in finite element analyses, in: B. Topping, P. Iványi (Eds.), The Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp, 2013, p. paper 24.
[13] J. Kleijnen, Design and Analysis of Simulation Experiments. Springer Science + Business Media, 2008.
[14] L. W. Friedman, Simulation Metamodel, 1st Edition, no. 978-1-4612-8556-4, Springer-Verlag New York, 1996.
[15] J. Kleijnen, Regression metamodels for generalizing simulation results, IEEE Transactions on Systems, Man, and Cybernetics 9 (2) (1979) 93 – 96.
[16] D. Fonseca, D. Navarres, G. Moynihan, Simulation metamodeling through artificial neural networks, Engineering Applications of Artificial Intelligence 16 (3) (2003) 177 – 183.
[17] M. F. Hussain, R. R. Barton, S. B. Joshi, Metamodeling: Radial basis functions, versus polynomials, European Journal of Operational Research 138 (1) (2002) 142 – 154.
[18] B. Ankenman, B. L. Nelson, J. Staum, Stochastic kriging for simulation metamodeling, Operations Research 58 (2) (2010) 371 – 382.
[19] J. Kleijnen, J. P. C. Kleijnen, R. G. Sargent, R. G. Sargent, A methodology for fitting and validating metamodels in simulation, European Journal of Operational Research 120 (2000) 14 – 29.
[20] J. Poropudas, K. Virtanen, Game-theoretic validation and analysis of air combat simulation models, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans - Special issue on model-based diagnosis 40 (5) (2010) 1057 – 1070.
[21] V. Buljak, Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics, xiv Edition, Springer Berlin Heidelberg, 2012.
[22] R. Cheng, S. Currie, Optimization by simulation metamodeling methods, in: : Proceedings of the 2004 Winter Simulation Conference, Washington, DC, 2004, pp. 473 – 478.
[23] Z. Duan, R. Sterling, Rapid prediction of surface settlements due to normal microtunneling operations, in: Proc. No-Dig, Montreal, Canada, 2002.
[24] G. J. Tsekouras, J. Koukoulis, M. -K. Nikolainakoy, N. E. Mastorakis, Prediction of face settlement during tunneling excavation using artificial neural network, in: International Conference on Engineering Mechanics, Structures, Engineering Geology, Crete Island, Greece, 2008, pp. 22–24.
[25] B. Pichler, R. Lackner, H. Mang, Back analysis of model parameters in geotechnical engineering by means of soft computing, International Journal for Numerical Methods in Engineering 57 (14) (2003) 1943–1978.
[26] M. Calvello, R. J. Finno, Selecting parameters to optimize in model calibration by reverse analysis, Computers and Geotechnics 31 (5) (2004) 410 – 424.
[27] X.-T. Feng, B.-R. Chen, C. Yang, H. Zhou, X. Ding, Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm, International Journal of Rock Mechanics and Mining Sciences 43 (5) (2006) 789 – 801.
[28] T. Schanz, M. M. Zimmerer, M. Datcheva, J. Meier, Identification of constitutive parameters for numerical models via inverse approach, Felsbau, Rock and Soil Engineering 24 (2) (2006) 11–21.
[29] Meier, Parameterbestimmung mittels inversen verfahren für geotechnische problemstellungen, Ph.D. thesis, Bauhaus-Universität Weimar (2008).
[30] Z. Song, S. Ren, Z. C. Guo, The tunnel surrounding rock parameters identification method based on ps-oann, Applied Mechanics and Materials 94-96 (2011) 637–640. doi:10.4028/www.scientific.net/AMM.94-96.637.
[31] G. Meschke, J. Ninic, J. Staschiet, A. Alsahl, Parallelized computational modeling of pile-soil interactions in mechanized tunneling, Engineering Structures 47 (2013) 35 – 44, invited paper for Special Issue Computational Mechanics.
[32] J. Meier, T. Schanz, Benchmarking of optimization algorithms, in: EURO:TUN 2013 Computational Methods in Tunneling and Subsurface Engineering, Bochum, Germany, 2013.
[33] P. Dadvand, R. Rossi, E. Ohate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, Archives of Computational Methods in Engineering 17 (2010) 253–297.
[34] F. Nagel, G. Meschke, An elasto-plastic three phase model for partially saturated soil for the finite element simulation of compressed air support in tunnelling, International Journal for Numerical and Analytical Methods in Geomechanics 34 (2010) 605–625, doi:10.1002/nag.828.
[35] H. Yu, CASM: a unified state parameter model for clay and sand, International Journal for Numerical and Analytical Methods in Geomechanics 48 (1998) 773–778.
[36] T. Laursen, Computational Contact and Impact Mechanics, Springer, Berlin-Heidelberg, 2002.
[37] T. Kasper, G. Meschke, On the influence of face pressure, grouting pressure and TBM design in soft ground tunnelling, Tunnelling and Underground Space Technology 21 (2) (2006) 160–171.
[38] B. T. Cao, K. Chmelina, J. Staschitz, G. Meschke, Enhanced monitoring and simulation assisted tunnelling (emsat), in: G. Meschke, J. Eberhardsteiner, K. Soga, T. Schanz, M. Thewes (Eds.), Computational Methods in Tunneling and Subsurface Engineering (EURO:TUN 2013), 2013, in print.
[39] W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, The bulletin of mathematical biophysics 5 (4) (1943) 115–133.
[40] B. Widrow, M. Hoff, Adaptive switching circuits, IRE WESCON Convention Record 4 (1960) 96–104.
[41] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Washington, D. C., Spartan Books, 1962.
[42] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533 – 536. doi:10.1038/323533a0.
[43] J. Kennedy, R. C. Eberhart, Particle swarm optimization, in: I. Press (Ed.), Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 1995, pp. 1942 – 1948.
[44] J. Kennedy, W. M. Spears, Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator, in: Proceedings of the IEEE Congress on Evolutionary Computation, 1998.
[45] J. P. T. Yusong, N. C. J. Prospero, Training neural networks using multi-objective particle swarm optimization, in: Advances in Natural Computation, Vol. 4221 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 879–888.
[46] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Systems 2 (1989) 303–314.