Identification and prediction of time-dependent structural behavior with recurrent neural networks for uncertain data

Steffen Freitag, Wolfgang Graf and Michael Kaliske
Institute for Structural Analysis, Technische Universität Dresden, Germany,
steffen.freitag@tu-dresden.de

Abstract. In this paper, an approach is introduced which permits a model-free identification and prediction of time-dependent structural behavior. The numerical approach is based on recurrent neural networks for uncertain data. Time-dependent results obtained from measurements or numerical analysis are used to identify the uncertain long-term behavior of engineering structures. Thereby, the uncertainty of time-dependent loadings, environmental influences, and structural responses is modelled by means of fuzzy processes. The identification of uncertain dependencies between structural action processes and structural response processes is realized with recurrent neural networks for fuzzy data. Algorithms for network training and prediction are presented. The new recurrent neural network approach for fuzzy data is verified by a fuzzy fractional rheological material model. The approach is applied to predict the long-term behavior of a textile strengthened reinforced concrete structure.

Keywords: time-dependent structural behavior, model-free prediction, fuzzy process, recurrent neural network, textile reinforced concrete

1. Introduction

The long-term behavior of civil engineering structures depends on a multiplicity of environmental influences. In general, all time-dependent influences of a structure are uncertain processes which lead to uncertain time-varying structural responses. Uncertain processes can be captured with non-traditional uncertainty models, see (Möller and Beer, 2008). These uncertain structural processes are divided into structural action processes containing time-dependent loadings, temperature, humidity, etc. and structural response processes reflecting the structural behavior, e.g. displacements, internal forces.

For robust design of structures, numerical methods are required which can be used to identify and predict uncertain time-dependent structural behavior. For this purpose, a novel approach is proposed, which combines neural computing (artificial neural networks, see e.g. (Haykin, 1999)) and mapping of fuzzy data (fuzzy analysis, see e.g. (Möller et al., 2000)). Recurrent neural networks for fuzzy data are introduced for the numerical prediction of time-dependent structural responses under consideration of uncertain action processes.

The artificial neural network concept is adapted from the structure and the functionality of the human brain. It is a powerful tool to capture and to learn functional dependencies in data. An overview of neural network applications in civil engineering is given in (Adeli, 2001). The

4th International Workshop on Reliable Engineering Computing (REC 2010)
Edited by Michael Beer, Rafi L. Muhanna and Robert L. Mullen

ISBN: 978-981-08-5118-7. *Published by* Research Publishing Services.
doi:10.3850/978-981-08-5118-7_026

widely-used type in engineering applications is the multilayer perceptron network with feed forward architecture (Haykin, 1999). In order to consider time-dependent effects of the structural behavior, advanced network architectures have to be applied. In this context, recurrent neural networks have been developed for temporal signal processing, see e.g. (Zell, 1996). They are suitable for the mapping of structural processes obtained by experiments or numerical monitoring onto time-dependent structural responses.

If a structural process is observed experimentally with the help of measurement devices, it is not possible to assign precise values to the observed events. That means, data uncertainty occurs which may result from scale-dependent effects, varying boundary conditions which are not considered, inaccuracies in the measurements, and incomplete sets of observations. Therefore, measured results are more or less characterized by data uncertainty which originates in imprecision. In this contribution, the imprecision is modelled by means of fuzzy sets. Time-dependent structural parameters are quantified as fuzzy processes, see e.g. (Möller and Beer, 2004) and (Möller and Reuter, 2007). Three types of mapping fuzzy processes are introduced in Section 2.

In Section 3, recurrent neural networks for fuzzy data are presented. As an extension to (Freitag et al., 2009b) and (Graf et al., (in press)), the prediction and the training of recurrent neural networks with trainable fuzzy network parameters is introduced. The mapping of fuzzy input onto fuzzy output values is described. Beside fuzzy data, also intervals and deterministic numbers may be processed.

The developed recurrent neural network approach for fuzzy data is verified by a fractional rheological material model, see (Oeser and Freitag, 2009), in Section 4. Uncertain stress-strain-time-dependencies obtained by numerical monitoring are used for the adjustment of fuzzy network parameters. The resulting network is utilized for the prediction of further stress-strain-time-dependencies.

In Section 5, the developed recurrent neural network approach is applied for the prediction of the long-term displacement of a reinforced concrete plate which was strengthened with textile reinforced concrete (TRC) layers. The numerical modelling of the long-term behavior of this new composite material requires an experimental data basis of the temporal structural alteration under environmental conditions. However, experimental investigations covering long-term periods are not efficient in engineering practice due to resource restrictions in time or money or both, see (Freitag et al., 2009a). In most cases, experimental data covering only limited time periods are available. They can be used to predict the time-dependent structural behavior for longer time periods.

2. Fuzzy processes

Time-dependent structural behavior may be determined by means of structural monitoring. In the conventional meaning, parameters representing the structural behavior are obtained with the aid of experimental investigations. However, mainly due to the increase in computational power, the numerical monitoring becomes important and provides insight into the behavior of structures or structural members. Thereby, the time-dependent structural behavior is numerically simulated utilizing nonlinear computational models. Repeated numerical simulations with varying input parameters may be also interpreted as numerical experiments.

Results of experimental investigations are generally characterized by uncertainty because of physically originated variations and imprecision. The description of the observed phenomena close to reality requires the consideration of uncertainty. The imprecision may be best modelled by means of the uncertainty model fuzziness. Time-dependent structural parameters are quantified with the aid of fuzzy processes.

2.1. Definition

A fuzzy process is defined according to

$$\tilde{x}(\tau) = \{\tilde{x}_\tau = \tilde{x}(\tau) \,\forall\, \tau \mid \tilde{x}_\tau \in \mathbf{F}(\mathbf{X})\} \tag{1}$$

as a set of fuzzy values $\tilde{x}_\tau$ belonging to the set $\mathbf{F}(\mathbf{X})$ of all fuzzy values defined on a fundamental set $\mathbf{X}$, see e.g. (Möller and Beer, 2004). Uncertain functional values $\tilde{x}(\tau)$ are assigned to each time τ. These uncertain functional values represent fuzzy values $\tilde{x}$ gradually assessed by membership functions $\mu(x)$.

Fuzzy data are described by convex fuzzy values in this paper. They are subdivided into fuzzy numbers and fuzzy intervals, see (Möller and Beer, 2004). For fuzzy numbers, one deterministic argument (kernel value) x exists with $\mu(x) = 1.0$. By contrast, the membership function of a fuzzy interval consists of a kernel interval $[{}_{1.0l}x,\ {}_{1.0r}x]$ with $\mu(x) = 1.0 \,\forall\, x \in [{}_{1.0l}x,\ {}_{1.0r}x]$.

A fuzzy triangular number $\tilde{x} =< x_1,\ x_2,\ x_3 >$ is a special fuzzy number whose membership function $\mu(x)$ is characterized by linear functions between $\mu(x) \to 0+$ and $\mu(x) = 1.0$. It is completely identified by the values x_1, x_2 and x_3 for which hold $\mu(x_1) \to 0+$, $\mu(x_2) = 1.0$ and $\mu(x_3) \to 0+$, respectively.

The support $B\,(\tilde{x})$ of a fuzzy value is an interval containing all elements $x \in \tilde{x}$ with a membership value $\mu(x) > 0$. The interval bounds may be obtained by the operation $\mu(x) \to 0+$.

2.2. Discretization

For the numerical prediction of fuzzy processes, two steps of discretization are required. At first, all fuzzy processes have to be discretized in time resulting in sets of fuzzy values. Then, the membership functions of the obtained fuzzy values are subdivided in sets of α-cuts.

The fuzzy processes are discretized in time by N equidistant time steps $[n = 1, \ldots, N]$. Fuzzy values ${}^{[n]}\tilde{x}$ are obtained for every time point ${}^{[n]}\tau$. As the fuzzy processes represent structural parameters, the applied discretization algorithm depends on the experimentally or numerically available data base resulting from structural monitoring.

- If the time step length in monitoring is identical to the time step length of discretization, the results obtained by structural monitoring may directly be applied within the numerical prediction approach.

- Otherwise, the fuzzy values ${}^{[n]}\tilde{x}$ may be determined by interpolation using the values before and after ${}^{[n]}\tau$ or by combination of multiple values within the time period ${}^{[n-1]}\tau < \tau < {}^{[n+1]}\tau$.

In the second step, the membership functions $\mu\left({}^{[n]}\tilde{x}\right)$ of the fuzzy values ${}^{[n]}\tilde{x}$ are subdivided into $s = 1, \ldots, S$ α-cuts, see (Möller et al., 2000). Regarding convex fuzzy values, a connected interval $\left[{}^{[n]}_{\alpha_s l}x,\ {}^{[n]}_{\alpha_s r}x\right]$ with the left bound

$$ {}^{[n]}_{\alpha_s l}x = \min\left[{}^{[n]}x \in {}^{[n]}\mathbf{X} \mid \mu({}^{[n]}x) \geq \alpha_s\right] \tag{2}$$

and the right bound

$$ {}^{[n]}_{\alpha_s r}x = \max\left[{}^{[n]}x \in {}^{[n]}\mathbf{X} \mid \mu({}^{[n]}x) \geq \alpha_s\right] \tag{3}$$

is obtained for each α-cut, see Figure 1.

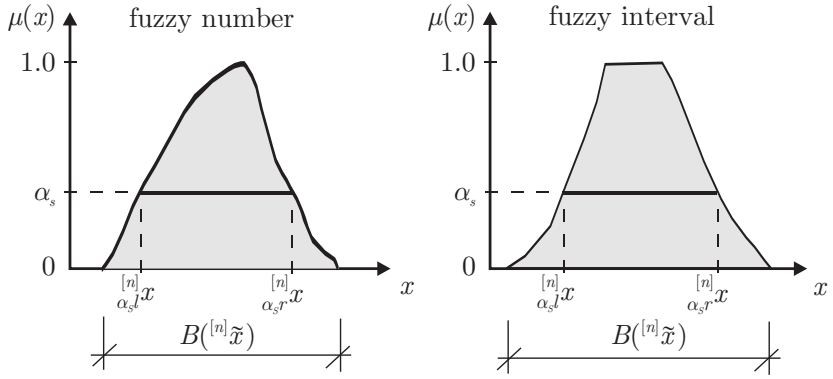

Figure 1. α-cuts of fuzzy values.

The interval $\left[{}^{[n]}_{\alpha_1 l}x,\ {}^{[n]}_{\alpha_1 r}x\right]$ of α-cut $\alpha_1 = 0$ is defined by the support $B\left({}^{[n]}\tilde{x}\right)$ of the discretized fuzzy value. For fuzzy numbers holds ${}^{[n]}_{\alpha_S l}x = {}^{[n]}_{\alpha_S r}x$.

2.3. Mapping of fuzzy processes

Fuzzy action processes can be mapped onto fuzzy response processes with numerical models in three different ways:

1. Fuzzy action processes $\underline{\tilde{x}}(\tau)$ are mapped onto fuzzy response processes $\underline{\tilde{z}}(\tau)$. All model parameters are deterministic numbers. This is denoted as **Type 1 mapping**.

$$ \underline{\tilde{x}}(\tau) \ \mapsto \ \underline{\tilde{z}}(\tau) \tag{4}$$

2. Deterministic action processes $\underline{x}(\tau)$ are mapped onto fuzzy response processes $\underline{\tilde{z}}(\tau)$. Fuzzy model parameters are required for this uncertain mapping denoted as **Type 2 mapping**.

$$ \underline{x}(\tau) \ \tilde{\mapsto} \ \underline{\tilde{z}}(\tau) \tag{5}$$

3. The combination of both mapping types results in the uncertain mapping of fuzzy action processes onto fuzzy response processes utilizing fuzzy model parameters. This is denoted as **Type 3 mapping**.

$$ \underline{\tilde{x}}(\tau) \ \tilde{\mapsto} \ \underline{\tilde{z}}(\tau) \tag{6}$$

3. Recurrent neural networks for fuzzy data

The long-term behavior of engineering structures is influenced by time-dependent uncertain alterations which can be modelled mathematically by means of fuzzy structural processes. In many cases, the structural response is characterized by a so-called fading memory phenomenon, see e.g. (Oeser and Freitag, 2009). Fading memory phenomenon means that stresses which have been applied to the structure with a distance in time gradually lose their impact on the future structural behavior. In order to capture time-history effects with neural networks, an architecture with internal time representation is required. Therefore, a partially recurrent neural network is developed which is based on an extended ELMAN network, see e.g. (Zell, 1996). The partially recurrent neural network contains elements of JORDAN networks (Jordan, 1990) and ELMAN networks (Elman, 1990).

3.1. Network architecture

The architecture of a partially recurrent neural network for fuzzy data is shown in Figure 2. The network consists of (M) layers, that is, an input layer, $(M-2)$ hidden layers and an output layer.

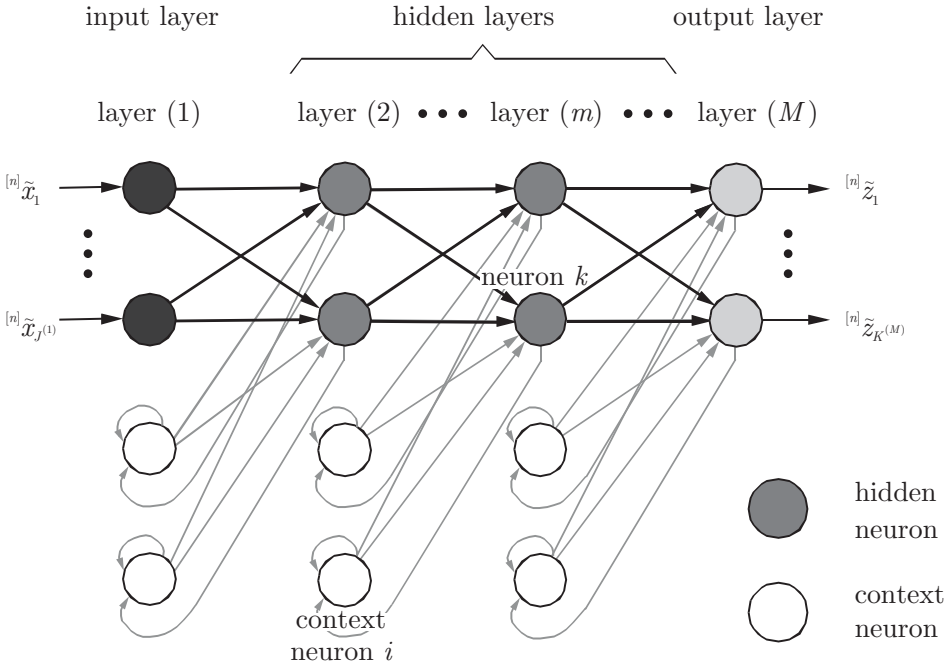

Figure 2. Partially recurrent neural network.

The input neurons are fed with uncertain structural actions $^{[n]}\tilde{x}_j, j = 1, \ldots, J^{(1)}$ at time step $[n]$. In the input neurons, the uncertain structural stresses $^{[n]}\tilde{x}_j$ are scaled to input signals $^{[n]}\tilde{x}_j^{(1)}$. The neurons of the hidden layers as well as the output layer are connected to context neurons, which enables the consideration of time-history. The context neurons send uncertain weighted signals with a time delay to the hidden and output neurons, see Figure 2. Uncertain output signals $^{[n]}\tilde{x}_k^{(M)}, k = 1, \ldots, K^{(M)}$ are computed by the recurrent network in each time step. The uncertain

output signals of layer (M) are scaled to the structural responses ${}^{[n]}\tilde{z}_k$, which contain the entire stress history of the structure.

The treatment of fuzzy data with recurrent neural networks requires an extension of the signal processing of precise data presented in (Oeser and Freitag, 2009). In general, two ways of computation are possible:

1. The neural network is used as deterministic fundamental solution of a fuzzy analysis. The prediction of fuzzy structural responses is realized by means of the α-level optimization (Möller et al., 2000) for each α-cut. This leads to a deterministic signal processing of the network.

2. The prediction is carried out by fuzzy arithmetic, which is an extension of interval arithmetic described e.g. in (Moore, 1979). Thereby, fuzzy signals are processed in the network.

The approach presented in this paper is based on fuzzy arithmetic and aims at the prediction of fuzzy structural response processes caused by fuzzy structural action processes. A prediction and a training algorithm for Type 3 mapping with recurrent neural networks are presented. The algorithms can be transferred to Type 1 and Type 2 mapping as special cases of Type 3 mapping. As an extension of the approach presented in (Freitag et al., 2009b), trainable fuzzy network parameters are introduced.

3.2. Prediction algorithm

In Figure 3, the structure of a hidden neuron is displayed. Neuron k is a member of the network layer (m) in Figure 2. The number of synaptic connections $J^{(m-1)}$ corresponds to the number of neurons in the previous network layer $(m-1)$. The number of synaptic connections $I^{(m)}$ corresponds to the number of context neurons.

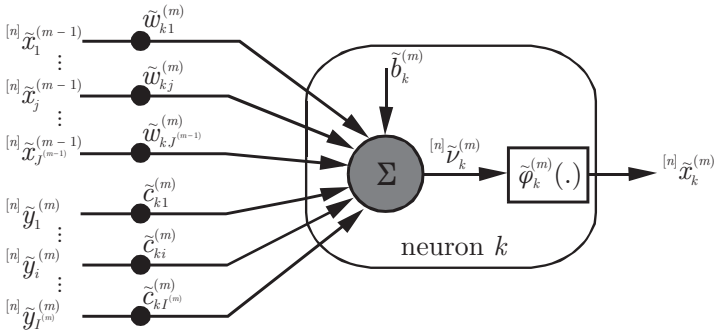

Figure 3. Hidden neuron.

In time step $[n]$, the fuzzy output signal ${}^{[n]}\tilde{x}_k^{(m)}$ of neuron k is obtained by fuzzy arithmetic computations. The argument ${}^{[n]}\tilde{\nu}_k^{(m)}$ of the fuzzy activation function $\tilde{\varphi}_k^{(m)}(.)$ is determined by interval arithmetic operations on each α-cut. The left bounds of the argument ${}^{[n]}\tilde{\nu}_k^{(m)}$ are obtained by

$$ {}^{[n]}_{\alpha_s l}\nu_k^{(m)} = {}^{[n]}_{\alpha_s l}W_k^{(m)} + {}^{[n]}_{\alpha_s l}C_k^{(m)} + {}_{\alpha_s l}b_k^{(m)} \tag{7} $$

and for the right bounds holds

$$ {}^{[n]}_{\alpha_s r}\nu_k^{(m)} = {}^{[n]}_{\alpha_s r}W_k^{(m)} + {}^{[n]}_{\alpha_s r}C_k^{(m)} + {}_{\alpha_s r}b_k^{(m)} \; . \tag{8} $$

The argument of the fuzzy activation function contains signals from previous layers in ${}^{[n]}\tilde{W}_k^{(m)}$ and signals form the context neurons in ${}^{[n]}\tilde{C}_k^{(m)}$. Additionally, a fuzzy bias value $b_k^{(m)}$ can be added as displayed in Figure 3. The values ${}^{[n]}_{\alpha_s l}W_k^{(m)}$ and ${}^{[n]}_{\alpha_s r}W_k^{(m)}$ in the Eqs. (7) and (8) are produced by multiplying the fuzzy input signals ${}^{[n]}\tilde{x}_j^{(m-1)}$ of neuron k with the fuzzy weights $\tilde{w}_{kj}^{(m)}$ and summing them. The left bounds of the product of two fuzzy values are obtained with the min-operator

$$ {}^{[n]}_{\alpha_s l}W_k^{(m)} = \sum_{j=1}^{J^{(m-1)}} \left[\min \begin{bmatrix} {}^{[n]}_{\alpha_s l}x_j^{(m-1)} \cdot {}_{\alpha_s l}w_{kj}^{(m)}, & {}^{[n]}_{\alpha_s l}x_j^{(m-1)} \cdot {}_{\alpha_s r}w_{kj}^{(m)}, \\ {}^{[n]}_{\alpha_s r}x_j^{(m-1)} \cdot {}_{\alpha_s l}w_{kj}^{(m)}, & {}^{[n]}_{\alpha_s r}x_j^{(m-1)} \cdot {}_{\alpha_s r}w_{kj}^{(m)} \end{bmatrix} \right] \tag{9} $$

and the right bounds are determined with the max-operator

$$ {}^{[n]}_{\alpha_s r}W_k^{(m)} = \sum_{j=1}^{J^{(m-1)}} \left[\max \begin{bmatrix} {}^{[n]}_{\alpha_s l}x_j^{(m-1)} \cdot {}_{\alpha_s l}w_{kj}^{(m)}, & {}^{[n]}_{\alpha_s l}x_j^{(m-1)} \cdot {}_{\alpha_s r}w_{kj}^{(m)}, \\ {}^{[n]}_{\alpha_s r}x_j^{(m-1)} \cdot {}_{\alpha_s l}w_{kj}^{(m)}, & {}^{[n]}_{\alpha_s r}x_j^{(m-1)} \cdot {}_{\alpha_s r}w_{kj}^{(m)} \end{bmatrix} \right] \; . \tag{10} $$

The values ${}^{[n]}_{\alpha_s l}C_k^{(m)}$ and ${}^{[n]}_{\alpha_s r}C_k^{(m)}$ in the Eqs. (7) and (8) include the fuzzy history signals ${}^{[n]}\tilde{y}_i^{(m)}$ multiplied by the fuzzy context weights $\tilde{c}_{ki}^{(m)}$. They are determined by

$$ {}^{[n]}_{\alpha_s l}C_k^{(m)} = \sum_{i=1}^{I^{(m)}} \left[\min \begin{bmatrix} {}^{[n]}_{\alpha_s l}y_i^{(m)} \cdot {}_{\alpha_s l}c_{ki}^{(m)}, & {}^{[n]}_{\alpha_s l}y_i^{(m)} \cdot {}_{\alpha_s r}c_{ki}^{(m)}, \\ {}^{[n]}_{\alpha_s r}y_i^{(m)} \cdot {}_{\alpha_s l}c_{ki}^{(m)}, & {}^{[n]}_{\alpha_s r}y_i^{(m)} \cdot {}_{\alpha_s r}c_{ki}^{(m)} \end{bmatrix} \right] \tag{11} $$

and

$$ {}^{[n]}_{\alpha_s r}C_k^{(m)} = \sum_{i=1}^{I^{(m)}} \left[\max \begin{bmatrix} {}^{[n]}_{\alpha_s l}y_i^{(m)} \cdot {}_{\alpha_s l}c_{ki}^{(m)}, & {}^{[n]}_{\alpha_s l}y_i^{(m)} \cdot {}_{\alpha_s r}c_{ki}^{(m)}, \\ {}^{[n]}_{\alpha_s r}y_i^{(m)} \cdot {}_{\alpha_s l}c_{ki}^{(m)}, & {}^{[n]}_{\alpha_s r}y_i^{(m)} \cdot {}_{\alpha_s r}c_{ki}^{(m)} \end{bmatrix} \right] \; . \tag{12} $$

The signals obtained by the Eqs. (7) and (8) are processed with monotonic and differentiable fuzzy activation functions $\tilde{\varphi}$. The left and right bounds of the fuzzy output signal ${}^{[n]}\tilde{x}_k^{(m)}$ of neuron k are obtained by

$$ {}^{[n]}_{\alpha_s l}x_k^{(m)} = {}_{\alpha_s l}\varphi_k^{(m)} \left({}^{[n]}_{\alpha_s l}\nu_k^{(m)} \right) \tag{13} $$

and

$$ {}^{[n]}_{\alpha_s r}x_k^{(m)} = {}_{\alpha_s r}\varphi_k^{(m)} \left({}^{[n]}_{\alpha_s r}\nu_k^{(m)} \right) \; . \tag{14} $$

Various types of monotonic and differentiable fuzzy activation functions can be used. In Figure 4, a nonlinear fuzzy activation function in the form of the area hyperbolic sine

$$ \tilde{\varphi}_k^{(m)} \left({}^{[n]}\tilde{\nu}_k^{(m)} \right) = \tilde{a}_k^{(m)} \cdot \operatorname{arsinh}\left({}^{[n]}\tilde{\nu}_k^{(m)} \right) = \tilde{a}_k^{(m)} \cdot \ln \left({}^{[n]}\tilde{\nu}_k^{(m)} + \sqrt{\left({}^{[n]}\tilde{\nu}_k^{(m)} \right)^2 + 1} \right) \tag{15} $$

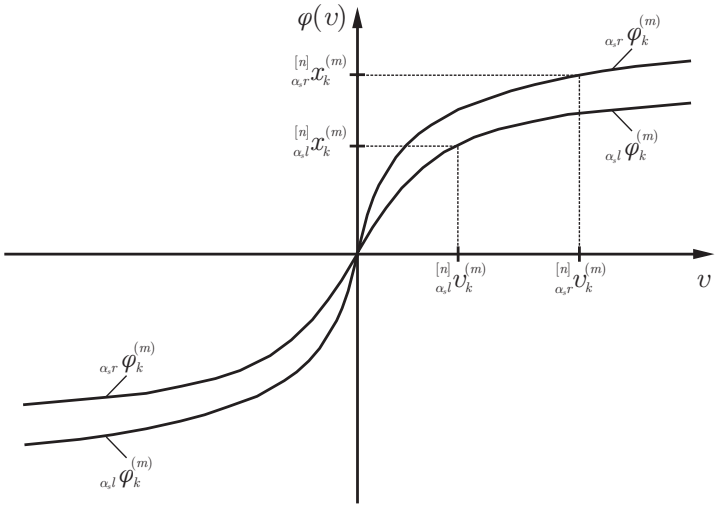

Figure 4. Signal processing with fuzzy area hyperbolic sine

is plotted as an example. The parameter $\tilde{a}_k^{(m)}$ is a fuzzy value, which enables an uncertain signal processing.

In Figure 5, the structure of the context neuron i as part of the recurrent neural network in Figure 2 is shown. The context neuron i of layer (m) is fed with the fuzzy output signal ${}^{[n-1]}\tilde{x}_i^{(m)}$ of the hidden neuron $k = i$.

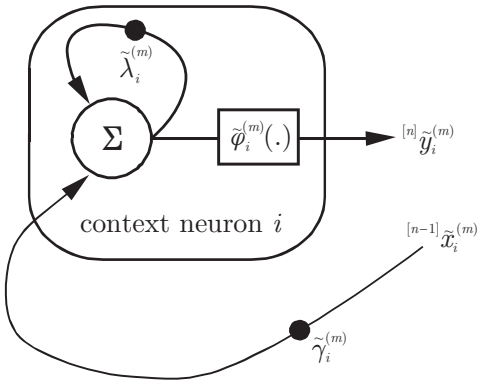

Figure 5. Context neuron.

The parameter $\tilde{\gamma}_i^{(m)}$ in Figure 5 is the fuzzy memory factor, which is defined as fuzzy value in the interval [0, 1]. The factor $\tilde{\lambda}_i^{(m)}$ represents the fuzzy feedback factor of the context neuron i. This factor, also defined as fuzzy value in [0, 1], controls the self feedback of the previous fuzzy history signal ${}^{[n-1]}\tilde{y}_i^{(m)}$ to the current fuzzy history signal ${}^{[n]}\tilde{y}_i^{(m)}$.

The fuzzy history signal ${}^{[n]}\tilde{y}_i^{(m)}$ of the context neuron i is determined by fuzzy arithmetic. The interval bounds on the α-cuts are obtained by

$$ {}^{[n]}_{\alpha_s l} y_i^{(m)} = {}^{[n-1]}_{\alpha_s l} G_i^{(m)} + {}^{[n-1]}_{\alpha_s l} L_i^{(m)} \tag{16} $$

and

$$ {}^{[n]}_{\alpha_s r}y_i^{(m)} = {}^{[n-1]}_{\alpha_s r}G_i^{(m)} + {}^{[n-1]}_{\alpha_s r}L_i^{(m)} \ . \tag{17} $$

For the first term in Eq. (16) holds

$$ {}^{[n-1]}_{\alpha_s l}G_i^{(m)} = \begin{cases} \left[{}^{[n-1]}_{\alpha_s l}x_i^{(m)} \cdot {}_{\alpha_s l}\gamma_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s l}x_i^{(m)} \geq 0 \\ \left[{}^{[n-1]}_{\alpha_s l}x_i^{(m)} \cdot {}_{\alpha_s r}\gamma_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s l}x_i^{(m)} < 0 \ . \end{cases} \tag{18} $$

The first term of Eq. (17) is obtained similarly by

$$ {}^{[n-1]}_{\alpha_s r}G_i^{(m)} = \begin{cases} \left[{}^{[n-1]}_{\alpha_s r}x_i^{(m)} \cdot {}_{\alpha_s r}\gamma_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s r}x_i^{(m)} \geq 0 \\ \left[{}^{[n-1]}_{\alpha_s r}x_i^{(m)} \cdot {}_{\alpha_s l}\gamma_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s r}x_i^{(m)} < 0 \ . \end{cases} \tag{19} $$

The second terms in the Eqs. (16) and (17) yield

$$ {}^{[n-1]}_{\alpha_s l}L_i^{(m)} = \begin{cases} \left[{}^{[n-1]}_{\alpha_s l}y_i^{(m)} \cdot {}_{\alpha_s l}\lambda_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s l}y_i^{(m)} \geq 0 \\ \left[{}^{[n-1]}_{\alpha_s l}y_i^{(m)} \cdot {}_{\alpha_s r}\lambda_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s l}y_i^{(m)} < 0 \end{cases} \tag{20} $$

and

$$ {}^{[n-1]}_{\alpha_s r}L_i^{(m)} = \begin{cases} \left[{}^{[n-1]}_{\alpha_s r}y_i^{(m)} \cdot {}_{\alpha_s r}\lambda_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s r}y_i^{(m)} \geq 0 \\ \left[{}^{[n-1]}_{\alpha_s r}y_i^{(m)} \cdot {}_{\alpha_s l}\lambda_i^{(m)}\right], & \text{if } {}^{[n-1]}_{\alpha_s r}y_i^{(m)} < 0 \ , \end{cases} \tag{21} $$

respectively.

As initial condition, no history is considered in the first time step $[n = 1]$. It holds ${}^{[1]}\tilde{y}_i^{(m)} = 0$ for $i = 1, \ldots, I^{(m)}$ and $m = 2, \ldots, (M)$. With this initial condition, the second terms ${}^{[1]}_{\alpha_s l}C_k^{(m)}$ and ${}^{[1]}_{\alpha_s r}C_k^{(m)}$ of the Eqs. (11) and (12) are zero. In the following time steps $[n > 1]$, the interval bounds of the fuzzy history signals ${}^{[n]}\tilde{y}_i^{(m)}$ are calculated with the Eqs. (16) and (17).

If all fuzzy memory factors $\tilde{\gamma}_i^{(m)}$ in the partially recurrent neural network are zero, no history is considered $\left({}^{[n]}\tilde{y}_i^{(m)} = 0\right)$. That is, a feed forward neural network for fuzzy data is obtained as a special case of the developed recurrent neural network for fuzzy data.

An essential advantage of the presented prediction algorithm is the warranty that each fuzzy result of the recurrent neural network verifies the properties of fuzzy values according to Section 2.1. A post-processing of the fuzzy results according to their convexity is not necessary.

3.3. Training algorithm

The objective of the network training is the adjustment of the network parameters. This requires an experimentally or numerically obtained data basis of uncertain structural action and response processes. A number of learning methods for artificial neural networks is available, see e.g. (Haykin, 1999) or (Zell, 1996). For recurrent neural networks, the temporal succession of the data series has to be considered. The modified backpropagation algorithms for partially recurrent neural networks,

described in (Freitag et al., 2009b) and (Graf et al., (in press)), provide a basis for the development of a new training algorithm with trainable fuzzy network parameters.

In the training, the fuzzy output signals ${}^{[n]}\tilde{x}_k^{(M)}$ of the recurrent neural network are compared to the desired responses ${}^{[n]}\tilde{d}_k^{(M)}, k = 1, \ldots, K^{(M)}$ (experimentally or numerically obtained data) caused by the fuzzy structural actions ${}^{[n]}\tilde{x}_j$, $j = 1, \ldots, J^{(1)}$ of all time steps $[n = 1, \ldots, N]$.

A measure for the distance of fuzzy values is required to determine the prediction error of the network. In general, different distance measures are possible, see e.g. (Beer, 2007) and (Johanyák and Kovács, 2005). In view of a gradient based training algorithm with fuzzy signals represented in α-cuts, according to Section 2.2, an error criterion is selected which contains information of all α-cuts. The total error at time step $[n]$ is defined as

$$ {}^{[n]}E = \frac{1}{2} \sum_{k=1}^{K^{(M)}} \left\{ \sum_{s=1}^{S} \left[\left({}^{[n]}_{\alpha_s l} d_k^{(M)} - {}^{[n]}_{\alpha_s l} x_k^{(M)} \right)^2 + \left({}^{[n]}_{\alpha_s r} d_k^{(M)} - {}^{[n]}_{\alpha_s r} x_k^{(M)} \right)^2 \right] \right\} . \tag{22} $$

This error measure enables the calculation of the local gradient at each bound of α-cut s without information of other bounds, see Eqs. (42), (43), (44) and (46).

In order to validate the recurrent neural network, all observed processes are divided into training and validation patterns. If different action processes and corresponding structural responses are available from experimental or numerical investigations, independent training and validation patterns can be obtained by separating complete time series. The averaged total training error

$$ E_{tr}^{av} = \frac{1}{\sum_{h_{tr}=1}^{H_{tr}} N_{h_{tr}}} \sum_{h_{tr}=1}^{H_{tr}} \left[\sum_{n=1}^{N_{h_{tr}}} \left[{}^{[n]}E \right] \right] \tag{23} $$

contains the errors of the structural responses of all training time series $h_{tr} = 1, \ldots, H_{tr}$. The averaged total validation error

$$ E_{v}^{av} = \frac{1}{\sum_{h_{v}=1}^{H_{v}} N_{h_{v}}} \sum_{h_{v}=1}^{H_{v}} \left[\sum_{n=1}^{N_{h_{v}}} \left[{}^{[n]}E \right] \right] \tag{24} $$

includes the errors of the structural responses of all validation time series $h_v = 1, \ldots, H_v$. If only one time series is available, the data of the time steps $[n = 1, \ldots, N_{tr}]$ form the training pattern and the data of the time steps $[n = N_{tr} + 1, \ldots, N]$ form the validation pattern. In this case, the averaged total training error

$$ E_{tr}^{av} = \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \left[{}^{[n]}E \right] \tag{25} $$

and the averaged total validation error

$$ E_{v}^{av} = \frac{1}{N - N_{tr}} \sum_{n=N_{tr}+1}^{N} \left[{}^{[n]}E \right] \tag{26} $$

are calculated, respectively. With the Eqs. (23) and (24) or (25) and (26), an error monitoring is performed to observe the prediction quality of the network during the training process.

An optimization task is formulated for the network adjustment. Hence, Eq. (23) or Eq. (25) can be used as an evolution equation. The objective of the optimization is the minimization of the averaged total training error. The network architecture (the number of hidden layers and neurons) is defined a priori. The following initial conditions are set. The fuzzy memory factors $\tilde{\gamma}_i^{(m)}$ as well as the fuzzy feedback factors $\tilde{\lambda}_i^{(m)}$ of the partially recurrent network are chosen randomly in the interval $[0, 1]$, and then kept constant throughout the training process. The fuzzy weights $\tilde{w}_{kj}^{(m)}$, fuzzy context weights $\tilde{c}_{ki}^{(m)}$ and fuzzy bias values $\tilde{b}_k^{(m)}$ are chosen randomly in an interval, e.g. $[-1,\ 1]$. In contrast to the fuzzy memory and fuzzy feedback factors, the fuzzy weights, fuzzy context weights and fuzzy bias values are modified in the training process.

Here, a sequential training mode is applied. That is, an update of the free fuzzy network parameters is computed in each time step with

$$ {}^{[n+1]}\tilde{w}_{kj}^{(m)} = {}^{[n]}\Delta\tilde{w}_{kj}^{(m)} + {}^{[n]}\tilde{w}_{kj}^{(m)} \ , \tag{27} $$

$$ {}^{[n+1]}\tilde{c}_{ki}^{(m)} = {}^{[n]}\Delta\tilde{c}_{ki}^{(m)} + {}^{[n]}\tilde{c}_{ki}^{(m)} \tag{28} $$

and

$$ {}^{[n+1]}\tilde{b}_{k}^{(m)} = {}^{[n]}\Delta\tilde{b}_{k}^{(m)} + {}^{[n]}\tilde{b}_{k}^{(m)} \ . \tag{29} $$

The incremental correction of the fuzzy weights is calculated by

$$ {}^{[n]}_{\alpha_s l}\Delta w_{kj}^{(m)} = -{}^{[n]}\eta \cdot \frac{\partial^{[n]} E}{\partial^{[n]}_{\alpha_s l} w_{kj}^{(m)}} + \beta \cdot {}^{[n-1]}_{\alpha_s l}\Delta w_{kj}^{(m)} \tag{30} $$

and

$$ {}^{[n]}_{\alpha_s r}\Delta w_{kj}^{(m)} = -{}^{[n]}\eta \cdot \frac{\partial^{[n]} E}{\partial^{[n]}_{\alpha_s r} w_{kj}^{(m)}} + \beta \cdot {}^{[n-1]}_{\alpha_s r}\Delta w_{kj}^{(m)} \ . \tag{31} $$

The correction of the fuzzy context weights is

$$ {}^{[n]}_{\alpha_s l}\Delta c_{ki}^{(m)} = -{}^{[n]}\eta \cdot \frac{\partial^{[n]} E}{\partial^{[n]}_{\alpha_s l} c_{ki}^{(m)}} + \beta \cdot {}^{[n-1]}_{\alpha_s l}\Delta c_{ki}^{(m)} \tag{32} $$

and

$$ {}^{[n]}_{\alpha_s r}\Delta c_{ki}^{(m)} = -{}^{[n]}\eta \cdot \frac{\partial^{[n]} E}{\partial^{[n]}_{\alpha_s r} c_{ki}^{(m)}} + \beta \cdot {}^{[n-1]}_{\alpha_s r}\Delta c_{ki}^{(m)} \ . \tag{33} $$

The fuzzy bias values are corrected by

$$ {}^{[n]}_{\alpha_s l}\Delta b_{k}^{(m)} = -{}^{[n]}\eta \cdot \frac{\partial^{[n]} E}{\partial^{[n]}_{\alpha_s l} b_{k}^{(m)}} + \beta \cdot {}^{[n-1]}_{\alpha_s l}\Delta b_{k}^{(m)} \tag{34} $$

and

$$ {}^{[n]}_{\alpha_s r}\Delta b_{k}^{(m)} = -{}^{[n]}\eta \cdot \frac{\partial^{[n]} E}{\partial^{[n]}_{\alpha_s r} b_{k}^{(m)}} + \beta \cdot {}^{[n-1]}_{\alpha_s r}\Delta b_{k}^{(m)} \ . \tag{35} $$

In the Eqs. (30) to (35), the parameter ${}^{[n]}\eta$ is the learning rate. Because of the history effects in the training patterns, a decreasing learning rate for increasing time steps is recommended. The parameter β is a momentum constant, see e.g. (Haykin, 1999). The error gradient in Eq. (30) is given in dependency of ${}^{[n]}_{\alpha_s l}W^{(m)}_{kj} = \min[.]$ in Eq. (9) by

$$\frac{\partial^{[n]}E}{\partial^{[n]}_{\alpha_s l}w^{(m)}_{kj}} = -\begin{cases} {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}_{\alpha_s l}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s l}W^{(m)}_{kj} \\ {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}_{\alpha_s r}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s l}W^{(m)}_{kj} \\ {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}_{\alpha_s l}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s l}W^{(m)}_{kj} \\ {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}_{\alpha_s r}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s l}W^{(m)}_{kj} \end{cases}. \tag{36}$$

The error gradient in Eq. (31) depends on ${}^{[n]}_{\alpha_s r}W^{(m)}_{kj} = \max[.]$ in Eq. (10), it is

$$\frac{\partial^{[n]}E}{\partial^{[n]}_{\alpha_s r}w^{(m)}_{kj}} = -\begin{cases} {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}_{\alpha_s l}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s r}W^{(m)}_{kj} \\ {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}x^{(m-1)}_{j} \cdot {}_{\alpha_s r}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s r}W^{(m)}_{kj} \\ {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}_{\alpha_s l}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s r}W^{(m)}_{kj} \\ {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}x^{(m-1)}_{j} \cdot {}_{\alpha_s r}w^{(m)}_{kj} = {}^{[n]}_{\alpha_s r}W^{(m)}_{kj} \end{cases}. \tag{37}$$

The error gradient of the left interval bond in Eq. (32) depends on ${}^{[n]}_{\alpha_s l}C^{(m)}_{ki} = \min[.]$ in Eq. (11) and is obtained by

$$\frac{\partial^{[n]}E}{\partial^{[n]}_{\alpha_s l}c^{(m)}_{ki}} = -\begin{cases} {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}_{\alpha_s l}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s l}C^{(m)}_{ki} \\ {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}_{\alpha_s r}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s l}C^{(m)}_{ki} \\ {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}_{\alpha_s l}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s l}C^{(m)}_{ki} \\ {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}_{\alpha_s r}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s l}C^{(m)}_{ki} \end{cases} \tag{38}$$

and the error gradient of the right interval bond in Eq. (33) depends on ${}^{[n]}_{\alpha_s r}C^{(m)}_{ki} = \max[.]$ in Eq. (12) and is

$$\frac{\partial^{[n]}E}{\partial^{[n]}_{\alpha_s r}c^{(m)}_{ki}} = -\begin{cases} {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}_{\alpha_s l}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s r}C^{(m)}_{ki} \\ {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s l}y^{(m)}_{i} \cdot {}_{\alpha_s r}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s r}C^{(m)}_{ki} \\ {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s l}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}_{\alpha_s l}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s r}C^{(m)}_{ki} \\ {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}^{[n]}_{\alpha_s r}\delta^{(m)}_{k}, & \text{if } {}^{[n]}_{\alpha_s r}y^{(m)}_{i} \cdot {}_{\alpha_s r}c^{(m)}_{ki} = {}^{[n]}_{\alpha_s r}C^{(m)}_{ki} \end{cases}. \tag{39}$$

For the fuzzy bias values the error gradients in the Eqs. (34) and (35) are obtained by

$$\frac{\partial^{[n]}E}{\partial^{[n]}_{\alpha_s l}b^{(m)}_{k}} = -{}^{[n]}_{\alpha_s l}\delta^{(m)}_{k} \tag{40}$$

and

$$\frac{\partial^{[n]} E}{\partial^{[n]}_{\alpha_s r} b_k^{(m)}} = -{}^{[n]}_{\alpha_s r}\delta_k^{(m)} \ . \tag{41}$$

The determination of the local gradients ${}^{[n]}_{\alpha_s l}\delta_k^{(m)}$ and ${}^{[n]}_{\alpha_s r}\delta_k^{(m)}$ in the Eqs. (36) to (41) is carried out layer by layer backward through the recurrent neural network. For the output layer (M), the left and right bounds of the left and right bounds of the local gradients are determined by

$${}^{[n]}_{\alpha_s l}\delta_k^{(M)} = \left({}^{[n]}_{\alpha_s l}d_k - {}^{[n]}_{\alpha_s l}x_k^{(M)}\right) \cdot \frac{\mathrm{d}}{\mathrm{d}\nu}{}_{\alpha_s l}\varphi_k^{(M)} \left({}^{[n]}_{\alpha_s l}\nu_k^{(M)}\right) \tag{42}$$

and

$${}^{[n]}_{\alpha_s r}\delta_k^{(M)} = \left({}^{[n]}_{\alpha_s r}d_k - {}^{[n]}_{\alpha_s r}x_k^{(M)}\right) \cdot \frac{\mathrm{d}}{\mathrm{d}\nu}{}_{\alpha_s r}\varphi_k^{(M)} \left({}^{[n]}_{\alpha_s r}\nu_k^{(M)}\right) \ . \tag{43}$$

The left bound of the local gradient on α-cut s in the hidden neurons is obtained by

$${}^{[n]}_{\alpha_s l}\delta_k^{(m)} = \frac{\mathrm{d}}{\mathrm{d}\nu}{}_{\alpha_s l}\varphi_k^{(m)} \left({}^{[n]}_{\alpha_s l}\nu_k^{(m)}\right) \cdot {}^{[n]}_{\alpha_s l}F_k^{(m)} \tag{44}$$

with

$${}^{[n]}_{\alpha_s l}F_k^{(m)} = \sum_{q=1}^{Q^{(m+1)}} \begin{cases} {}^{[n]}_{\alpha_s l}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s l}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s l}x_k^{(m)} \cdot {}_{\alpha_s l}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s l}W_{qk}^{(m+1)} \\ {}^{[n]}_{\alpha_s l}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s r}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s l}x_k^{(m)} \cdot {}_{\alpha_s r}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s l}W_{qk}^{(m+1)} \\ {}^{[n]}_{\alpha_s r}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s l}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s r}x_k^{(m)} \cdot {}_{\alpha_s l}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s l}W_{qk}^{(m+1)} \\ {}^{[n]}_{\alpha_s r}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s r}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s r}x_k^{(m)} \cdot {}_{\alpha_s r}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s l}W_{qk}^{(m+1)} \end{cases} \tag{45}$$

in dependency of ${}^{[n]}_{\alpha_s l}W_{qk}^{(m+1)} = \min[.]$, see Eq. (9). The right bound of the local gradient is determined in dependency of ${}^{[n]}_{\alpha_s r}W_{qk}^{(m+1)} = \max[.]$, see Eq. (10), by

$${}^{[n]}_{\alpha_s r}\delta_k^{(m)} = \frac{\mathrm{d}}{\mathrm{d}\nu}{}_{\alpha_s r}\varphi_k^{(m)} \left({}^{[n]}_{\alpha_s r}\nu_k^{(m)}\right) \cdot {}^{[n]}_{\alpha_s r}F_k^{(m)} \tag{46}$$

with

$${}^{[n]}_{\alpha_s r}F_k^{(m)} = \sum_{q=1}^{Q^{(m+1)}} \begin{cases} {}^{[n]}_{\alpha_s l}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s l}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s l}x_k^{(m)} \cdot {}_{\alpha_s l}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s r}W_{qk}^{(m+1)} \\ {}^{[n]}_{\alpha_s l}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s r}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s l}x_k^{(m)} \cdot {}_{\alpha_s r}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s r}W_{qk}^{(m+1)} \\ {}^{[n]}_{\alpha_s r}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s l}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s r}x_k^{(m)} \cdot {}_{\alpha_s l}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s r}W_{qk}^{(m+1)} \\ {}^{[n]}_{\alpha_s r}w_{qk}^{(m+1)} \cdot {}^{[n]}_{\alpha_s r}\delta_q^{(m+1)}, & \text{if } {}^{[n]}_{\alpha_s r}x_k^{(m)} \cdot {}_{\alpha_s r}w_{qk}^{(m+1)} = {}^{[n]}_{\alpha_s r}W_{qk}^{(m+1)} \ . \end{cases} \tag{47}$$

The derivatives of the fuzzy activation functions are needed in the Eqs. (42), (43), (44), and (46). The calculation of the interval bounds of the derivatives for area hyperbolic sine and linear fuzzy activation functions are presented in (Freitag et al., 2009b).

The incremental correction of the fuzzy weights ${}^{[n]}\Delta\tilde{w}_{kj}^{(m)}$, fuzzy context weights ${}^{[n]}\Delta\tilde{c}_{ki}^{(m)}$ and fuzzy bias values ${}^{[n]}\Delta\tilde{b}_{k}^{(m)}$ calculated with Eqs. (30) to (35) can lead to improper fuzzy values. This requires to check the incremental corrections of the trainable fuzzy network parameters according to convexity, see Section 2.1, and to correct improper fuzzy values. The correction can be realized by rearranging all obtained interval bounds, as proposed e.g. in (Ishibuchi et al., 1995).

4. Verification

The developed recurrent neural network approach for fuzzy data is verified with a reference solution based on a fractional rheological material model. Time-dependent constitutive behavior of special materials, e.g. asphalt, elastomer or textile reinforced concrete can be modelled by means of fractional rheological bodies (Oeser and Freitag, 2009). Conventional rheological formulations found upon time derivatives of integer order, whereas fractional rheological formulations rely on time derivatives of real order, see Eq. (48). The fuzzy stochastic fractional NEWTON body is introduced in (Freitag et al., 2008) to model the uncertain long-term behavior of materials with fading memory. Here, the fuzzy fractional NEWTON body is used to verify the Type 3 mapping with the developed recurrent neural network approach. The differential equation of the fuzzy fractional NEWTON body is

$$\tilde{\sigma}(\tau) = \tilde{p}\,\frac{\mathrm{d}^{\tilde{r}}}{\mathrm{d}\tau^{\tilde{r}}}\,\tilde{\varepsilon}(\tau)\;. \tag{48}$$

The fuzzy stress process $\tilde{\sigma}(\tau)$ depends on the parameter $\tilde{p}$ and the fractional derivative of the strain $\tilde{\varepsilon}(\tau)$ with respect to the time τ. The operator $\tilde{r}$ represents the order of the derivative, which can adopt fuzzy values between 0 and 1. For $\tilde{r} = r = 1$, a dashpot is described with the viscosity $\tilde{p}$. The fuzzy fractional NEWTON body represents an elastic spring for $\tilde{r} = r = 0$ with the elasticity $\tilde{p}$.

The fractional differential equation (48) is solved by the LAPLACE transform. A relaxation function is obtained with the strain boundary condition $\tilde{\varepsilon}(\tau) = \tilde{\varepsilon}^{*}$. In order to account for time varying stresses, a convolution of the relaxation function can be conducted. With a time step discretization of the fuzzy strain process in equidistant time steps $\Delta\tau$, the stress in time step $[n]$ is obtained by

$${}^{[n]}\tilde{\sigma} = \sum_{i=1}^{n}\left\{\frac{\tilde{p}\cdot{}^{[i]}\Delta\tilde{\varepsilon}}{\Gamma(2-\tilde{r})\cdot\Delta\tau^{\tilde{r}}}\left[(n+1-i)^{(1-\tilde{r})}-(n-i)^{(1-\tilde{r})}\right]\right\} \tag{49}$$

for incremental linear strains. The determination of the stress in time step $[n]$ requires knowledge of the entire strain history.

The verification aims at the approximation of Eq. (49) with the developed recurrent neural networks for fuzzy data. The training and validation data are obtained by solving Eq. (49) with the fuzzy analysis, see (Möller et al., 2000). The time step length $\Delta\tau = 100$ s is chosen.

Whereas in (Freitag et al., 2009b) Type 1 and Type 2 mapping are verified, here, the Type 3 mapping of fuzzy processes, see Section 2.3, is presented. The parameters of the fuzzy fractional NEWTON body are selected as deterministic parameter p with $\tilde{p} = p = 101\,000\ \mathrm{MN\,s}^{r}/\mathrm{m}^{2}$ and as fuzzy parameter $\tilde{r}$, which is a fuzzy triangular number with $\tilde{r} =< 0.13,\ 0.14,\ 0.15 >$.

For the network training four fuzzy strain processes are used as input values. The interval bounds of two α-cuts are plotted in the left diagram of Figure 6. The number of increments is $N_{h_{tr}} = 20$ for each discretized fuzzy strain process $\tilde{\varepsilon}_{h_{tr}}(\tau)$ with $h_{tr} = 1, \ldots, 4$.

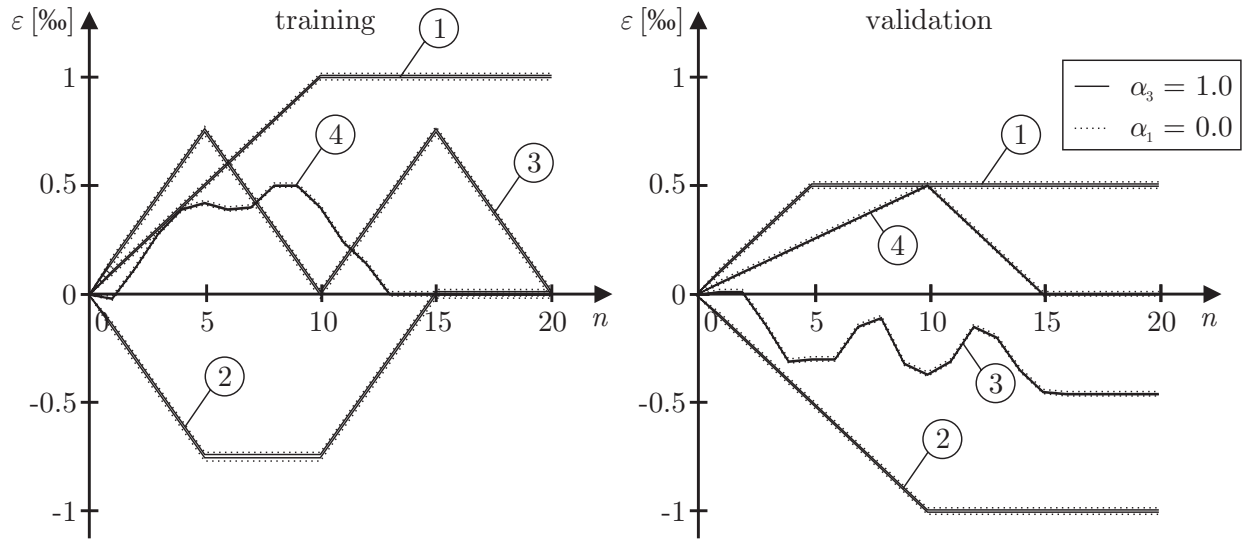

Figure 6. Fuzzy strain processes for network training and validation.

The resulting fuzzy stress processes determined with the fuzzy analysis (FA) are shown in the left diagram of Figure 7. Three α-cuts ($\alpha_1 = 0$, $\alpha_2 = 0.5$ and $\alpha_3 = 1.0$) are selected for the verification of the recurrent neural network approach.

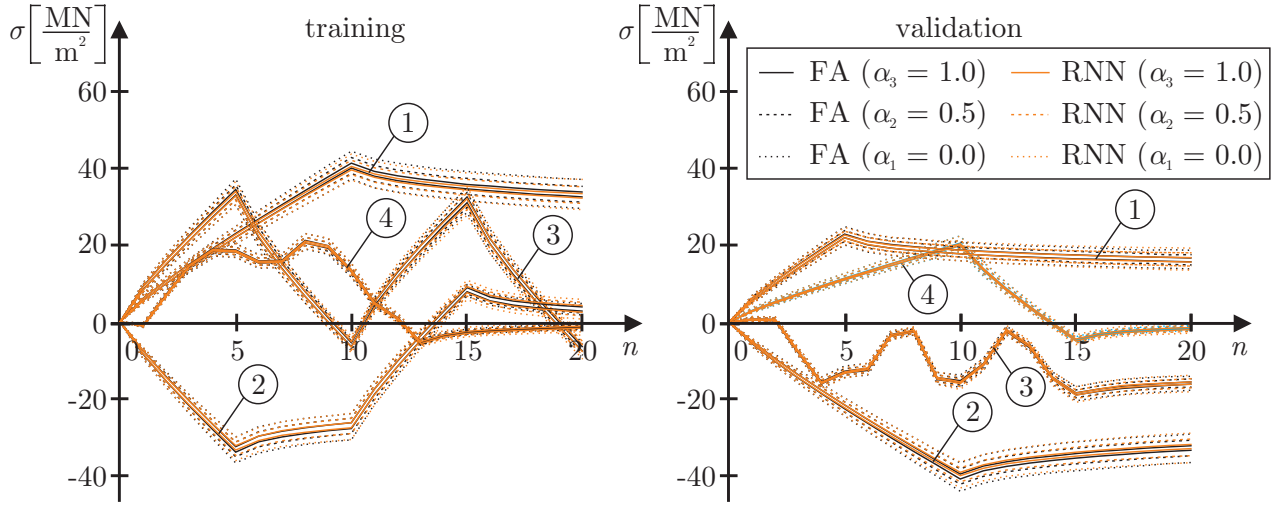

Figure 7. Fuzzy stress processes for network training and validation.

These four fuzzy strain processes and the corresponding fuzzy stress processes are used as input and output processes for Type 3 mapping with recurrent neural networks. Therefore, fuzzy network parameters are required. Here, fuzzy weights and fuzzy context weights are chosen. The recurrent

neural network used in this example consists of one input neuron, three hidden layers and one output neuron. The network architecture is summarized as $1-5-5-3-1$. As mentioned in Section 3.2, each hidden neuron and the output neuron have one context neuron. The responses of all four fuzzy strain processes determined by the neural network are displayed in the left diagram of Figure 6. The recurrent neural network predictions (RNN) show a very good agreement with the results obtained by the fuzzy analysis (FA).

The accuracy of the neural network approximation of Eq. (49) is validated with the aid of four additional fuzzy strain processes $\tilde{\varepsilon}_{h_v}(\tau)$ with $h_v = 1, \ldots, 4$, see right diagram of Figure 6. None of these fuzzy input processes has been used for network training. Hence, a comparison between the predicted network responses and the results determined by Eq. (49) for the same fuzzy strain processes are displayed in the right diagram of Figure 7. The neural network predictions of the validation processes show a very suitable agreement with the desired responses obtained by Eq. (49).

The results of the verification show that the developed recurrent neural network approach can approximate uncertain stress-strain-time-dependencies of the fuzzy fractional NEWTON body for Type 3 mapping.

5. Example

The recurrent neural network approach is applied to predict the long-term deformation of a reinforced concrete (RC) plate according to Figure 8.

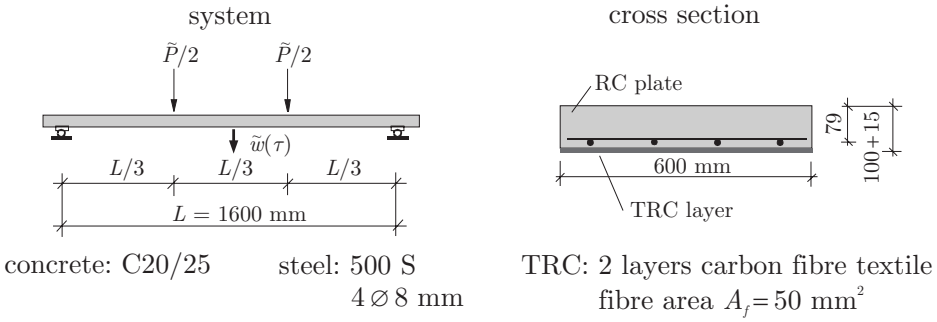

Figure 8. Geometry of the investigated plate.

The RC plate was strengthened with textile reinforced concrete (TRC) on the underside before the load has been applied. The new composite material TRC consists of fine-grained concrete with two layers of carbon fibre textiles in this application. The plate was loaded according to the scheme in Figure 8 by means of steel blocks, see (Weiland et al., 2006). As the dead loads of the steel blocks and additionally used slats are not precisely known, the loading is modelled by means of a fuzzy process

$$\tilde{P}(\tau) = \begin{cases} 0, & \text{if } \tau = 0 \\ < 44.5,\ 45.4,\ 46.3 > \text{ kN}, & \text{if } \tau > 0 \end{cases} \tag{50}$$

which is a time-independent fuzzy triangular number for $\tau > 0$.

Here, an adaptive prediction on the basis of experimental data is presented. In a first prediction, data measured during a period of 240 days are used. In a second prediction, the network is updated with additional data measured during a period of 286 days. The time-dependent displacement $\tilde{w}(\tau)$ in gravity direction in the middle of the plate was monitored with the help of measurements. Additionally, the temperature and the humidity next to the plate were recorded. The measured data are discretized in constant time-steps $\Delta\tau$ with a time-step length of 1 day for the processing in the recurrent neural network. The values at each point in time are modelled as fuzzy triangular numbers. The left bound of the support (α-cut $\alpha_1 = 0$) is obtained in time step $[n]$ as the minimal measured value of the period $\left[{}^{[n-1]}\tau,\ {}^{[n]}\tau\right]$. The maximal measured value is considered as right bound of the support. The measured value of ${}^{[n]}\tau$ is assumed as kernel value (α-cut $\alpha_2 = 1.0$) of the fuzzy triangular number. The resulting fuzzy processes of the temperature $\tilde{T}(\tau)$, the relative humidity $\tilde{h}(\tau)$, and the displacement $\tilde{w}(\tau)$ are plotted in the Figures 9, 10 and 11 indicated by their trajectories $\alpha_1 = 0$ and $\alpha_2 = 1.0$.

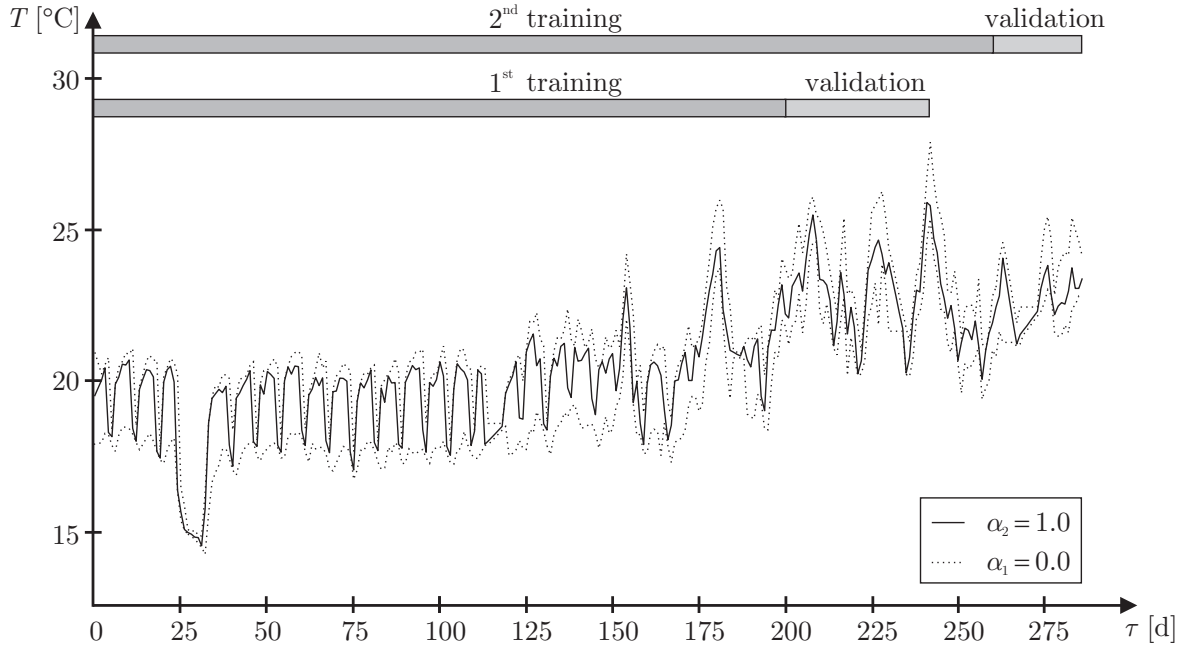

Figure 9. Temperature process $\tilde{T}(\tau)$.

The structure is subjected to the discretized fuzzy processes ${}^{[n]}\tilde{P}$, ${}^{[n]}\tilde{T}$ and ${}^{[n]}\tilde{h}$. The three corresponding fuzzy triangular numbers are the input values of a recurrent neural network in each time step. The output of the network yields a prediction of the displacement ${}^{[n]}\tilde{w}$ under consideration of the whole stress history. A suitable network architecture is found by an adaptive approach. Starting with one hidden layer, neurons and layers are added while monitoring the training and validation errors. The final recurrent neural network consists of three hidden layers. The network architecture is summarized as $3-6-4-4-1$. A context neuron is assigned to each hidden neuron and to the output neuron.

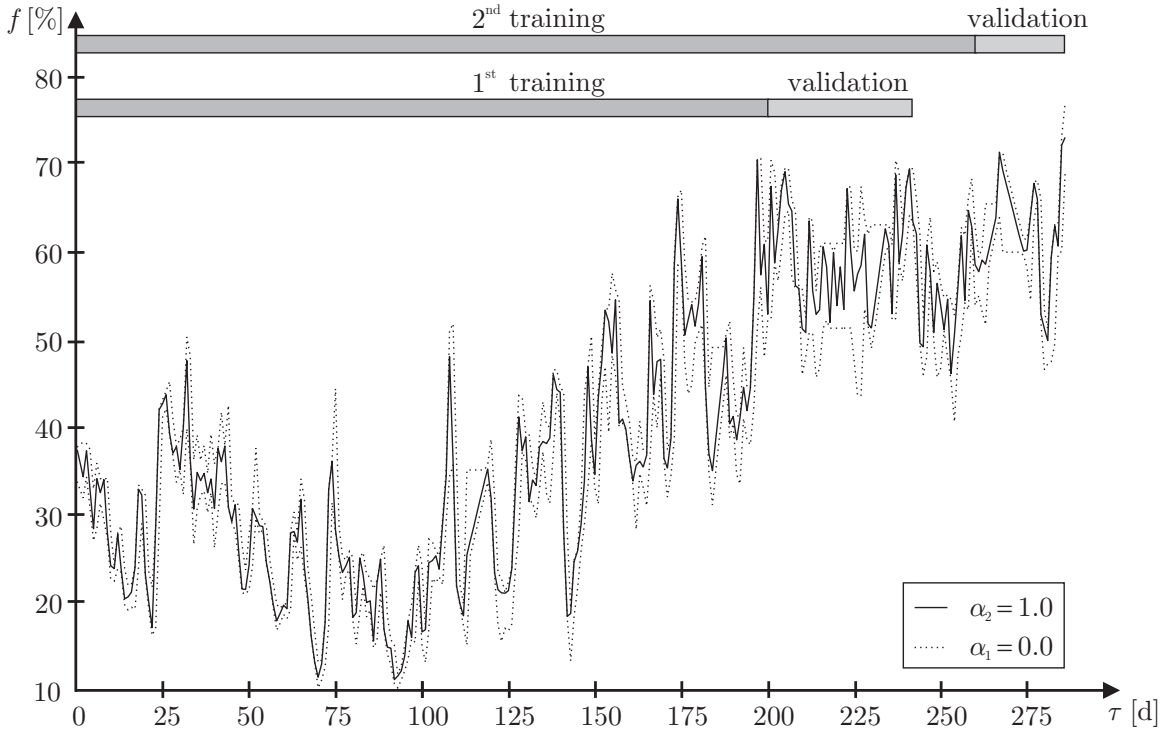

Figure 10. Humidity process $\tilde{h}(\tau)$.

The first training data set contains 200 time steps. The following 40 time steps are separated for the network validation. In Figure 11, the desired response and the network prediction after the first training and validation is presented. A good agreement is evidently between the desired response (measurement series) and the first prediction of the recurrent neural network.

The recurrent neural network is updated with additional data for a second prediction (adaptive prediction). The second training and validation data sets contain 260 and 26 time steps respectively. The second network prediction is presented in Figure 11, too. The membership functions of the measured and predicted fuzzy displacements $\tilde{w}(\tau = 260\ \mathrm{d})$ displayed in Figure 11 are obtained as linear interpolation of the results for $\alpha_1 = 0$ and $\alpha_2 = 1.0$.

Measured data of the temperature and the humidity are available for 314 days. Here, a simple periodic repetition of the fuzzy temperature and humidity processes are assumed. Predictions of the fuzzy displacement process $\tilde{w}(\tau)$ are shown for 480 time steps (first prediction) and 1000 time steps (second prediction) in Figure 11.

For future applications, the prediction of the input time series $\tilde{T}(\tau)$ and $\tilde{h}(\tau)$ can also be realized with forecasting methods for fuzzy data, see (Möller and Reuter, 2007). Additionally, time varying loads can be considered. In this case, the data basis has to be extended to a number of different stress scenarios for network training and validation.

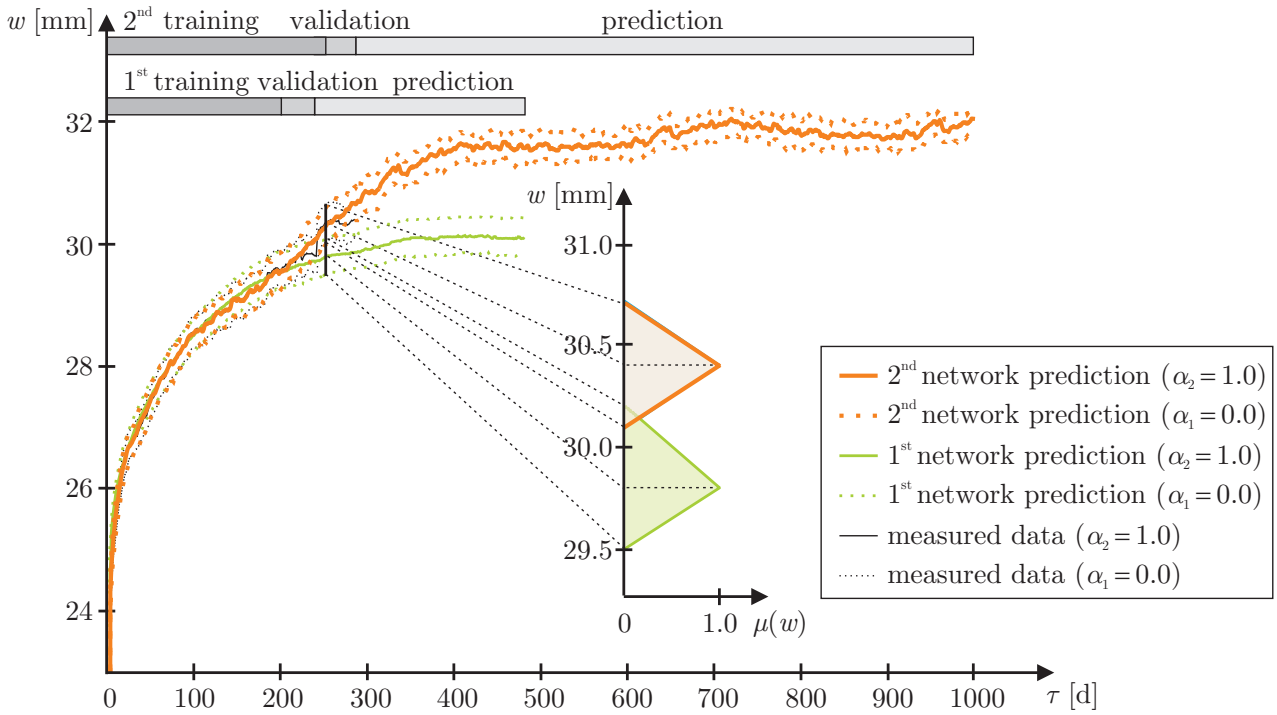

Figure 11. Displacement $\tilde{w}(\tau)$, measured data and adaptive neural network prediction.

6. Conclusion

For identification and prediction of time-dependent structural behavior, a new approach based on artificial neural networks is presented. Recurrent neural networks for fuzzy data enable the processing of uncertain temporal sequences. Three types of mapping fuzzy processes are described. For Type 2 and Type 3 mapping fuzzy network parameters are required. New prediction and the training algorithms for a priori defined and trainable fuzzy network parameters are introduced. The developed algorithms can be applied for the numerical prediction of fuzzy processes, e.g. fuzzy structural response processes caused by structural action processes. Beside fuzzy processes, also interval and deterministic processes can be considered within the recurrent neural network approach. Because of this feature and due to the independence of specific construction materials, the new model-free approach has a high degree of generality and flexibility. The approach has been verified by a fractional rheological material model. The results of a numerical analysis based on the fuzzy fractional NEWTON body and the model-free recurrent neural network prediction show an excellent agreement. A possible application is presented by an example. The long-term deformation behavior of a TRC strengthened reinforced concrete plate is investigated. Thereby, the fuzzy displacement process of the plate is predicted with recurrent neural networks for fuzzy data by an adaptive approach. In future works, the developed approach can be applied for the approximation of the long-term constitutive material behavior. Recurrent neural networks trained by experimentally or numerically obtained fuzzy processes can by considered as uncertain time-dependent material models in finite element analyses for robust design of engineering structures.

Acknowledgements

The authors gratefully acknowledge the support of the German Research Foundation (DFG) within the framework of the Collaborative Research Centre 528 entitled "Textile Reinforcements for Structural Strengthening and Repair".

References

Adeli, H. Neural Networks in Civil engineering: 1989-2000. *Computer-Aided Civil and Infrastructure Engineering*, 16:126–142, 2001.

Beer, M. Model-free sampling. *Structural Safety*, 29:49–65, 2007.

Elman, J. L. Finding Structure in Time. *Cognitive Science*, 14:179–211, 1990.

Freitag, S., M. Beer, W. Graf, and M. Kaliske. Lifetime prediction using accelerated test data and neural networks. *Computers & Structures*, 87:1187–1194, 2009.

Freitag, S., W. Graf, M. Kaliske, and J.-U. Sickert. Prediction of Structural Behaviour with Recurrent Neural Networks for Fuzzy Data. In B. H. V. Topping and Y. Tsompanakis, editors, *Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, paper 28, Funchal, 2009. Civil-Comp Press, Stirlingshire.

Freitag, S., W. Graf, S. Pannier, and J.-U. Sickert. Reliability of Structures under Consideration of Uncertain Time-Dependent Material Behaviour. In D. Dubois, M. A. Lubiano, H. Prade, M. A. Gil, P. Grzegorzewski, and O. Hryniewicz, editors, *Soft Methods for Handling Variability and Imprecision*, pp. 383–390. Springer-Verlag, Advances in Soft Computing, 48, Berlin, 2008.

Graf, W., S. Freitag, M. Kaliske, and J.-U. Sickert. Recurrent neural networks for uncertain time-dependent structural behaviour. *Computer-Aided Civil and Infrastructure Engineering*, in press.

Haykin, S. *Neural Networks – A Comprehensive Foundation*. Prentice-Hall, Upper Saddle River, 1999.

Ishibuchi, H., K. Morioka, and I. B. Turksen. Learning by Fuzzified Neural Networks. *International Journal of Approximate Reasoning*, 13:327–358, 1995.

Johanyák, Z. C. and S. Kovács. Distance based similarity measures of fuzzy sets. In *Proceedings of 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence*, Herl'any, 2005.

Jordan, M. I. Attractor dynamics and parallelism in a connectionist sequential machine. In J. Diederich, editor, *Artificial neural networks: concept learning*, pp. 112–127, IEEE Press, Piscataway, 1990.

Möller, B. and M. Beer. *Fuzzy Randomness – Uncertainty in Civil Engineering and Computational Mechanics*. Springer-Verlag, Berlin, 2004.

Möller, B. and M. Beer. Engineering computation under uncertainty – Capabilities of non-traditional models. *Computers & Structures*, 86(10):1024–1041, 2008.

Möller, B., W. Graf, and M. Beer. Fuzzy structural analysis using α-level optimization. *Computational Mechanics*, 26(6):547–565, 2000.

Möller, B. and U. Reuter. *Uncertainty Forecasting in Engineering*. Springer-Verlag, Berlin, 2007.

Moore, R. E. *Methods and Applications of Interval Analysis*. SIAM, Studies in Applied Mathematics, 2, Philadelphia, 1979.

Oeser, M. and S. Freitag. Modeling of materials with fading memory using neural networks. *International Journal for Numerical Methods in Engineering*, 78(7):843–862, 2009.

Weiland, S., R. Ortlepp, and M. Curbach. Strengthening of predeformed slabs with textile reinforced concrete. In *Proceedings of the 2nd fib-Congress*, Naples, 2006.

Zell, A. *Simulation Neuronaler Netze*. Addison-Wesley, Bonn Paris, 1996.