DIRECT INVERSE CONTROL OF TWO-TANK SYSTEM USING NEURAL NETWORKS

Conference Paper · October 2021

4 authors:

Pavle Stepanic
Lola Institut

Jelena Vidakovic
Lola Institut

Andrija Dević
Lola Institut

Nedeljko Ducic
University of Kragujevac, Faculty of Technical Sciences Čačak, Serbia

SPMS 2021
38. Savetovanje Proizvodnog mašinstva Srbije
ICPE-S 2021
38th International Conference on Production Engineering -Serbia

Srbija
Society of Production Engineering

Faculty of technical sciences Čačak
University of Kragujevac

Čačak, Serbia, 14 – 15. October 2021.

DIRECT INVERSE CONTROL OF TWO-TANK SYSTEM USING NEURAL NETWORKS

Pavle STEPANIĆ[1,*], Jelena VIDAKOVIĆ[1], Andrija DEVIĆ[1], Nedeljko DUČIĆ[2]
[1]LOLA Institute Ltd, Belgrade, Serbia
[2]Faculty of Technical Sciences Čačak, University of Kragujevac, Serbia
*pavle.stepanic@li.rs

Abstract: *In this paper, the implementation of the controller based on neural networks for controlling two-tank system is presented. A ready-made mathematical model of the Amira DTS200 system, which is a typical example of a slow nonlinear process, is used. Among the most important applications of artificial neural networks is their application in the control of nonlinear processes. The applied controlling structure represents Direct Inverse Control. Experimental results of the obtained process response for a given reference input using implemented inverse controller are given.*

Keywords: *Neural Network, Direct Inverse Control, Two-Tank System.*

1. INTRODUCTION

Linear algebra offers a multitude of various tools for linear process analysis and control. Consequently, the assumption of system linearity is often conveniently used in order to apply a certain control theory that achieves the desired behaviour of linear processes. However, most real-word systems are nonlinear in nature, and the use of linear models often cannot "capture" the complex dynamic behaviors of nonlinear systems. Conventional methods of modeling and control of dynamic processes try to form a physically based mathematical model that is close to the input-output relation of the observed real system. In practice, this usually involves engineering assumptions to obtain simpler system models, which require extensive experience and often result in poorer controller performance. On the other hand, modeling and control of nonlinear systems involve specific problems, with no simple and unified theory available.

Neural networks provide an alternative approach to the identification and control of nonlinear processes in process engineering [1]. Process modeling using neural networks does not require a priori knowledge of physical process phenomena. Instead, neural networks learn by using data to extract existing templates, i.e., patterns that describe the relationship between inputs and outputs regardless of the physical nature of the process. The neural network has been trained when the appropriate inputs are applied, through which it acquires knowledge of the process environment. As a result, the neural network adapts itself according to information that can be called up later. Neural networks are able to perform fast processing of complex nonlinear problems and reduce the engineering effort required to develop an appropriate controller. The possibility to approximate an arbitrary continuous nonlinear function up to the desired accuracy is their most important feature from the point of view of modeling, identification, and control of nonlinear processes.This paper is divided into five sections. In Section 2, the two-tank system and the theoretical setup of the neural network model are described. The design of the controller using a neural network is presented in Section 3. Experimental results are presented in Section 4. Final conclusions are given in Section 5.

2. THEORETICAL BACKGROUNDS

Figure 1 shows a simplified representation of the controlled system, which consists of two tanks of equal height and cross-sections *A* [2]. The water is pumped into the first tank, from where it flows into the second tank via a pipe, with the cross-section of the pipe regulated via a valve a_1. The second tank has a drainpipe, the cross-section of which is regulated by the second valve a_2. Pump pressure u, which regulates the inflow of water into the first tank, is a manipulated variable, while the water level h_2 in the second tank is the controlled variable. Valve openings are herein considered as system disturbances.

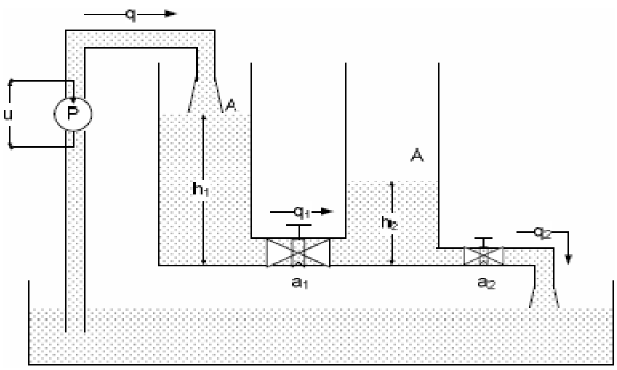

Figure 1. Schematic representation of a two-tank system

A ready-made nonlinear model of the Amira DTS200 system was used as a process model in this paper. Due to the fact that there is a nonlinear model, which very faithfully describes the actual process, all simulations were performed on this nonlinear model, which was realized in Simulink [3], Figure 2.

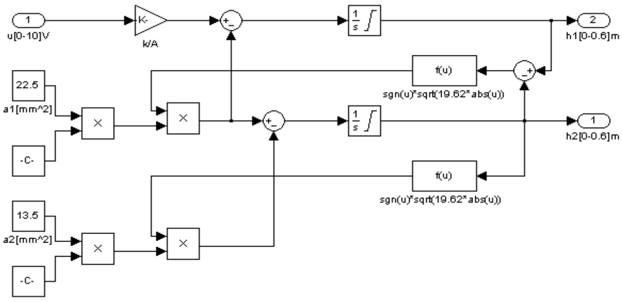

Figure 2. Simulink model of the two-tank system

2.1 Neural Network

For the purpose of this paper, the neural network is considered as a function approximator herein. By adjusting neural network weighting coefficients, an unknown function can be approximated by it so that it produces the same output when the same input data is applied to that particular arbitrary mathematical function (Figure 3). Herein, the unknown function is, in fact, a real process (a two-tank system) that is controlled, and the neural network is used to implement an already identified process model. Additionally, an unknown function can also represent an inversion of the system we want to control, and in that case, the neural network is used to implement the controller.

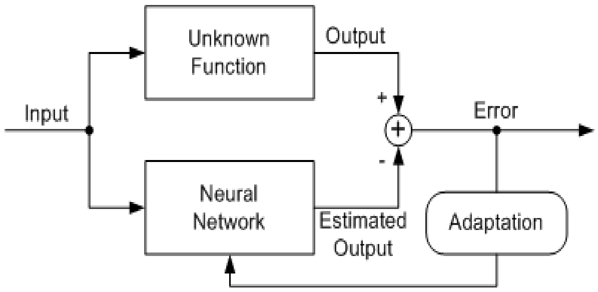

Figure 3. A neural network as a function approximator

For the purpose of nonlinear process modeling, the multilayer perceptron can be interpreted as Nonlinear AutoRegressive with eXogenous input model (NARX).

2.2 Neural Network AutoRegressive with eXogenous input model (NNARX)

The general NARX model is obtained by applying nonlinear regression to previous samples of process output and input signals. The NARX model of a nonlinear process with one input and one output can be presented as [4,5]:

$$y_p(k) = f\begin{pmatrix} y_p(k-1),\ldots,y_p(k-n), \\ u(k-1),\ldots,u(k-m) \end{pmatrix} + w(k) \cdot \tag{1}$$

where $y_p(k)$ and $u(k)$ samples of process outputs and inputs at the *k*-th time instance respectively, *n* is the number of previous output values, *m* is the number of previous input values, $f(\cdot)$ is a nonlinear function that describes the behavior of the process, $w(k)$ is additive Gaussian white noise. The first term in equation (1) depends on the previously measured input and output process signals, while the second term does not, and thus cannot be identified. Given that real process output $y_p(k)$ is available, the associated predictor is a feedforward network (there is no feedback):

$$\hat{y}(k+1) = \hat{f}\begin{pmatrix} y_p(k),\ldots,y_p(k-n+1), \\ u(k),\ldots,u(k-m+1) \end{pmatrix} \tag{2}$$

where $\hat{y}(k+1)$ is the output signal of the process model at the moment $k+1$, and $\hat{f}$ is the approximation function of f. The output signal of the process model $\hat{y}(k+1)$ is calculated in the *k*-th step, based on the currently available measured values of input and output process signals $\left[y_p(k),\ldots,y_p(k-n+1),u(k),\ldots,u(k-m+1)\right]$. Therefore, $\hat{y}(k+1)$ represents the estimated value of the process output signal $y(k+1)$ in $k+1$-th step, calculated one step ahead, in *k*-th step. Therefore, model (2) is called the predictor model of the process, and the error between the output signals of the process and the model is called the prediction error:

$$e(k+1) = y_p(k+1) - \hat{y}(k+1) \cdot \tag{3}$$

If the assumed model is correct, the prediction error $e(k+1)$ is equal to noise $w(k+1)$, and its variance is minimal. Therefore, a neural predictor can be created using a multilayer perceptron (Figure 4):

$$\hat{y}(k+1) = g\begin{pmatrix} y_p(k),\ldots,y_p(k-n+1), \\ u(k),\ldots,u(k-m+1);\theta \end{pmatrix} \tag{4}$$

where $g(\cdot)$ is a function realized by a neural network and θ is a vector containing the weighting coefficients of the network. In abbreviated notation, NNARX can be described as:

$$\hat{y}(k+1) = g(\varphi(k);\theta) \tag{5}$$

where

$$\varphi(k) = \left[y_p(k)\ldots y_p(k-n+1)\ u(k)\ldots u(k-m+1)\right]^T$$

is a regression vector that defines the regression structure of the neural network input. The NNARX model does not contain feedback, i.e., its regressors do not depend on the model parameters. This feature makes the NNARX model numerically stable while making the numerical procedures for estimating model parameters simpler compared to feedback models. Consequently, the NNARX model structure allows the simple application of static neural networks for the approximation of nonlinear functions.

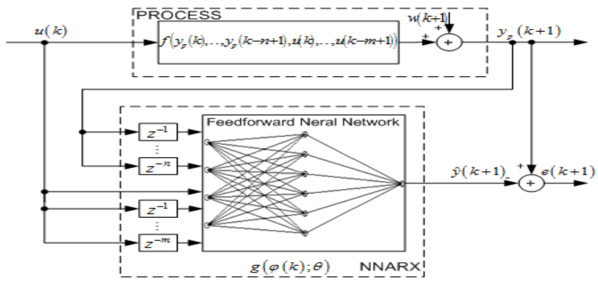

Figure 4. Process modeling using a neural NARX model

3. DESIGN OF THE CONTROLLER USING A NEURAL NETWORK

There is a significant number of controller designs based on the application of neural networks that can be found in the literature [6]. Herein, the structure and application of the Direct Inverse Control strategy, in which the controller represents an inverse process model, is presented. Inverse control is based on the application of an inverse process model, which is applied in series with the process, thus creating a system with an instantaneous unit gain response between the input to the inverse model $r(t)$ and the process output $y(t)$. Therefore, the inverse process model, represented by a neural network, acts as a controller. This is the most basic control strategy using a neural controller. Figure 5 shows an off-line diagram of neural network training, i.e., inverse process modeling.

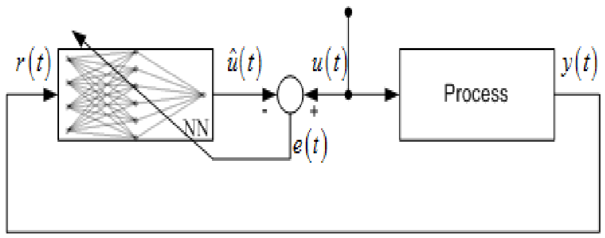

Figure 5. Inverse process modeling by a neural network

The NARX process model (1) can be written in the following form:

$$y(k+1) = f\begin{pmatrix} y(k),\dots,y(k-n+1), \\ u(k),\dots,u(k-m+1) \end{pmatrix} \tag{6}$$

where $y(k)$ and $u(k)$ are process output and input samples. The inverse model of the process described by expression (6), which output is in fact the manipulated signal, is given by the following equation:

$$u(k) = f^{-1}\begin{pmatrix} y(k+1),\dots,y(k-n+1), \\ u(k-1),\dots,u(k-m+1) \end{pmatrix} \tag{7}$$

The inverse process model (7) also belongs to a NARX structure. By applying a neural network that approximates function $f^{-1}(\cdot)$, an inverse neural network controller is obtained. According to expression (7), the output of the inverse process model in *k*-th step depends on the value of the process output in $k+1$-th step, i.e., $y(k+1)$, which is not available in *k*-th step. However, the reference (desired value) of the process output signal $r(k+1)$ is available at the *k*-th step, and this value is used to implement the inverse neural controller. Thus, the inverse neural controller can be described by the following expression:

$$\hat{u}(k) = \hat{f}^{-1}\begin{pmatrix} r(k+1),\dots,y(k-n+1), \\ u(k-1),\dots,u(k-m+1);\theta \end{pmatrix} \tag{8}$$

where $\hat{f}^{-1}(\cdot;\theta)$ is approximation function, i.e., neural network function of $f^{-1}(\cdot)$ with parameter vector θ. Figure 6 shows the strategy of direct inverse control using NNARX controllers.

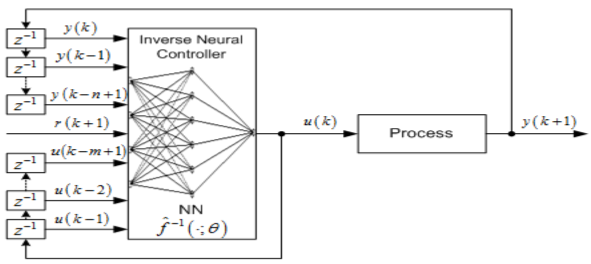

Figure 6. Inverse control system using NNARX controller

4. EXPERIMENTAL RESULTS

Figure 7 shows the implementation of a direct inverse controller used to control the two-tank system previously described. The NNARX inverse model presented in the previous section was used as a controller. The controller generates a control signal in the range [0-10] V.

Figure 8a) shows the process output for a given reference signal with implemented inverse controller, while the generated control output is given in Figure 8b).

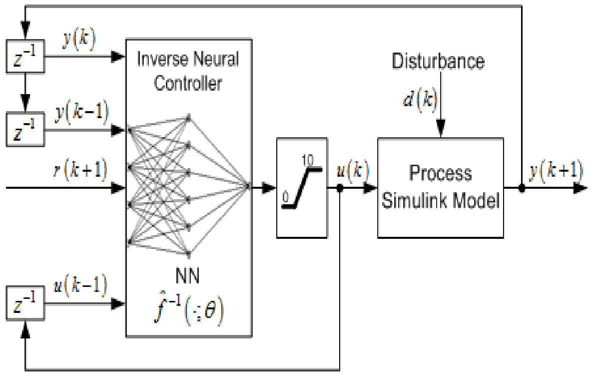

Figure 7. Direct inverse control of the two-tank system

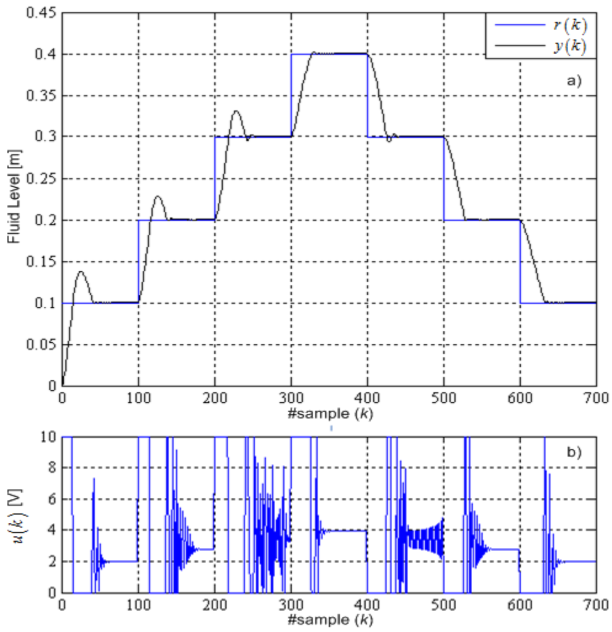

Figure 8. Direct inverse control of the two-tank system a) r(k) - reference input and y(k) - process output b) controller output

From the process response from Figure 8, it can be concluded that for the instances of the step changes in the set-point value, in order to quickly reduce the error between the set-point value and the current value of the output signal (the level of the second tank h_2), a large positive or negative value of manipulated signal (depending on whether the change of the set-point signal is positive or negative) is generated. Due to physical limitations, the values of the controller output are limited to the range [0-10] V, which is most commonly used in practice. Also, in order to reduce the abrupt jump-like large changes and to mitigate large oscillations of the controller output, a reference prefilter can be introduced in order to smooth out the jumps (large step changes) of the reference value. In this way, the controller is adjusted to the system dynamics, and overshoot is reduced.

5. CONCLUSION

In this paper, the application of static neural networks for the purpose of the identification and control of nonlinear dynamic processes is presented. From the point of view of process identification, their most important feature is the ability to approximate arbitrary continuous functions. The NARX structure of the neural network, which relies the process outputs as feedback, is used for the development of the inverse and the direct process models. Consequently, obtained neural network-based models are numerically stable, and numerical training procedures to obtain appropriate network parameters are simpler compared to recurrent neural networks.

Direct inverse control applied to a real nonlinear mathematical model is also presented. Presented controller structure based on direct inverse control gave satisfactory results for the set reference value tracking, even in the presence of disturbances. The presented control system is not able to completely suppress the disturbance, and consequently, the steady-state error appears, which is, however, negligible for the case of the presented two-tank system. Controller output oscillations can be reduced by introducing a reference prefilter into the control system.

ACKNOWLEDGEMENT

This research has been supported by the research grants of the Serbian Ministry of Education, Science and Technological Development.

REFERENCES

[1] H.B. Demuth, M.H. Beale, O. De Jess, M.T. Hagan: Neural Network Design. Martin Hagan, 2014.

[2] Amira, DTS200: Laboratory Setup Three-Tank-System, Amira GmbH, Duisburg, 1998.

[3] MATLAB 2018a, The MathWorks, Inc., Natick, Massachusetts, United States, 2018.

[4] I. J. Leontaritis, S.A. Billings: Input-output parametric models for non-linear systems. Part I: deterministic non-linear systems. Part II: stochastic non-linear systems. International Journal of Control 41, pp. 303-344, 1985.

[5] I. Rivals, L. Personnaz: Black-Box Modeling with State-Space Neural Networks. In: Neural Adaptive Control Technology, R. Zbikovski and K.J. Hunt eds., World Scientific, pp. 237-264, 1996.

[6] M. Hagan, H. Demuth: Neural Networks for Control. Invited Tutorial, American Control Conference. June, 1999, San Diego, pp. 1642-1656, 1999.