NN-SANARX model based control of a multi tank liquid-level system

Juri Belikov [1], **Eduard Petlenkov** [2]

[1] *Institute of Cybernetics, Tallinn University of Technology,*
Akadeemia tee 21,
Tallinn, 12618, Estonia

E-mail: jbelikov@cc.ioc.ee

[2] *Department of Computer Control, Tallinn University of Technology,*
Ehitajate tee 5,
Tallinn, 19086, Estonia

E-mail: eduard.petlenkov@ttu.ee

Received 20 March 2014

Accepted 15 October 2014

Abstract

This paper is devoted to application of neural network with a specific restricted connectivity structure for identification and control of a real process. Parameters of the identified model are used to design a controller based on dynamic feedback linearization. The designed neural network based controller is verified on mathematical model within MATLAB/Simulink environment and applied to the real-time control of a laboratory plant. Water tank system was chosen as a case study to illustrate applicability of the proposed approach. Several experimental results as well as comparison with some alternative methods are provided.

Keywords: Nonlinear Control Systems, Computational Intelligence, Neural Networks, Feedback Linearization.

1. Introduction

In this paper, we explore the abilities of the so-called Neural Networks based Simplified Additive Auto Regressive eXogenous (NN-based SANARX) models [1] to be used in industrial applications. Though we decided to focus on a water tank system, in principle, the proposed approach can be applied to control of more complex dynamical systems. NN-based SANARX is a subclass of a more general ANARX model class [2] and inherits all advantages of its parent class. The most important from the control point of view is that such models are always linearizable by dynamic output feedback. The latter means that it is always possible to write down equations of the linearizing feedback. In this light it is attractive to use models of this class in different real-life or academical applications. However, in order to apply linearizing feedback, it is necessary to calculate corresponding control signal from equations of a controller. This in turn appeared to be a difficult task as was indicated in [3]. Therefore, several techniques were proposed to solve the problem. In [4] Newton-Raphson method was applied to backing up control of a truck trailer. The method based on additional static neural network was considered in [5]. In [6] Taylor series expansion method was employed. However, the most simple and reliable technique among developed appeared to be the method based

Co-published by Atlantis Press and Taylor & Francis

on the idea of simplification of the first sublayer in a neural network (mentioned NN-SANARX), reducing the overall problem to the solution of system of linear equations. So far the developed algorithm has found its application in smart buildings' climate control (contract with YOGA intelligent building `http://www.yogasystems.com/`).

As a case study the water tank (multi tank) system prototype is chosen. Though the problem of liquid level control in a tank is not new, it still has not lost its actuality. Level regulators are used in industry to maintain a constant fluid pressure, or a constant fluid supply to a process, or in a waste storage [7]. The common examples of possible industrial applications include chemical industry and food processing [8] as well as different irrigation systems like dams, water reservoirs, etc. Through the years various techniques have been used to tackle the problem.

The application and extensive discussion of classical techniques is shown in [9], where five different PID controllers (Ziegler-Nichols, Integral of Time-weighted Absolute Error, Internal Model Control, poles placement and dual loop) were designed and analyzed. Recent analytic solutions employ tensor product based methods [10] and decoupling control [11]. Slightly more advanced techniques based on fractional-order calculus are presented in [12] and [13], where PI and PID controllers were designed, respectively. Traditionally, accurate mathematical model-based strategies have been applied to deal with control problems. However, systems become more and more complex as well as requirements tightened (energy efficiency, reliability, flexibility, etc.). Thus, there is a clear message that more advanced techniques need to be introduced. Recently, methods based on computational intelligence have started to gain popularity and are applied either solely or in combination with some classical techniques [14]. There are numerous papers addressing application of computational intelligence based techniques to control the liquid-level system. An adaptive model-based reference fuzzy controller for controlling the water level in a water tank was discussed in [15]. In [16] a backpropagation neural network algorithm was used to adjust parameters of the PID controller. The authors of [17] utilized Sugeno fuzzy model as the model structure for a linear model based predictive control of the liquid level.

The present contribution may be seen in application of the classical control technique (linearization via dynamic output feedback) and neural networks based modeling to control a water level in a tank system. In the paper we describe all design steps: starting from collecting the input-output (i/o) data of a process and finishing with implementation and test of the synthesized controller on a real plant. One of the most complicated parts of the overall design procedure was solved using feedback linearization technique. Like any other analytic method the linearization by dynamic output feedback provides one with equations of a controller, whose coefficients are taken directly from the identified model. In other words, selecting NN-SANARX model, one merges together numeric parts of the modeling and control synthesis. To the best knowledge of the authors, there is no any similar research made on the application of NN-SANARX models to control of real plants. Therefore, the research, presented in the paper, can be seen as a preliminary step toward real industrial application.

The paper is organized as follows. In Section 2 we give a brief overview of ANARX type structures, the main idea of linearization via dynamic output feedback, and the neural network formalism. The next section is devoted to the neural networks based controller design with a brief explanation of extension on the case of multi-input multi-output systems. In Section 4 a detailed step-by-step algorithm for constructing the feedback linearizing controller is presented. The mathematical model of the plant as well as experimental results are presented in the next section. Discussion and concluding remarks are drawn in the last section.

2. Preliminaries

Hereinafter, in order to make expression of the material visually more compact, if $\xi : \mathbb{Z} \to \mathbb{R}$ and $k \in \mathbb{N}$, then $\xi^{[k]}$ stands for kth-step forward time shift of ξ and is defined by $\xi^{[k]} := \xi(t+k)$. Similarly for backward shift if $k \in \mathbb{Z}^-$. Note that $\xi := \xi^{[0]} = \xi(t)$. The nonlinear control systems are typically represented either by the higher order single-input single-output (SISO) difference equation

$$y^{[n]} = \varphi\left(y, y^{[1]}, \ldots, y^{[n-1]}, u, u^{[1]}, \ldots, u^{[n-1]}\right), \quad (1)$$

or by the state equations

$$\begin{aligned} x^{[1]} &= f(x,u) \\ y &= h(x), \end{aligned} \quad (2)$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the output; φ, f and h are real analytic functions.

The system, represented by (1), is known in the literature as a discrete-time Nonlinear AutoRegressive eXogenous (NARX) model. This model can be used to identify a wide class of complex processes with a high degree of accuracy, see [18]. However, from the control point of view, models of the form (1) have several drawbacks. The most important for the practice is linearizability by dynamic output feedback. In general, this property does not hold for models with NARX structure, see [19] for details. Therefore, the Additive NARX (ANARX) structure

$$\begin{aligned} y^{[n]} = f_1\left(y^{[n-1]}, u^{[n-1]}\right) + \cdots \\ + f_1\left(y^{[1]}, u^{[1]}\right) + f_n(y,u) \end{aligned} \quad (3)$$

was introduced in [2]. This is a modification/subclass of the NARX model (1) with separated time instances. The statistical analysis, presented in [20], indicated that in some situations such simplification is rather advantage than restriction. Model (3) can always be linearized using the following dynamic output feedback

$$\begin{aligned} \eta_1 &= f_1(y,u) \\ \eta_1^{[1]} &= \eta_2 - f_2(y,u) \\ &\vdots \\ \eta_{n-2}^{[1]} &= \eta_{n-1} - f_{n-1}(y,u) \\ \eta_{n-1}^{[1]} &= v - f_n(y,u), \end{aligned} \quad (4)$$

where $\eta \in \mathbb{R}^{n-1}$ is a state vector of the controller and $v \in \mathbb{R}$ is a reference signal (desired output).

The linearizability by dynamic output feedback means that it is always possible explicitly to construct a controller once the model of the form (3) was obtained. Another important advantage of ANARX model is that it can be directly represented via state equations

$$\begin{aligned} x_1^{[1]} &= x_2 + f_1(x_1, u) \\ &\vdots \\ x_{n-1}^{[1]} &= x_n + f_{n-1}(x_1, u) \\ x_n^{[1]} &= f_n(x_1, u) \\ y &= x_1. \end{aligned}$$

State-space form is an important property of the control system, which provides a convenient and compact way for its further modeling and analysis. The latter can be used, for example, to design state controller [21].

2.1. Neural Networks based Simplified ANARX model

In order to perform analysis and design of the appropriate controller for the process, one is usually interested in mathematical equations rather than in the black-box description. In fact, one can derive the model from the first principles, relying on the Newton equations. However, most likely in many cases such an approach will result in a quite complex model. Thus, one may start from the measured data and identify relations between variables. One of the most popular approaches consists in employing Neural Networks (NN) formalism. Thus, the theory developed for ANARX models was adopted to the case of neural networks in [2], [22]. To be more specific, NN-ANARX model can be represented as

$$y = \sum_{i=1}^{n} C_i \phi_i \left(W_i \left[y^{[-i]} \quad u^{[-i]}\right]^{\mathrm{T}}\right), \quad (5)$$

where i stands for the number of sublayer, ϕ_i is an activation function, C_i and W_i are $1 \times l_i$ and $l_i \times 2$ dimensional matrices of the output and input synaptic weights, respectively. In addition, l_i is the number of hidden neurons. A schematic diagram of the neural network, representing ANARX structure, is depicted in Fig. 1.

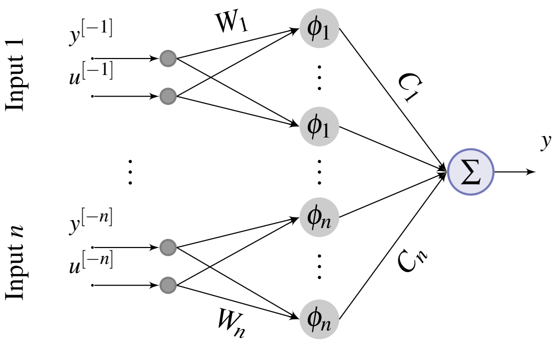

Fig. 1. Neural network with ANARX structure.

In order to avoid additional notation, we need the following technical assumption. Next, we will show that it is not restrictive and can always be incorporated into the control system.

Assumption 1 *ANARX model perfectly describe the process meaning that $\widehat{y} = y$ with $\widehat{y}$ being the output of the neural network.*

3. NN-based controller design

Note that the application of the feedback linearization algorithm to ANARX model leads to the finite discrete-time linear closed-loop system described by the equation $y^{[n]} = v$ or equivalently transfer function $Y(z)/V(z) = 1/z^n$. However, for the real-life applications poles of the transfer function corresponding to such a model can be too *fast* resulting in an undesirable behavior of the control system (for example, significant overshootings, high control signals, etc.). Therefore, an algorithm based on a more general reference model can be used instead. In this case a closed-loop system can be described as

$$\begin{aligned} y + a_1 y^{[-1]} + \cdots + a_n y^{[-n]} \\ = b_1 v^{[-1]} + \cdots + b_n v^{[-n]}, \end{aligned} \qquad (6)$$

where $a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{R}$ are parameters of the reference model. Observe that (3) is a linear discrete-time reference model that can be used to predefine dynamics of the closed-loop system. Thus, the issue with undesirable behavior can be solved selecting appropriate zeros and poles. As a result, the dynamic output feedback (4) can be rewritten using parameters of the neural network as

$$\eta_1 = a_1 y - b_1 v + C_1 \phi_1 \left(W_1 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \qquad (7)$$

and

$$\begin{aligned} \eta_1^{[1]} &= \eta_2 + b_2 v - a_2 y - C_2 \phi_2 \left(W_2 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \\ &\vdots \\ \eta_{n-2}^{[1]} &= \eta_{n-1} + b_{n-1} v - a_{n-1} y \\ &\quad - C_{n-1} \phi_{n-1} \left(W_{n-1} \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \\ \eta_{n-1}^{[1]} &= b_n v - a_n y - C_n \phi_n \left(W_n \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right). \end{aligned} \qquad (8)$$

Next, we formulate the condition of equivalence between the closed-loop system and the reference model.

Proposition 1. *Application of* (7) *and* (8) *to* (5) *results in* (3).

Proof. Applying the dynamic output feedback (7) and (8) to NN-ANARX model (5), we get

$$\begin{aligned} a_1 y^{[-1]} - b_1 v^{[-1]} + C_1 \phi_1 \left(W_1 \begin{bmatrix} y^{[-1]} & u^{[-1]} \end{bmatrix}^{\mathrm{T}} \right) \\ = b_n v^{[-n]} - a_n y^{[-n]} + \cdots + b_2 v^{[-2]} - a_2 y^{[-2]} \\ - \sum_{i=2}^{n} C_i \phi_i \left(W_i \begin{bmatrix} y^{[-i]} & u^{[-i]} \end{bmatrix}^{\mathrm{T}} \right). \end{aligned} \qquad (9)$$

Next, regrouping terms in (9) yields

$$\begin{aligned} a_1 y^{[-1]} + \cdots + a_n y^{[-n]} \\ + \sum_{i=1}^{n} C_i \phi_i \left(W_i \begin{bmatrix} y^{[-i]} & u^{[-i]} \end{bmatrix}^{\mathrm{T}} \right) \\ = b_1 v^{[-1]} + b_2 v^{[-2]} + \cdots + b_n v^{[-n]}. \end{aligned} \qquad (10)$$

Finally, one can easily see that the summation operator on the left-hand side of (10) is exactly the right-hand side of (5). Therefore, (10) can be rewritten as

$$\begin{aligned} y + a_1 y^{[-1]} + \cdots + a_n y^{[-n]} \\ = b_1 v^{[-1]} + \cdots + b_n v^{[-n]}. \end{aligned} \qquad (11)$$

It remains to stress that (11) describes both reference model and closed-loop . □

Remark 1. The control technique presented in [1] is a special case of (7), (8) for $b_1 = \cdots = b_{n-1} = a_1 = \cdots = a_n = 0$ and $b_n = 1$.

In order to simplify calculation of the control signal defined by (7), we assume like in [1] that ϕ_1 is a linear function, resulting in a structure of the neural network known as an NN-based Simplified ANARX model. Then (7) can be rewritten as

$$u = T_2^{-1}(\eta_1 - (T_1 + a_1)y + b_1 v), \qquad (12)$$

where T_1 and T_2 are the first and second elements of the vector $C_1 W_1$, respectively. Note that T_2 cannot be equal to zero. This fact has to be taken into account during identification procedure. The overall structure of the corresponding control system is represented in Fig. 2.

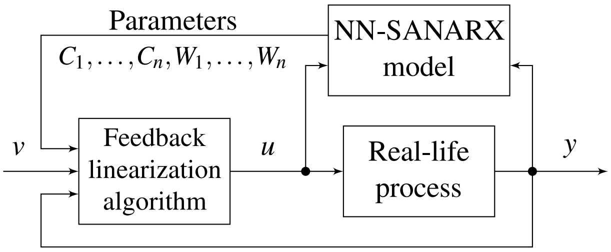
Parameters
$C_1,\dots,C_n,W_1,\dots,W_n$
NN-SANARX model
v
Feedback linearization algorithm
u
Real-life process
y

Fig. 2. General scheme of a control system.

3.1. Comments on the extension to MIMO case

Recall that in the sections above to simplify the exposition of the material we focused on the SISO case. However, the same mathematical tools can be adopted to the case of multi-input multi-output (MIMO) systems in a relatively straightforward manner. To be more precise, the scalar signals in (1) and (2) have to be replaced by the corresponding vectors of output, input, and reference signals as $y := [y_1, \dots, y_p]^{\mathrm{T}} \in \mathbb{R}^p$, $u := [u_1, \dots, u_m]^{\mathrm{T}} \in \mathbb{R}^m$, and $v := [v_1, \dots, v_p]^{\mathrm{T}} \in \mathbb{R}^p$, respectively. Then, in the neural networks based description (and so forth) of ANARX model (5) matrices of synaptic weights C_i and W_i will have $p \times l_i$ and $l_i \times (p+m)$ dimensions. Finally, it is important to explain in which way the feedback linearization algorithm based on the reference model is different in MIMO case. Let us rewrite equation (3) as follows

$$\begin{aligned} y + A_1 y^{[-1]} + \cdots + A_n y^{[-n]} \\ = B_1 v^{[-1]} + \cdots + B_n v^{[-1]}, \end{aligned} \qquad (13)$$

where $A_1, \dots, A_n$ and $B_1, \dots, B_n$ are square matrices with real valued coefficients defined as

$$A_i := \begin{bmatrix} a_{1i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{pi} \end{bmatrix}, \quad B_i := \begin{bmatrix} b_{1i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_{pi} \end{bmatrix}$$

for $i = 1, \dots, n$.

Consider the matrix of reference models, represented by linear discrete-time transfer functions

$$G(z) = \begin{bmatrix} \frac{\sum_{i=1}^n b_{1i} z^{n-i}}{z^n + \sum_{i=1}^n a_{1i} z^{n-i}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\sum_{i=1}^n b_{pi} z^{n-i}}{z^n + \sum_{i=1}^n a_{pi} z^{n-i}} \end{bmatrix}. \qquad (14)$$

Then Proposition 1 can be extended as follows.

Proposition 2. *The closed-loop system consisting of nonlinear MIMO system* (5) *and compensator*

$$\begin{aligned} \eta_1 &= C_1 \phi_1 \left(W_1 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) + A_1 y - B_1 v \\ \eta_1^{[1]} &= \eta_2 + B_2 v - A_2 y - C_2 \phi_2 \left(W_2 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \\ &\vdots \\ \eta_{n-2}^{[1]} &= \eta_{n-1} + B_{n-1} v - A_{n-1} y \\ &\quad - C_{n-1} \phi_{n-1} \left(W_{n-1} \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \\ \eta_{n-1}^{[1]} &= B_n v - A_n y - C_n \phi_n \left(W_n \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}} \right) \end{aligned}$$

is equivalent to the matrix of reference models (14).

Proof. The proof follows the same lines as in the case of Proposition 1. □

If we assume NN-SANARX model in our calculations, then

$$\begin{aligned} y = D \begin{bmatrix} y^{[-1]} & u^{[-1]} \end{bmatrix}^{\mathrm{T}} \\ + \sum_{i=2}^{n} C_i \phi_i \left(W_i \begin{bmatrix} y^{[-i]} & u^{[-i]} \end{bmatrix}^{\mathrm{T}} \right), \end{aligned} \qquad (15)$$

where

$$D := C_1 W_1 = \begin{bmatrix} D_1 & D_2 \end{bmatrix}^{\mathrm{T}}$$

is $p \times (p+m)$-dimensional matrix. Here matrix $D_1 \in \mathbb{R}^{p\times p}$ stands for coefficients of outputs $y^{[-1]}$ and matrix $D_2 \in \mathbb{R}^{p\times m}$ stands for coefficients of inputs $u^{[-1]}$. Furthermore, the vector of control signals u can be calculated as

$$u = D_2^{-1}\left(\eta_1 + B_1 v - (D_1 + A_1)y\right).$$

4. Algorithm and software implementation

Next, we provide a basic idea about implementation of the theory presented in the previous sections. The algorithm can be implemented in any programming language supporting neural networks formalism.
Algorithm:

Step 1. Collect training data performing real-life experiment. This has to be data collected from real measurements. Sometimes it is hard to obtain such samples. In such cases on the preliminary step to describe the process one may rely on the approximate mathematical model. Split collected data into two sets S_1 and S_2.

Step 2. Use *a priori* available information about the process to determine important parameters of the neural network such as order of the identified model, number of sublayers, etc.

Step 3. Create initial guess NN model with chosen parameters and predefined structure depicted in Fig. 1 with linear first sublayer (corresponding to NN-SANARX structure).

Step 4. Use data from the set S_1 to train the neural network with any available training algorithm like Levenberg-Marquardt, different variations of gradient descent algorithms, etc.

Step 5. Verify obtained model using the second set S_2 with collected i/o data. If the performance is satisfactory, then go to Step 6; otherwise, go back to Step 3 and repeat training procedure.

Step 6. According to the prespecified control requirements, write down equations of the control signal (12) and dynamics of controller (8).

Step 7. Put plant and controller into the closed-loop and verify the designed control system.

Step 8. End of the Algorithm.

Fig. 3 illustrates the basic idea of the Algorithm described above.

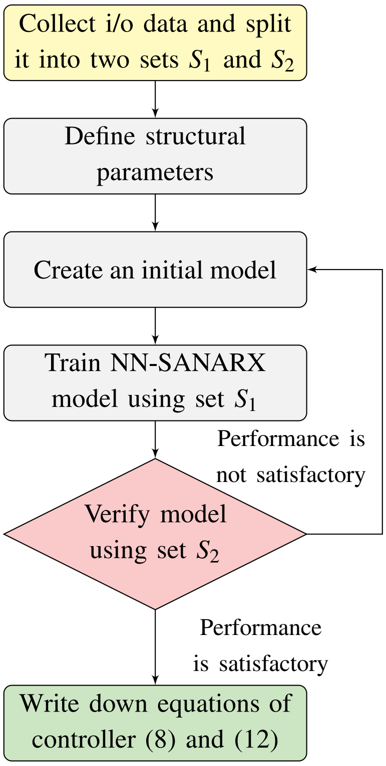
Collect i/o data and split it into two sets S_1 and S_2
Define structural parameters
Create an initial model
Train NN-SANARX model using set S_1
Performance is not satisfactory
Verify model using set S_2
Performance is satisfactory
Write down equations of controller (8) and (12)

Fig. 3. Schematic diagram of the algorithm.

5. Experimental results

All the real-life experiments were performed on the equipment available in the laboratory at the Department of Computer Control, Tallinn University of Technology, see [23].

5.1. *Mathematical model of a multi tank system*

The graphical representation of a multi tank system is borrowed from the user's manual, provided by

[24], and depicted in Fig. 4.

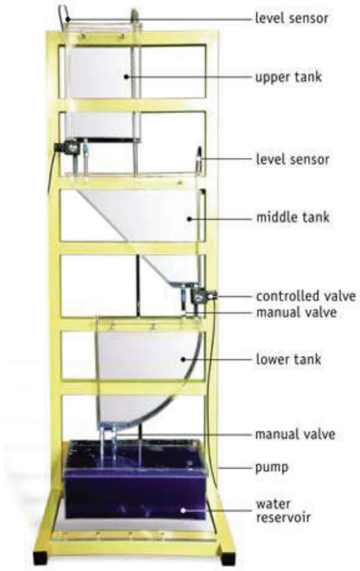

Fig. 4. Model of the multi tank system.

The physical meanings of parameters are listed in Table 1, where *i* indicates the number of a tank.

Table 1. Nomenclature.

Parameters	Physical description
x_i	fluid level in the *i*th tank
w	width of a tank
a	length of the upper tank
b,c	lengths of the upper and lower part of the middle tank
R	length of the upper part of the lower tank
h	height of the middle tank
C_i	resistance of the output orifice of the *i*th tank
α_i	flow coefficient for the *i*th tank

The numerical values of the parameters from Table 1 can be formally divided into two groups. Part of them have constant values (units are given in meters):

$$a = 0.25, b = 0.345, c = 0.1, R = 0.364,$$
$$h = 0.35, w = 0.035.$$

Note that the maximum height of each tank is 0.35m. However, the maximum reachable height may vary with respect to safety requirements and experimental configuration. In addition, state variables and control signals have natural saturations due to the physical limitations of the system. Note that for the laminar flow the outflow rate from a tank is governed by the Bernoulli's law that corresponds to the case $\alpha_i = 1/2$. In fact, this is a typical assumption made in the academic research. However, in case of real process such issues like turbulence and acceleration of the liquid in the tube cannot be usually neglected. Therefore, in more general cases $\alpha_i \in (0, 0.5] \subset \mathbb{R}$ has to be assumed.

The multi tank system is equipped with valves and level sensors for each tank. The upper tank has a constant cross-sectional area. However, the middle and lower tanks have variable (conic and spheral) cross-sectional areas. The system is equipped with direct current (DC) pump providing liquid transportation from the lowest water reservoir to the upper tank. The pump is supplied from the power interface by an appropriate pulse-width modulation (PWM) control signal. Each pair (automatic and manual) of valves between tanks can be separately controlled changing this way the output flow and, if necessary, the number of inputs and outputs of the system. Thus, the system can be reconfigured with respect to the prespecified requirements. The plant is designed to operate with an external PC-based digital controller. The computer communicates with the level sensors, automatic valves and pump by a dedicated i/o board and the power interface. The i/o board is controlled by the real-time software which operates in Simulink on the basis of MATLAB Real-Time Windows Target environment.

Due to the flexible structure of the system, numerous different combinations are possible. For more specific details and assumptions made for the model, we refer the interested reader to the manual available at [24].

5.2. Pump-controlled scenario

In order to control the water level in a tank, we use settings listed in Table 2 for our experiments.

Table 2. Pump-controlled scenario: configuration.

Tank #	Pump	Manual valve	Automatic valve
1	varies	fully opened	closed
2		fully opened	closed
3		fully opened	closed

This configuration leads to the so-called pump-controlled version of the system in which pump is used as a generator of the control action (input signal). In this case, the differential equations, describing the dynamics of the system, can be derived, assuming the laminar outflow rate of an *ideal fluid* from a tank, by means of mass balance as

$$\begin{aligned}\dot{x}_1 &= \frac{1}{aw}\left(u - r_1 x_1^{\alpha_1}\right)\\ \dot{x}_2 &= \frac{h}{cwh + bwx_2}\left(r_1 x_1^{\alpha_1} - r_2 x_2^{\alpha_2}\right)\\ \dot{x}_3 &= \frac{1}{w\sqrt{R^2 - (R - x_3)^2}}\left(r_2 x_2^{\alpha_2} - r_3 x_3^{\alpha_3}\right),\end{aligned} \tag{16}$$

where r_i is a resistance of the output orifice of the manual valve in the ith tank. In principle, such configuration allows simultaneous control of water levels in several tanks. However, this type of control will barely be illustrative.

Suppose we are interested in a control of the water level in the first tank. Hence, a single-input single-output system can be obtained from equations (16) with inflow and the water level in the first tank being the input u and the output $y := x_1$. The physical parameters of the plant have the following numerical values $w = 0.035$m, $a = 0.25$m, and the maximum inflow provided by the pump is $1.0284 \cdot 10^{-4}\mathrm{m}^3/\mathrm{s}$. Note that the input signal can be defined as $u : \mathscr{U}_1 \to \mathscr{U}_2$, where $\mathscr{U}_1 = \{u(t) \in \mathbb{R} \mid 0 \leqslant u(t) \leqslant 1.2394 \cdot 10^{-4}\}$ and $\mathscr{U}_2 = \{u(t) \in \mathbb{R} \mid 0 \leqslant u(t) \leqslant 1\}$. In fact, it is a necessary transformation that allows us to simplify the numerical calculations.

The identification data was collected from the real plant with sampling time 0.5s. The collected i/o data was used to train NN-SANARX structure by means of gradient descent with adaptive learning rate backpropagation algorithm. The network shown in Fig. 1 was trained with two sublayers, corresponding to the second order ($n = 2$) of the model, and 3 neurons on each sublayer, i.e., $l_1 = l_2 = 3$. The pure linear activation function was chosen on the first and output sublayers , whereas hyperbolic tangent sigmoid activation function (tansig) on the second sublayer.[*] The identified model has the following structure

$$y = T_1 y^{[-1]} + T_2 u^{[-1]} + C_2 \mathrm{tansig}\left(W_2 \begin{bmatrix} y^{[-2]} & u^{[-2]} \end{bmatrix}^{\mathrm{T}}\right). \tag{17}$$

Since the identified model is of the second order, using (8), (12), and parameters of model (17), the controller becomes

$$\begin{aligned} u &= T_2^{-1}(\eta_1 - T_1 y)\\ \eta_1^{[1]} &= v - C_2 \mathrm{tansig}\left(W_2 \begin{bmatrix} y & u \end{bmatrix}^{\mathrm{T}}\right). \end{aligned} \tag{18}$$

Note that we have intentionally chosen parameters of the reference model to be $b_1 = a_1 = a_2 = 0$ and $b_n = 1$ due to the fact that the transient process possesses relatively slow nature. The reference signal v was chosen as a piecewise constant function presented in Table 3.

Table 3. Set points.

Value [m]	Time interval [s]
0.20	$0 \leqslant t < 180$
0.05	$180 \leqslant t < 270$
0.10	$270 \leqslant t < 360$
0.15	$360 \leqslant t < 450$

Note that v was chosen this way to illustrate the ability of the proposed method to perform in the whole region of set points. It should be mentioned that C/C++ builder, provided by Real-Time Windows Target, does not allow one to use functions that are not available in the core version of MATLAB. This causes certain number of issues that have to be handled during the design stage.

[*] Note that the choice of second order for the identified model is due to the requirements of the control algorithm (7), (8). In other words, the minimal possible order has to be at least 2, see [1]. Moreover, at least one layer of the network has to be nonlinear in order to reflect nonlinearities of the process.

For example, hyperbolic tangent sigmoid transfer function has to be implemented by means of standard blocks in the explicit form using formula $\mathrm{tansig}(x) = 2/(1+\mathrm{e}^{-2x}) - 1$. The quality of the control algorithm is depicted in Fig. 5.

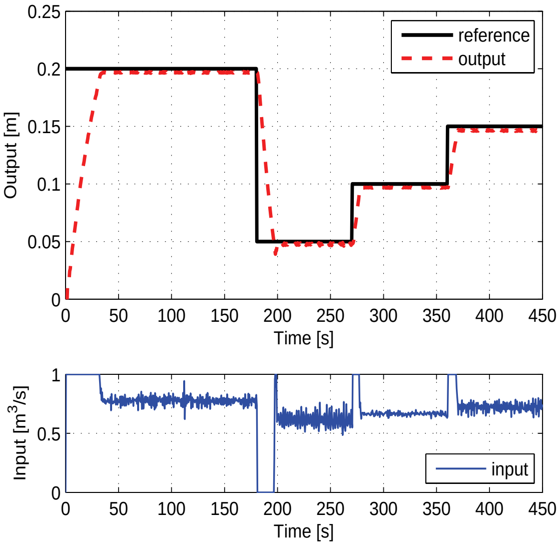

Fig. 5. Experimental results of the water level control in the first tank. The upper plot shows the output (water level). The lower plot reflects the corresponding input (control) signal.

Fig. 5 presents experimental results. The upper plot shows changes (solid line) of water levels and corresponding reaction (dashed line) of the control system. The lower plot represents control signal generated by controller (18). Hence we may conclude that control system is capable of tracking the reference signal v and react correctly to changes in a set point. Finally, it is important to stress that the same controller works accurately on the whole region of set points.

To make the paper self-sufficient, we provide a brief comparison of the technique proposed above with several widely used approaches such as:

- **Anti-Windup PID** control.
- **Relay** control.
- **Analytic** approach based on the so-called exact state feedback linearization technique.

In addition, we discuss the strong and weak points of each technique including the one proposed in this paper. The simulation results are depicted in Fig. 6.

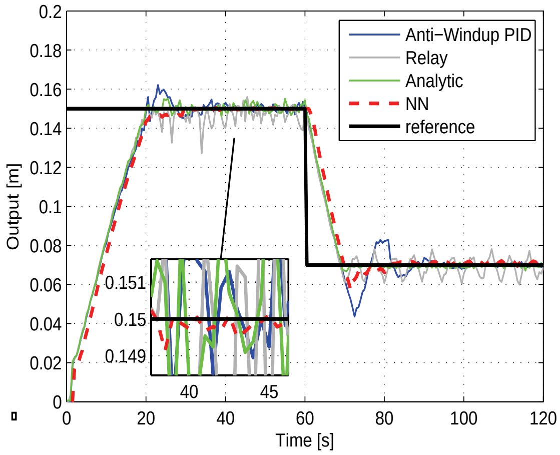

Fig. 6. Comparison results: outputs.

Moreover, Fig. 7 describes the performance of the control signal for each particular approach.

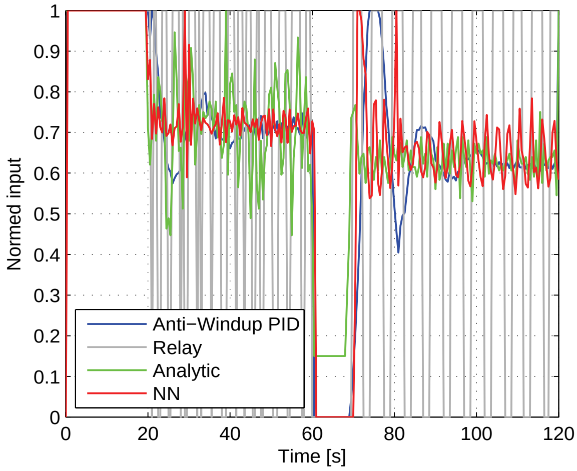

Fig. 7. Comparison results: control signals.

To evaluate the quality of each control algorithm we rely on several statistical tools. The results are presented in Table 4.

Co-published by Atlantis Press and Taylor & Francis

Table 4. Statistical measure of performance in steady-state.

Method	MSE	SSE	$\sum \lvert v-y \rvert$
AW PID	$3.15 \cdot 10^{-5}$	0.2681	26.5788
Relay	$2.47 \cdot 10^{-5}$	0.2103	34.2955
Analytic	$2.6 \cdot 10^{-6}$	0.0221	10.3010
NN	$4.73 \cdot 10^{-6}$	0.0402	11.0672

The evaluation of each method is summarized in Table 5.

Table 5. Brief overview of the control methods.

Criteria	AW PID	Relay	Analytic	NN
complexity	medium	low	high	medium
versatility	medium	high	low	high
robustness	high	medium	medium	high
model	–	–	required	–
quality of u	medium	low	medium	medium

Hence, it follows that:

- **Anti-Windup PID**: *drawback*: strongly depends on the working point and the quality of the corresponding linear model; *advantage*: naturally eliminates steady-state error.
- **Relay**: *drawback*: poor and not effective performance of the control signal (heavily exploits an actuator); *advantage*: simple in realization.
- **Analytic**: *drawback*: requires mathematical model of the process, yielding dependence on the quality of the identified parameters such as α_1 and c_1; *advantage*: in the presence of small measurement noise provides the most efficient control performance.
- **NN**: *drawback*: relies on the heuristic identification techniques; *advantage*: works in the whole range of set points with the same accuracy.

From Table 5 one can see that results of both NN-based and analytical approaches are very close. The difference between them, compared to other techniques, is quite small. However, the advantage of neural networks formalism is that it does not require mathematical model of a process and relies only on the measured i/o data. This means that the same idea can be applied to more complex and even MIMO systems without significant changes.

5.3. Pump/valve-controlled scenario

Here, we cover the case of simultaneous control of water level in several tanks using pump and automatic valves as input signals. Consider settings listed in Table 6.

Table 6. Pump/valve-controlled scenario: configuration.

Tank #	Pump	Manual valve	Automatic valve
1	varies	fully opened	varies
2		fully opened	closed
3		fully opened	closed

The differential equations (16) can be extended to the case of pump/valve-controlled system. Configuration presented in Table 6 yields

$$\begin{aligned} \dot{x}_1 &= \frac{1}{aw}\left(u_1 - u_2 x_1^{\alpha_1} + C_1 x_1^{\alpha_1}\right) \\ \dot{x}_2 &= \frac{h}{cwh + bwx_2}\left(u_2 x_1^{\alpha_1} + C_1 x_1^{\alpha_1} - C_2 x_2^{\alpha_2}\right). \end{aligned} \tag{19}$$

It follows from Table 6 and equations (19) that we are interested in simultaneous water level control in the first and the second tanks using pump and automatic valve (placed between two tanks) as control inputs. Hence, $y_1 = x_1, y_2 = x_2$ and $p = m = 2$. Obviously, the maximum possible inflow provided by the pump remains the same for all performed experiments. However, the automatic valve between tanks has significant dead-zone and performs at the scale between 0.7 and 1, i.e., $u_2(t) \in [0.7, 1]$. The input signal of the pump was scaled into the unity interval as in the previous example, i.e., $u_1(t) \in [0, 1]$.

In the same manner as in the case of pump-controlled scenario, the i/o data was collected with the sampling rate of 0.5sec performing open-loop experiment. Using measurements the neural network, having structure (15), was identified with two sublayers, corresponding to the overall second-order ($n = 2$) of the model and $l_1 = 2, l_2 = 9$ neurons on each hidden sublayer. The pure linear activation function was chosen on the first and output sublayers, whereas hyperbolic tangent sigmoid activation function on the second sublayer. On the basis of the obtained model the feedback linearizing controller from Proposition 2 was constructed. The reference signals are chosen to be

$$v_1 = \begin{cases} 0.12 & \text{if} \quad 0 \leqslant t < 130, \\ 0.1 & \text{if} \quad 130 \leqslant t < 240 \end{cases}$$

and $v_2 = 0.15$. The experimental results are depicted in Fig. 8. It can be seen from Fig. 8 that the outputs of the system are capable of tracking both reference signals.

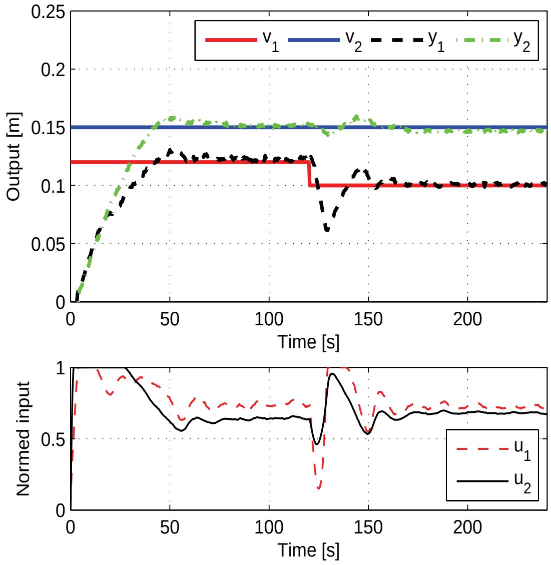

Fig. 8. The upper plot represents experimental results of water levels in the first and second tanks. The lower plot depicts the corresponding control signals.

6. Discussion and Conclusions

In this paper, classical dynamic output feedback linearization algorithm with NN-based SANARX model was applied to control water level in the real tank system. The controller is designed incorporating a reference model, which makes possible to predefine the desired dynamics of the designed control system. The choice of NN-SANARX model class has allowed to merge numeric parts of system modeling and control synthesis that reduced computational complexity and in turn simplified implementation. All numeric calculations were performed in a MATLAB/Simulink without using any specific toolbox functions, making it possible to transfer results of this paper to other environments and programm even in specialized embedded devices.

The paper describes all the steps required to design a controller starting from collection of the i/o data of a process and finishing with implementation and test of the synthesized controller on the prototype of a real plant. We analyzed several possible configurations of the system. The real experimental results are provided. We propose a *systematic* and *constructive* approach for system analysis and controller design of a large class of industrial systems. Note that the derived controllers provide possibility to work in the whole region of set points with the same accuracy and not around some set points. Since the resulting closed-loop system is linear, this approach can be easily combined with existing linear control methods such as pole-placement.

Next, we discuss several limitations appearing within the proposed approach. First, it is always necessary to assume at least the second order of the identified model due to the structure of the controller (7), (8). In fact, this is a natural assumption that has to be made in the majority of the applications. Second, from (12) it follows that, in general, one has to take care of non-singularity of the matrix T_2^{-1}. Finally, it may happen that the model may not necessarily be identifiable with acceptable accuracy by the original SANARX model due to the presence of the high nonlinearity in the data set with respect to the pair $\{y^{[-1]}, u^{[-1]}\}$. Then, some other sublayers can be used, see [25] for more details. Therefore, all mentioned limitations are not restrictive and can be avoided during one of the preimplementation steps.

Acknowledgments

The authors would like to thank their colleagues and anonymous reviewers for the valuable comments and suggestions.

References

[1] E. Petlenkov, “NN-ANARX structure based dynamic output feedback linearization for control of nonlinear MIMO systems,” in *The 15th Mediterranean Conference on Control and Automation*, Athena, Greece, 2007, pp. 1–6.

[2] Ü. Kotta, F. N. Chowdhury, and S. Nõmm, “On realizability of neural networks-based input-output models in the classical state-space form,” *Automatica*, **42** (7), 1211–1216, 2006.

[3] J. Belikov and E. Petlenkov, “Calculation of the control signal in MIMO NN-based ANARX models: analytical approach,” in *The 10th International Conference on Control, Automation, Robotics & Vision*, Hanoi, Vietnam, 2008, pp. 2196–2201.

[4] J. Belikov, E. Petlenkov, and S. Nõmm, “Application of neural networks based ANARX structure to backing up control of a truck-trailer,” in *The 6th IFAC Symposium on Intelligent Autonomous Vehicles*, Toulouse, France, 2007, pp. 1–5.

[5] E. Petlenkov and J. Belikov, “NN-ANARX structure for dynamic output feedback linearization of nonlinear SISO and MIMO systems: neural networks based approach,” in *The 26th Chinese Control Conference*, Zhangjiajie, China, 2007, pp. 138–145.

[6] J. Belikov, K. Vassiljeva, E. Petlenkov, and S. Nõmm, “A novel Taylor series based approach for control computation in NN-ANARX structure based control of nonlinear systems,” in *The 27th Chinese Control Conference*, Kunming, China, 2008, pp. 474–478.

[7] W. C. Dunn, *Fundamentals of Industrial Instrumentation and Process Control*. McGraw-Hill Professional, 2005, p. 322.

[8] C. Kern and M. Manness, “PID controller tuning for mixed continuous/discrete event processes using dynamic simulation,” in *Dynamic Modeling Control Applications for Industry Workshop*, Vancouver, BC, Canada, 1997, pp. 37–43.

[9] E. Cornieles, M. Saad, G. Gauthier, and H. Saliah-Hassane, “Modeling and simulation of a multivariable process control,” in *IEEE International Symposium on Industrial Electronics*, Montreal, Quebec, Canada, 2006, pp. 2700–2705.

[10] R. Precup, L.-T. Dioanca, E. M. Petriu, M. Radac, S. Preitl, and C. Dragos, “Tensor product-based real-time control of the liquid levels in a three tank system,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Montreal, ON, Canada, 2010, pp. 768–773.

[11] Q. Wang, L. Xing, and X. Shi, “Decoupling control of three-tank liquid level systems based on feedforward compensation,” in *Chinese Control and Decision Conference*, Guilin, China, 2009, pp. 5863–5866.

[12] K. Sundaravadivu, V. Jeyakumar, and K. Saravanan, “Design of fractional order PI controller for liquid level control of spherical tank modeled as fractional order system,” in *IEEE International Conference on Control System, Computing and Engineering*, Penang, Malaysia, 2011, pp. 522–525.

[13] A. Tepljakov, E. Petlenkov, J. Belikov, and M. Halás, “Design and implementation of fractional-order PID controllers for a fluid tank system,” in *The 32nd American Control Conference*, Washington, DC, USA, 2013, pp. 1780–1785.

[14] L. Liang, “The application of fuzzy PID controller in coupled-tank liquid-level control system,” in *International Conference on Electronics, Communications and Control*, Ningbo, China, 2008, pp. 2894–2897.

[15] J. T. Naman, M. Z. Abdulmuin, and H. Arof, “Development and application of a gradient descent method in adaptive model reference fuzzy control,” in *TENCON*, Kuala Lumpur, 2000, pp. 358–363.

[16] Y. Xiao, H. Hu, H. Jiang, J. Zhou, and Q. Yang, “A adaptive control based neural network for liquid level of molten steel smelting non-crystalloid flimsy alloy line,” in *The 4th World Congress on Intelligent Control and Automation*, Shanghai, China, 2002.

[17] J. A. Roubos, R. Babuška, P. M. Bruijn, and H. Verbruggen, “Predictive control by local linearization of a Takagi-Sugeno fuzzy model,” in *IEEE World Congress on Computational Intelligence*, Anchorage, AK, USA, 1998, pp. 37–42.

[18] I. J. Leontaritis and S. A. Billings, “Input-output parametric models for non-linear systems Part I: deterministic non-linear systems,” *International Journal of Control*, **41** (2), 303–328, 1985.

[19] R. Pothin, Ü. Kotta, and C. H. Moog, “Output feedback linearization of nonlinear discrete time systems,” in *The IFAC Conference on Control Systems Design*, Bratislava, Slovak Republic, 2000, pp. 181–186.

[20] S. Nõmm and Ü. Kotta, “Comparison of neural networks-based ANARX and NARX models by application of correlation tests,” in *International Joint Conference on Neural Networks*, San Jose, CA, USA, 2011, pp. 2113–2118.

[21] K. Vassiljeva, E. Petlenkov, and J. Belikov, “State-space control of nonlinear systems identified by ANARX and neural network based SANARX models,” in *The World Congress on Computational Intelligence*, Barcelona, Spain, 2010, pp. 3816–3823.

[22] E. Petlenkov, S. Nõmm, and Ü. Kotta, “Neural networks based ANARX structure for identification and model based control,” in *The 9th International Conference on Control Automation Robotics and Vision*, Singapore, 2006, pp. 2284–2288.

[23] ALab. (2014). Alpha Control Laboratory, Tallinn University of Technology, [Online]. Available: `http://a-lab.ee/`.

[24] INTECO Sp. z o. o. (2013). INTECO, [Online]. Available: `http://www.inteco.com.pl`.

[25] J. Belikov and E. Petlenkov, “NN-ANARX model based control of nonlinear discrete-time systems with input delays,” in *The 3rd IEEE Multi-conference on Systems and Control*, St. Petersburg, Russia, 2009, pp. 1039–1044.