Journal of the Brazilian Society of Mechanical Sciences and Engineering (2021) 43:248
https://doi.org/10.1007/s40430-021-02957-y

TECHNICAL PAPER

A neural network-based inversion method of a feedback linearization controller applied to a hydraulic actuator

Fábio Augusto Pires Borges[1] · Eduardo André Perondi[2] · Mauro André Barbosa Cunha[3] · Mario Roland Sobczyk[2]

Received: 10 February 2020 / Accepted: 18 March 2021 / Published online: 10 April 2021

Abstract
In this work, we use a neural network as a substitute for the traditional analytic functions employed as an inversion set in feedback linearization control algorithms applied to hydraulic actuators. Although very effective and with strong stability guarantees, feedback linearization control depends on parameters that are difficult to determine, requiring large amounts of experimental effort to be identified accurately. On the other hands, neural networks require little effort regarding parameter identification, but pose significant hindrances to the development of solid stability analyses and/or to the processing capabilities of the control hardware. Here, we combine these techniques to control the positioning of a hydraulic actuator, without requiring extensive identification procedures nor losing stability guarantees for the closed-loop system, at reasonable computing demands. The effectiveness of the proposed method is verified both theoretically and by means of experimental results.

Keywords Hydraulic actuator control · Neural network-based identification · Feedforward multilayer perceptron · Feedback linearization

1 Introduction

Due to their high force/size ratios, hydraulic actuators are widely used in tasks combining high forces with limited dimensions. On the other hands, their dynamics depend on strongly nonlinear phenomena such as valve saturation, behavior of the flow rates through the valve orifices and friction forces in the piston [1]. Moreover, their open-loop dynamics present low damping, and their mathematical modeling suffers from significant uncertainties in many key parameter values, such as leakages and dead zones in the control valves. These are challenging characteristics for their corresponding controllers to cope with, making it difficult to use such actuators in high-precision applications. Several control strategies have been proposed with the goal of enhancing the precision of hydraulic actuators, e.g., backstepping [2–6], feedback linearization [7–9] and sliding mode control [10–12]. Many other algorithms are based on combinations of these and other techniques, as illustrated in [13–15], for instance. An important feature of these model-based control approaches lies in the possibility of using Lyapunov methods for analyzing the controlled plants, providing stability guarantees even when nonlinear effects are explicitly taken into account. All of the aforementioned papers include some strategy to overcome the parametric uncertainties and external disturbances present in hydraulic actuators. Many works include online learning methods, where parameters and disturbances are estimated in real time, such as adaptive control [6, 10, 13], extended state observer (ESO) [2, 6], extended disturbance observer (EDO) [4] and extended differentiator [3]. Even though they

Technical Editor: Victor Juliano De Negri.

✉ Fábio Augusto Pires Borges
fborges@furg.br

Eduardo André Perondi
eduardo.perondi@ufrgs.br

Mauro André Barbosa Cunha
maurocunha@ifsul.edu.br

Mario Roland Sobczyk
mario.sobczyk@ufrgs.br

[1] N2E Research Group, Engineering School, Federal University of Rio Grande, Av Italia, km 8, Rio Grande, RS 96203-900, Brazil

[2] Federal University of Rio Grande Do Sul, Rua Sarmento Leite 425, Porto Alegre, RS 90050-170, Brazil

[3] Automation and Control Research Group, Sul-Rio-Grandense Federal Institute for Education, Science, and Technology, Praça Vinte de Setembro, 455, Pelotas CampusPelotas, RS 96015-360, Brazil

yield reduced position errors compared to fixed-structure controllers, they also have important drawbacks for practical applications, such as the need for tuning large numbers of control parameters. Usually, such tuning process must take into account a compromise between the robustness of the controller, the computing capabilities of the available control hardware and the convergence speed of the estimation procedure, which is a rather difficult task. Furthermore, other factors like sensor noise tend also to affect the convergence performance. Thus, at least some of the aforementioned online estimation approaches may require highly sophisticated and expensive sensors for attaining accurate results.

Intelligent strategies, such as fuzzy logic or neural networks, have been widely used in many different applications, mainly because of their ability of "learning by themselves" how to adapt to their working environment. In the control of hydraulic actuators, such methods form the core of many algorithms [16–22], and many of them rely on fully online learning procedures to compensate for the unknown disturbances and unmodeled dynamics. Some works are based on radial basis function (RBF) neural networks [20, 21], others use simplified feedforward multi-layer perceptron neural networks linearizated by means of Taylor's series expansion [19, 23], an approach based on [24] and similar to that of several works in other areas, such as [25, 26]. These approaches are suitable for deriving Lyapunov-based stability proofs, but they are also prone to the same problems discussed in the last paragraph: Since hydraulic actuators have several highly nonlinear effects and uncertain parameters, these algorithms are difficult to tune unless the available control hardware is very powerful. Furthermore, as pointed out in [27], a fully online training procedure for these systems may result in very slow convergence due to the local minimum problem, which in turn leads to unacceptably large trajectory-tracking errors for exceedingly long time periods.

An effective approach for controlling hydraulic systems is based on *feedback linearization* methods, which compensate for nonlinear phenomena by estimating their effects and applying a control signal that opposes them. This technique depends on two critical aspects of the mathematical model of the system: (i) *inversibility*, i.e., finding a set of invertible state functions to compose the model; (ii) *accuracy*, which determines how effectively such nonlinearities can be canceled. For hydraulic actuators, such model usually involves the piston position and velocity, the pressures in the piston's chambers (or their equivalent hydraulic force) and the input voltage applied to the control valve. Although some examples of empirical models can be found [28], these relations are more commonly described as analytic functions depending on many parameters that are uncertain and/or difficult to measure, thus requiring several time-consuming experimental procedures to reliably identify the corresponding values [7, 9, 13]. Moreover, even when different systems are assembled with components of the same model and manufacturer, their dynamic responses may differ significantly if the identification procedure is not repeated for each specific set. Thus, even though the results from this approach are usually satisfactory in terms of accuracy, significant time and effort must be consumed in their implementation in hydraulic actuators, reducing their attractiveness for practical or commercial applications. A partial solution to this problem was given in [17, 22, 29], which employ multi-layer perceptron feedforward neural networks instead of a classical, invertible state equation set for controlling the hydraulic actuator. These networks undergo two training procedures: The first one is *offline*, bearing the brunt of the learning process and bringing the network near to its best tuning condition; the second is performed *online*, for small weight adjustments with minor computing efforts and fast convergence. This approach helps to avoid the extensive manual labor involved in traditional parameter identification procedures involved in model-based feedback linearization controllers, which is desirable. However, no formal stability proofs are given for the closed-loop system in these cases, which is a major drawback in terms of safe operation guarantees.

In this work, we propose a hybrid control scheme applied to a hydraulic positioning system. It employs two control loops, based on the main phenomena involved in the system: The outer loop is concerned with its mechanical variables, whereas the inner loop is a feedback linearization algorithm, which aims at compensating for the nonlinearities due to the hydraulic dynamics. In the inner loop, we substitute just one part of the traditional analytic model with neural networks, namely the computing of the actual control signal to be applied to the control valve, which is performed by an offline, extensively trained feedforward multi-layer perceptron neural network. Since most of the critical uncertain parameters of the system are necessary in this specific step of the control loop, we still retain the basic advantages of feedback linearization control, such as direct cancelation of undesirable nonlinear effects and rigorous stability guarantees, while avoiding large amounts of experimental effort to find suitable parameters for the inverse set, a significant advantage when compared with the traditional offline methods. Furthermore, the offline training procedure breaks the application of the proposed controller into two stages, training and trajectory tracking, which reduces significantly the computing efforts involved in each one, thus making the proposed method amenable to control hardwares more modest than these required for fully online-trained controllers. The effectiveness of the proposed strategy is demonstrated both analytically, with a rigorous stability proof using Lyapunov's second method, and in practice, with comprehensive experimental results. This approach was introduced in [16], being expanded here by including its main theoretical aspects: the

Lyapunov stability proofs and an enhanced discussion on the development of the neural network.

The remainder of this paper is structured as follows. In Sect. 2, the hydraulic actuator model and the experimental setup are discussed. In Sect. 3, we present the overall control strategy, whereas Sect. 4 is dedicated to the development of the proposed neural network-based inversion set. In Sect. 5, the stability proof is detailed, while in Sect. 6, the proposed method is evaluated by means of experimental results. Finally, the main conclusions are outlined in Sect. 7.

2 System description

Figure 1 illustrates a hydraulic actuator, which comprises a differential cylinder attached to a load and controlled by a symmetrical four-way servo valve, as depicted in Fig. 1b: By regulating the volumetric flow rates Q_1 and Q_2 into and out of chambers 1 and 2, the servo valve causes a hydraulic force F_H to be applied to the piston, thus controlling its motion. The most important variables and parameters are also highlighted: p_1 and p_2 are the chamber pressures, p_s is the supply pressure, p_0 is the reference pressure, M is the total mass, F_A is the friction force, A_1 is the piston cross-section area,

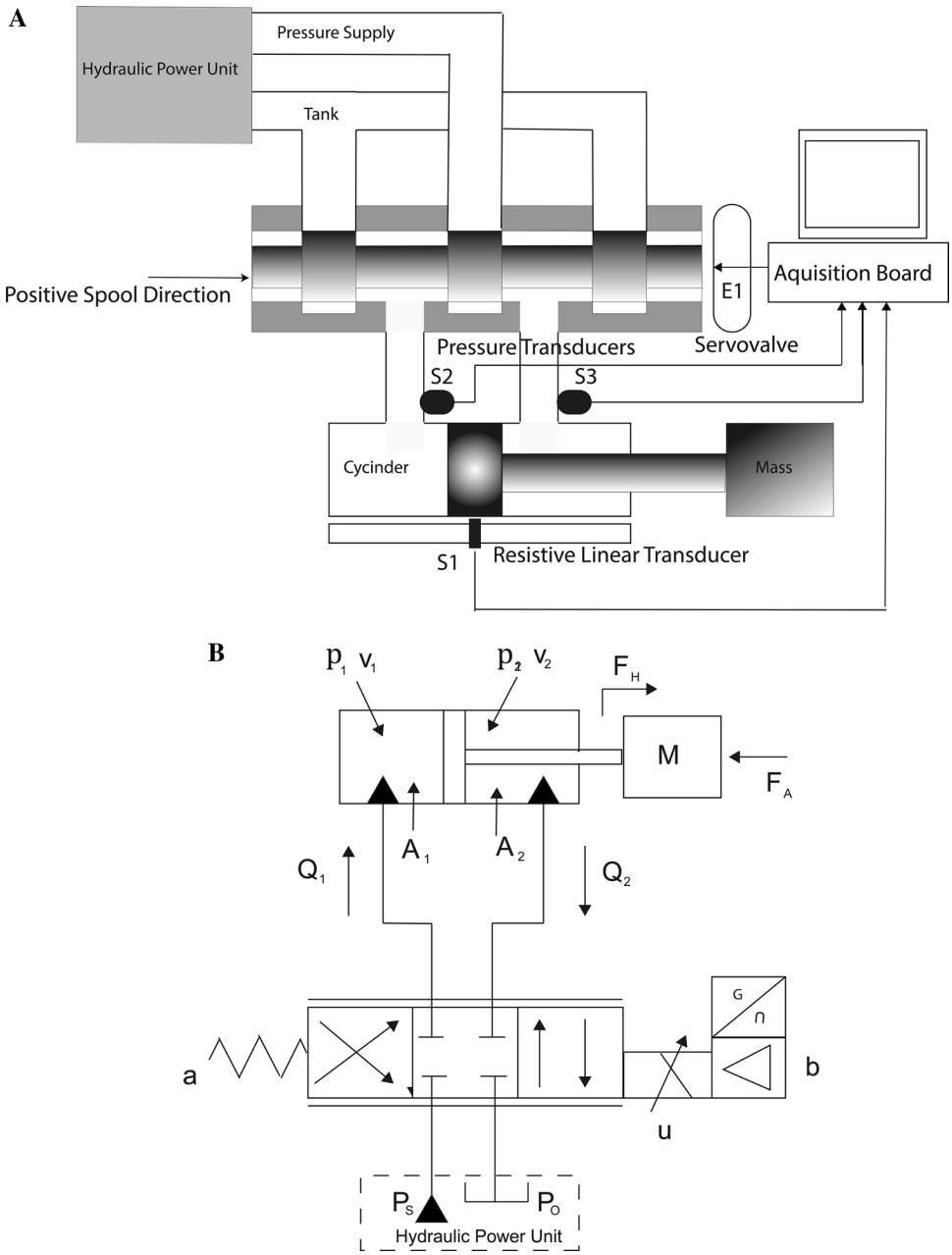

Fig. 1 Graphical description of the hydraulic actuator: **a** basic elements, **b** schematic circuit

A_2 is the difference between the areas of the piston and the rod, v_1 and v_2 are the total volumes in lines 1 and 2, and u is the electrical control signal applied to the valve. The corresponding experimental rig is shown in Fig. 2. The actuator is a Bosch Rexroth CDT3ME5 double action hydraulic cylinder with 200 mm utile stroke, controlled by means of a 4-way Bosch Rexroth 4WRPEH6 directional servo valve. All hydraulic connections have diameter hoses of 1/4-inch (6.285 mm) diameter. Piston position is measured by means of a Novotechnic TLH300 linear resistive position transducer (S1), the pressure transducers are Rexroth HM-18–210 bar (S2) for chamber 1, and Huba Tp-491–400 bar (S3) for chamber 2. The piston velocity is estimated by numeric differentiation of the position signal. The oil temperature is measured by a Novus 8,803,829,780 temperature transmitter (S4) attached to a pt100 sensor installed inside the tank of the hydraulic power unit. All measured signals and the control algorithm are processed in a PC-hosted DSpace 1104 real-time control board, programmed through a MATLAB-Simulink software package. Processing sample time for the control loop is 0.5 ms, and the corresponding control signal is applied to valve input E1.

In the remainder of this section, we describe the developing of a dynamic model for this system and the difficulties involved in determining its parameters.

2.1 Dynamic modeling of a hydraulic actuator

The system traditional dynamic model is obtained by using Newton's second law and flow-continuity considerations, as described in detail in [1]. Usually, these models assume that the proportional valve is ideal, i.e., its opening area is directly proportional to the electric control signal u. Under such assumption, the equations representing the system dynamics are:

$$M\ddot{y} + F_A = F_H - F_D \tag{1}$$

$$\dot{p}_1 = \frac{\beta}{v_1 + A_1 y}\left(Q_1 - A_1\dot{y}\right) \tag{2}$$

$$\dot{p}_2 = -\frac{\beta}{v_2 - A_2 y}\left(Q_2 - A_2\dot{y}\right) \tag{3}$$

where y,$\dot{y}$ and $\ddot{y}$ are the position, velocity and acceleration of the piston-load assembly, respectively, $F_H = p_1A_1 - p_2A_2$ is the hydraulic force applied to the piston, and β is the bulk modulus. F_D represents a generic disturbance force such as dry friction or, according to how the piston is mounted, gravity. The viscosity coefficient B determines the friction force F_A, expressed by:

$$F_A = B\dot{y} \tag{4}$$

Fig. 2 Actual experimental setup: **a** actuator and sensors, **b** hydraulic power unit, **c** acquisition board console

The volumetric flow rates through the valve orifices are functions of the pressures in the chambers and the input signal applied to the valve, expressed by:

$$Q_1 = K_{v1} u g_1,\ g_1 = \begin{cases} \sqrt{p_s - (p_1 + l_1)}, & u \geq 0 \\ \sqrt{p_1 - l_3}, & u < 0 \end{cases}$$
$$Q_2 = K_{v2} u g_2,\ g_2 = \begin{cases} \sqrt{p_2 - l_4}, & u \geq 0 \\ \sqrt{p_s - (p_2 + l_2)}, & u < 0 \end{cases} \quad (5)$$

where K_{v1} and K_{v2} are the flow rate gains that characterize each orifice of the valve, whereas l_1.. l_4 are the pressure losses caused by the hydraulic line couplers, which are significant and must be taken into account when high-precision tasks are considered.

Equations (1)-(5) form an open-loop model of the system. The values of its parameters are given in Table 1, and the difficulties involved in their determination are discussed in Sect. 3.2.

3 Control strategy

Feedback linearization is one of most popular nonlinear control strategies [7, 30, 31]. Its basic idea is to use a nonlinear function of the states of the system in the feedback loop to cancel nonlinear effects, converting the plant into a linear one. From this point, an external loop based on classical linear strategies, such as pole placement by state feedback, for example, could be used to control the system. Without loss of generality, this approach is readily understood when the system model can be written in the so-called control/input affine form, i.e.,

Table 1 System parameters

Parameter	Value	Estimation method
V_{10}	1.2446×10^{-4} m^3	Direct measurement
V_{20}	9.9060×10^{-5} m^3	Direct measurement
y_0	0.1 m (center)	Direct measurement
K_{v1}; K_{v2},	$\sqrt{2}\cdot 15.11\times10^{-9} m^3 \big/ (s\times\sqrt{Pa})$	Manufacturer catalog
l_1	$5.68\times10^{10} Q_1$ Pa	Interactive simulation
l_2	$4.35\times10^{10} Q_2$ Pa	Interactive simulation
l_3	$3.59\times10^{10} Q_1$ Pa	Interactive simulation
l_4	$3.59\times10^{10} Q_2$ Pa	Interactive simulation
M	14.54 kg	Direct measurement
A_1	4.91×10^{-4} m^2	Direct measurement
A_2	2.37×10^{-4}m^2	Direct measurement
β	1.0×10^9 N/m^2	Typical value
Os	50×10^5 Pa	Input choice
B	3600 Ns/m	Experimental measurement

$$x^{(n)} = f(\boldsymbol{x}) + b(\boldsymbol{x})u \quad (6)$$

where u is a scalar control input, x is the scalar output of interest, $\boldsymbol{x} = \left[x, \dot{x}, .., x^{(n-1)}\right]$ is the state vector, and $f(\boldsymbol{x})$ and $b(\boldsymbol{x}) \neq 0$ are nonlinear state functions. If $f(\boldsymbol{x})$ and $b(\boldsymbol{x})$ are known, defining $v(\boldsymbol{x})$ as a linear term matching the desired dynamics for the closed-loop system, it is straightforward that the input

$$u = b^{-1}(\boldsymbol{x})\left[v(\boldsymbol{x}) - f(\boldsymbol{x})\right] \quad (7)$$

leads the controlled nonlinear system to perform as a linear one that presents the desired dynamic behavior, i.e.,

$$x^{(n)} = v(x) \quad (8)$$

Considering the discrepancies between the model and the actual system, the closed-loop dynamics can be expressed as

$$x^{(n)} = v(x) + \varepsilon \quad (9)$$

where ε is the residue from imperfect cancelations due to the intrinsic differences between the real system and its related model. In the next section, we discuss how this control approach is integrated into the proposed algorithm.

3.1 The proposed controller

Hydraulic actuators present two main dynamic phenomena: the pressure variations in the chambers due to opening the control valve, and the motion of the piston due to the forces applied to it. The controller employed here is based on interpreting the actuator as two interconnected subsystems: one hydraulic and one mechanical. In this framework, the force F_H generated in the hydraulic subsystem is regarded as an intermediate input applied to the mechanical one, resulting in a *cascade* structure. This interpretation allows one to develop the system controller in two steps:

(i) In the mechanical subsystem (outer loop), compute a first control law, representing an "ideal" hydraulic force that leads the piston to track its desired trajectory;

(ii) In the hydraulic subsystem (inner loop), develop a second control law, implemented as a valve opening that causes the generated hydraulic force to track its desired value as closely as possible.

The main advantage of this interpretation lies in the independent choice of control laws for each subsystem, so that their most important problems can be tackled separately. This strategy is used in several control algorithms applied to fluid-driven actuators, such as [13] and [32].

The control law applied to the mechanical subsystem is a variation of the classic computed torque control used, by instance, in [33] for controlling an electric robot arm. Its

definition relies on a reference acceleration $\ddot{y}_r$ and an auxiliary error measure z, defined as follows:

$$\dot{y}_r = \dot{y}_d - \lambda\tilde{y}, z = \dot{\tilde{y}} + \lambda\tilde{y} \tag{10}$$

where y_d is the desired piston position, $\tilde{y} = y - y_d$ is the position tracking error, and λ is a positive constant. All terms marked with one or two dots are the first or second time derivatives of the corresponding variables, respectively.

The desired force in the mechanical subsystem is calculated as

$$F_{H_d} = M\ddot{y}_r - K_d z + B\dot{y} \tag{11}$$

In the hydraulic subsystem, we use feedback linearization control. When all the terms are written in the form of Eq. (6), one proceeds as follows. First, define $\boldsymbol{x} = [y, \dot{y}, F_H]^T$ and the auxiliary terms f_1 and f_2 as:

$$f_1 = \frac{\beta}{v_1 + A_1 y}, f_2 = \frac{\beta}{v_2 - A_2 y} \tag{12}$$

and replacing Q_1, Q_2, f_1, and f_2 in (2) and (3) by their corresponding terms given in (5) and (12):

$$\begin{aligned} b(\boldsymbol{x}) &= A_1 f_1 K_{v1} g_1 + A_2 f_2 K_{v2} g_2 \\ f(\boldsymbol{x}) &= -(A_1^2 f_1 + A_2^2 f_2)\dot{y} \end{aligned} \tag{13}$$

From (1) and (13), the dynamics for this subsystem are given by:

$$\dot{F}_H = -(A_1^2 f_1 + A_2^2 f_2)\dot{y} + \left(A_1 f_1 K_{v1} g_1 + A_2 f_2 K_{v2} g_2\right)u \tag{14}$$

since the objective is to cancel its nonlinear effects and generate the desired force F_{Hd}, the proposed control law is

$$\begin{aligned} u^* &= (A_1 f_1 K_{v1} g_1 + A_2 f_2 K_{v2} g_2)^{-1}\left[\dot{F}_{Hd} - K_p \tilde{F}_H + (A_1^2 f_1 + A_2^2 f_2)\dot{y}\right] \\ &= b^{-1}\left[v + f\right], \end{aligned} \tag{15}$$

where $b = A_1 f_1 K_{v1} g_1 + A_2 f_2 K_{v2} g_2$, $v = \dot{F}_{Hd} - K_p \tilde{F}_H$, $f = (A_1{}^2 f_1 + A_2{}^2 f_2)\dot{y}$, $\dot{F}_{Hd}$ is the time derivative of the desired hydraulic force, $\tilde{F}_H = F_H - F_{Hd}$, and K_p is a positive feedback gain. As it will be proven in Sect. 5, if all parameters of the system are known, substitution of this control input into Eq. (14) causes the suppression of all undesired nonlinear effects of the system, leading the closed-loop tracking errors to converge asymptotically to zero.

If parametric uncertainties are present, their effect can be modeled as a residual term $\varepsilon(\boldsymbol{x})$, and the actual control signal applied to the valve is calculated by means of:

$$u = u^*(x) + \varepsilon(x) \tag{16}$$

where $u^*(\boldsymbol{x})$ is the ideal control signal that leads to the perfect cancelation of all undesirable nonlinear effects in the system.

3.2 Implementation difficulties

The residue $\varepsilon(x)$ expressed in Eqs. (9) and (16) affects the performance of the controlled system in tracking its desired dynamic behavior, making it possible, for instance, to compromise its transient and steady-state behavior and even its stability properties. Therefore, obtaining an accurate and comprehensive model is a critical step in developing an effective feedback linearization controller. In the case of a hydraulic actuator, this step involves physical and geometric parameters whose values are difficult to determine. The terms A_1, A_2, v_1 and v_2 can usually be determined with good precision from catalog data or regular measurement procedures [32]. In order to identify the other parameters, based both on previous work [1, 34, 35] and our own experience, we can state that such task is seriously hindered by the following issues:

a. In standard models, the valve gains K_{v1} and K_{v2} are widely regarded as constants. In fact, however, they depend on the pressure losses in the valve orifices, which reflect the energy dissipated due to turbulent flow [1]. As this effect is highly nonlinear and depends on fluid particles' velocity, these coefficients change significantly with flow rate variations. Moreover, due to geometric and electromechanical factors involved in its construction, the relation between the voltage input applied to the valve and its corresponding orifice areas can be rather specific for each given valve. Finally, the valve is subject to intrinsic discrepancies between the dimensions of its orifices and those of the spool landings that control the opening area. Commonly, these landings are slightly wider than the orifices, causing the appearance of a dead zone in the near-zero operation region. Combined, all these effects imply that the values of the valve gains (as functions of the control signal u) can be highly variable, especially for smaller inputs, as demonstrated experimentally by [34] and illustrated in Fig. 3. Thus, to achieve a reliable representation of this behavior, such functions must be obtained by means of direct measurements, which must be repeated for each new valve to be employed;
b. The pressure drops l_1 *to* l_4 depend on the volumetric flow rates in the system lines. Their calculation is based on experiments that require pressure sensors to be installed in difficult-access spots. Moreover, the sensors themselves cause additional pressure losses which interfere with the measured values. This problem can be avoided by estimating these losses by means of interactive simulation procedures, but only at significant costs in terms of time and effort.
c. Most models disregard internal leakages in the valve, which are significant in common-quality equipment.

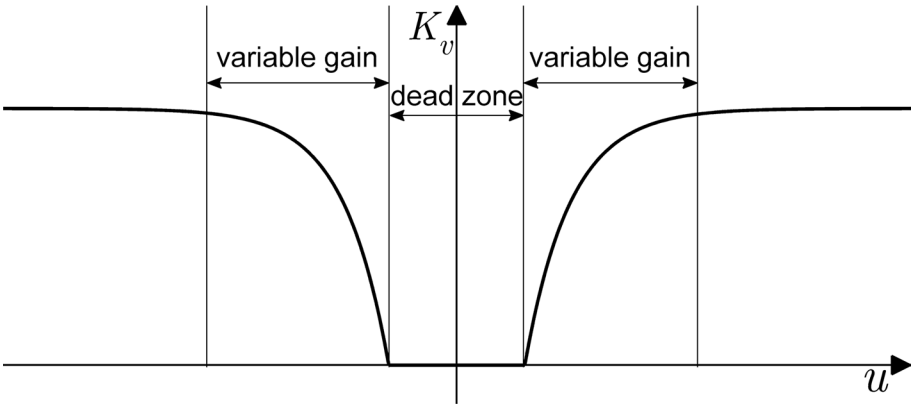

Fig. 3 Flow rate gain characteristics for a typical hydraulic valve. u stands for the control signal applied to the valve, whereas K_v is the flow rate gain

This also affects the values of the valve gains, and its compensation usually involves the use of more complex models and additional experimental efforts.

d. Variations in the hydraulic fluid temperature affect strongly all the parameters listed in Table I [35].

Due to the above reasons, developing an accurate invertible model of a hydraulic actuator by fully analytical means is a difficult task, which jeopardizes the effectiveness of feedback linearization-based control algorithms, which justifies the search for alternative modeling approaches.

3.3 Neural network-based inversion method

Seeking to ensure that $\varepsilon(\boldsymbol{x})$ is small even if all necessary parameters are difficult to determine accurately, we calculate u by replacing the term $b^{-1}(\boldsymbol{x})$ in Eq. (15) by the output $\Omega(\boldsymbol{x})$ of a neural network. In the specific case of the hydraulic actuator, considering which variables can be more easily measured, such neural network is $\Omega(y, \dot{y}, p_1, p_2, T)$, yielding:

$$u = \Omega(y, \dot{y}, p_1, p_2, T)\left[\nu + f\right] + \varepsilon, \tag{17}$$

where T is the hydraulic fluid temperature measured inside the tank of the hydraulic power unit.

The employed neural network is a feedforward multilayer perceptron, which will be discussed with more detail in Sect. 4. The control strategy is described schematically in Fig. 4.

4 Feedforward multi-layer perceptron neural network

The feedforward multi-layer perceptron (MLP) neural network, presented in Fig. 5, is usually applied to static mappings and defined as

$$o = \Gamma\left[W_n\Gamma\left[W_{n-1}...\Gamma\left[W_1u + b_1\right] + ... + b_{n-1}\right] + b_n\right] \tag{18}$$

where W_n is the weighting matrix of the n-th layer, b_n is the bias vector associated with each layer node, and $\Gamma(x) = [\gamma_1(x), \gamma_2(x), \ldots, \gamma_n(x)]$ is a nonlinear operator where each $\gamma_n(.)$ is a monotonic and continuously differentiable activation function. In the present work, the sigmoidal logistic function described in Eq. (19) was used. In Appendix, the equations applied in normalization of the neural network inputs for values between 0 and 1 are presented.

$$\gamma(x) = \frac{1}{1 + e^{-x}}. \tag{19}$$

The main advantage in employing the feedforward MLP neural network in this specific context lies in the existence of guaranteed bounds to the estimation errors of its output [29, 36, 37]. In traditional feedback linearization control

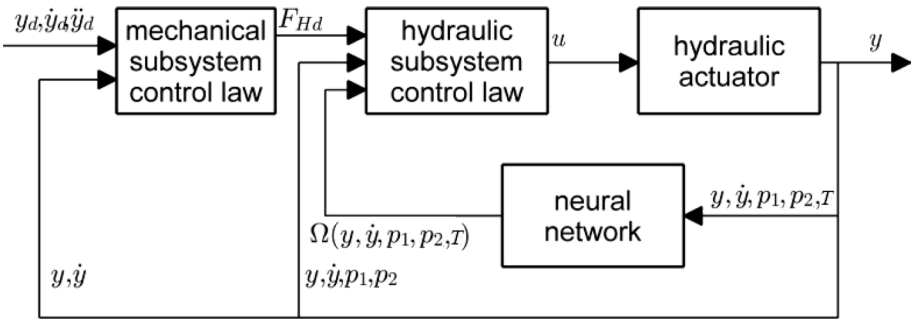

Fig. 4 Application scheme for the proposed controller

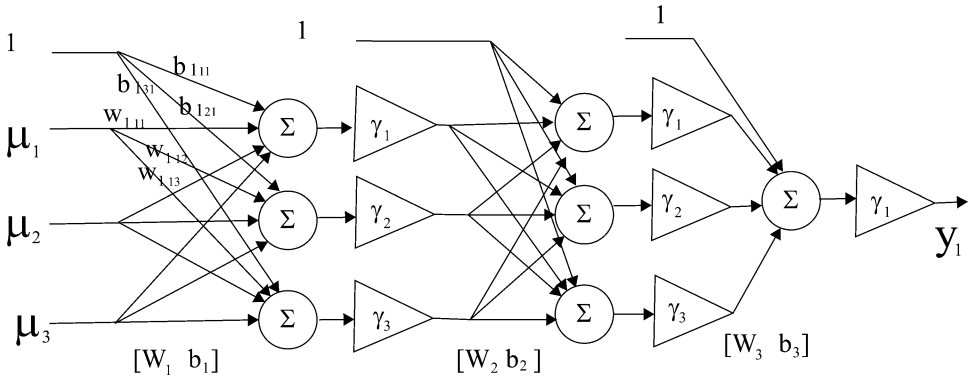

Fig. 5 Feedforward multi-layer perceptron neural network

approaches, the key argument in the corresponding stability proofs is the existence of a known limit to the amplitude of the cancelation residue ϵ [32]. Therefore, as shown in Sect. 5, such proofs can be readily extended to the case when the output of the network replaces the formal inversion of the theoretical model, which is a critical feature of the proposed control approach.

4.1 Neural network design

To carry out a satisfactory inversion using the neural network applied in Eq. (17), we used a hybrid approach to generate the training sets and Eq. (20) is used to compute the desired output signal $\Omega(\boldsymbol{x})$ of the neural network:

$$\Omega(y, \dot{y}, p_1, p_2, T) = \frac{u}{\dot{F}_H + (A_1^2 f_1 + A_2^2 f_2)\dot{y}} \tag{20}$$

highlighting that the function f (Eq. (17)) is an analytical function with known parameters.

The measures $\dot{F}_H$ and $\dot{y}$ are the force and position derivatives obtained by the application of a set of signals u in the plant and are numerically computed. When these values are close to zero, the output of the function grows to infinity. To avoid this effect, the outputs correspondent to very low velocities applied in Eq. (20) are not considered in the training process. According to [32], the feedforward neural networks are able to generalize outputs from inputs that were not considered in the training process.

The neural network training is performed offline and Fig. 6 illustrates the process that uses the Quickprop algorithm [38] to minimize e_d. The inputs of the neural network are the oil temperature, the pressions in the piston chambers, piston position and piston velocity.

A very important issue in this case is a suitable choice of the set of signals u, which is based on the trajectories that the system will perform during its regular operation. In the current case, the design of controllers based on the control strategy defined in the previous section requires a smooth tracking position trajectory [32]. Therefore, for testing purposes, the typical position trajectory performed by the hydraulic piston in the present paper is the 7th-order polynomial reference position trajectory depicted in Fig. 7.

Training and testing of the neural network were carried out using the widely employed cross-validation approach [27]. This method consists in the use of three distinct experimental sets: a *training* set, where the domain of the function to be learned must be represented; a *validation* set, where the

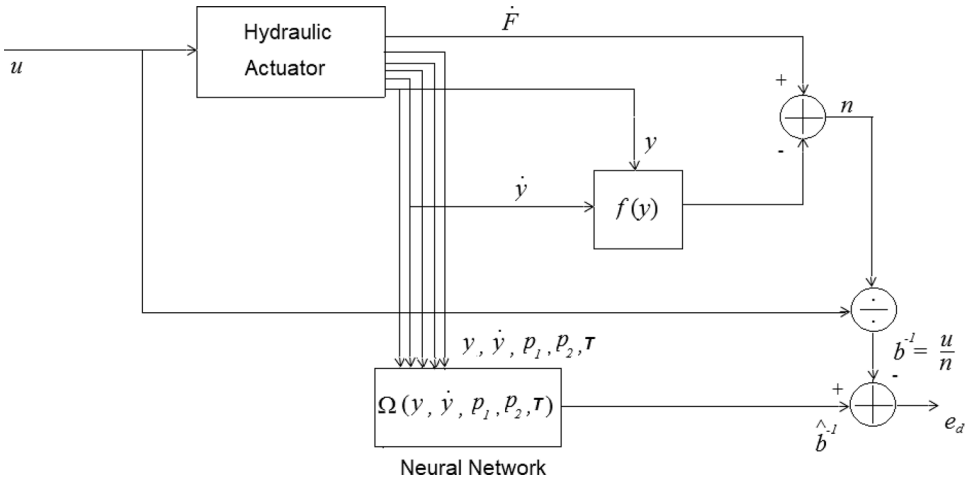

Fig. 6 Training process of the neural network

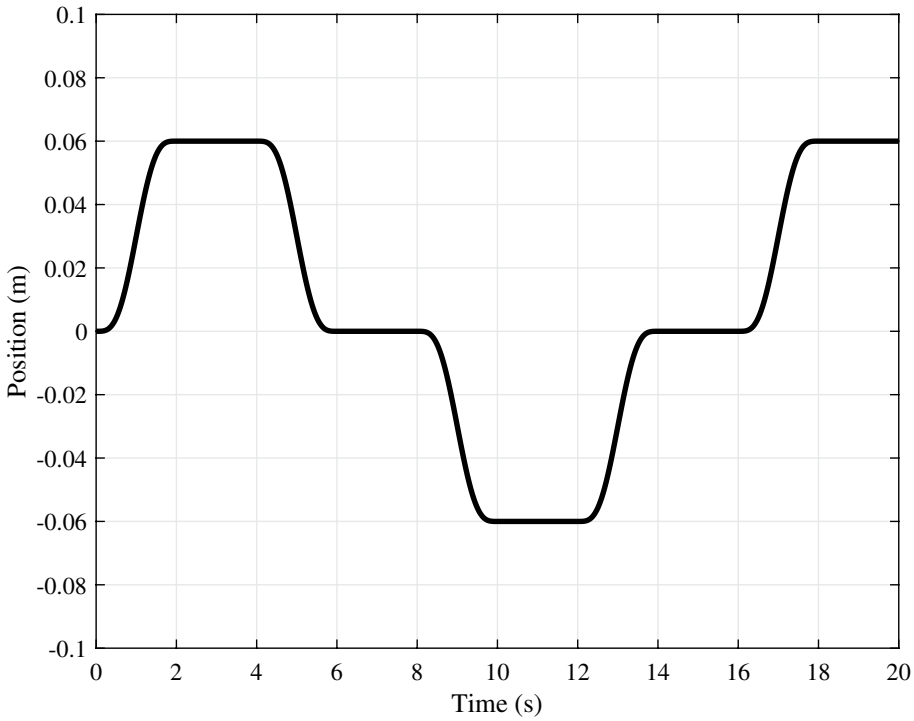

Fig. 7 Reference trajectory used for testing the trained network

Table 2 Training set per temperature value

Position trajectory (m)	Time(s)	Total time (s)	Samples	Total samples
$0.08sen(t)$	6.28	65.358	400	2000
$0.08sen(0.875t)$	7.17		400	
$0.08sen(0.625t)$	10.04		400	
$0.08sen(0.375t)$	16.74		400	
$0.08sen(0.25t)$	25.12		400	

Table 3 Validation set (T = 30 °C)

Position trajectory (m)	Time (s)	Total time (s)	Samples	Total samples
$0.08sen(0.93t)$	6.75	14.5	600	1200
$0.08sen(0.81t)$	7.75		600	

training is validated to prevent the overfitting and a *testing* set, which is used to verify whether the training was successful in generalizing the network response for a minimal desired error.

The training strategy applied in current work is based on [18] and comprises basically the choice of a set of sinusoidal desired position trajectories that are applied to the plant for generating the set of signals u calculated through a proportional controller. This profile corresponds to use sinusoidal functions with different frequencies while maintaining the piston position excursion constant for all trajectories. The highest velocities of the sinusoidal trajectories were defined according to the highest velocity of the testing polynomial trajectory. Table 2 outlines the position trajectories applied to compound the training set. These experiments were repeated for oil temperatures at 20 °C, 25 °C, 30 °C, 35 °C, 40 °C and 45 °C, totalizing 12,000 samples for the training set. The validation set trajectories are outlined in Table 3 and comprise samples generations for an oil temperature at 30 °C. Table 4 lists the test set, where the 7th-order polynomial position trajectory, presented in Fig. 7, was applied for oil temperatures at 20 °C, 30 °C and 40 °C. The positions where the piston remains stationary were discarded in the test set.

The complete procedure applied to find a suitable setup for the feedforward neural network used in current work is described in [18]. Table 5 outlines the layers, epochs and training error for such neural network. The complete sets of weights for the neural network are presented in Appendix.

The performance criterion applied to compare the different inversion sets is based on the root-mean-square error presented in Eq. (21):

$$RMSE = \sqrt{\frac{1}{P}\sum_{n=1}^{P}\left(y_t - y_p\right)^2} \tag{21}$$

where y_t and y_p are the target and predicted values, respectively, and P is the number of testing samples.

Table 4 Test set

Position trajectory (m)	Time(s)	Total time (s)	Samples	Oil Temp. (°C)	Total samples
7^{th} order polynomial (Fig. 7)	8	24	1200	20	3600
	8		1200	30	
	8		1200	40	

Table 5 Neural network setup

Neural network	Condition	Layers setup	Training epoch	Training error (RMSE)
$\Omega(y, \dot{y}, P_1, P_2, T)$	$\dot{y} \geq 0$	5–10-10–1	2000	7.5 10^{-7} V/(N/s)
	$\dot{y} < 0$	5–10-1	3500	12.2 10^{-7} V/(N/s)

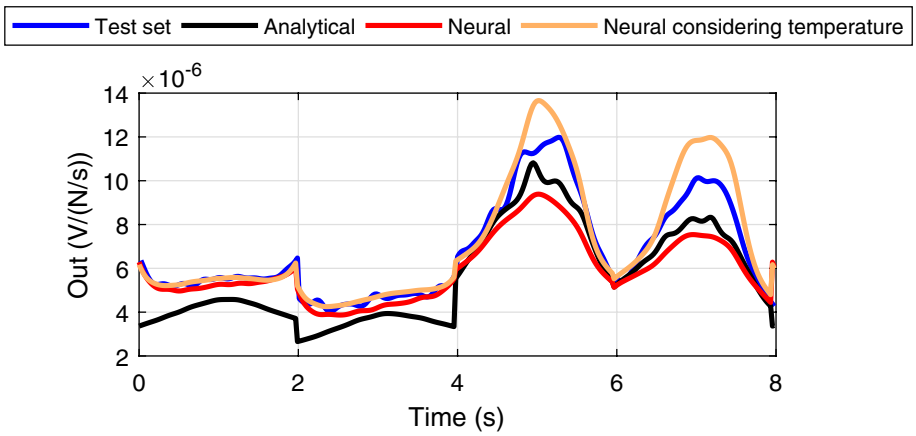

Fig. 8 Inversion sets performance comparison for hydraulic fluid at 20 °C ± 1.7 °C

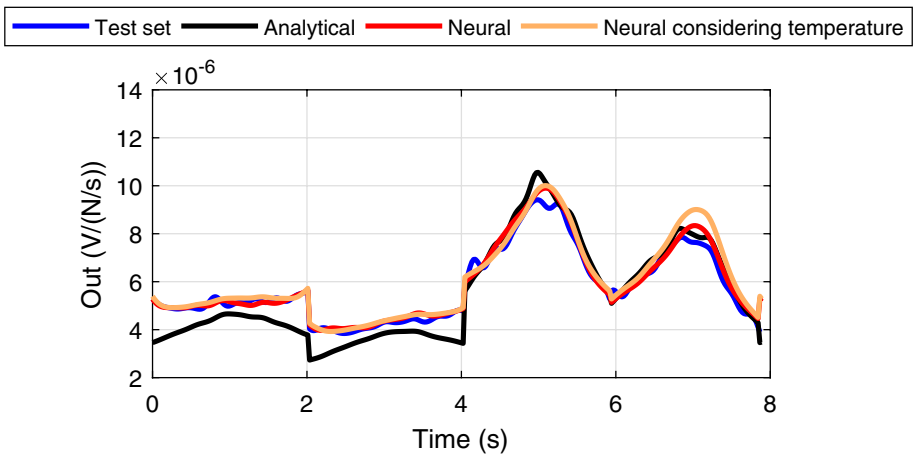

Fig. 9 Inversion sets performance comparison for hydraulic fluid at 30 °C ± 1.7 °C

Figures 8, 9, 10show the output of the inversion set obtained with three different hydraulic fluid temperatures: 20, 30 and 40 °C ± 1.7 °C. For each one of these temperatures, three outputs were compared: the one obtained with the analytic version of the inversion set, and the outputs of Ω with and without including temperature information during the training. Table 6 outlines the results of Eq. (21) taking into account the test set and the outputs from the three kinds of inversion set. The results are summarized in Table 6.

Excepting the 30 °C case, the inversion set that uses neural network considering temperature provided lower deviation in respect to the experimental data.

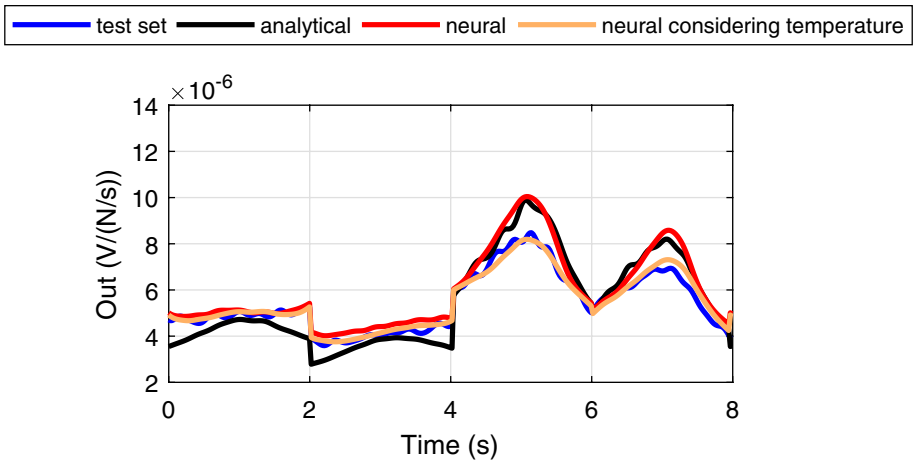

Fig. 10 Inversion sets performance comparison for hydraulic fluid at 40 °C ± 1.7 °C

Table 6 RMSE for the inversion sets

Temperature	Analytical	Neural	Neural temp
20 °C	1.25×10^{-6}	1.20×10^{-6}	8.04×10^{-7}
30 °C	7.17×10^{-7}	2.70×10^{-7}	4.31×10^{-7}
40 °C	7.55×10^{-7}	7.18×10^{-7}	2.19×10^{-7}

5 Stability analysis

Consider the open-loop model of the system, given by Eqs. (1) and (14), and the proposed control structure represented by Eqs. (11) and (17). Due to the continuity conditions implied by using the Lyapunov method for analyzing stability properties, we assume that the desired piston position $y_d(t)$ and its time derivatives up to 3rd order are continuous bounded functions. All parameters regarding the model of the mechanical subsystem are assumed to be known, but the hydraulic subsystem is subject to parameter uncertainties. The trajectory-tracking errors of the system are defined in terms of the auxiliary vector $\boldsymbol{\rho} = \left[\tilde{y}\ \dot{\tilde{y}}\ F_H\right]^T$. In order to simplify the analysis of the closed-loop system considering the effects of the cancelation residue ε, it is convenient to rewrite it in terms of a percentage error term δ, where, based on [36], we assume that $-1 < \delta \leq \bar{\delta}$. Whether the neural network reproduces exactly the function b^{-1}, then $\delta = 0$.

$$u = \Omega(y,\dot{y},p_1,p_2,T)\left[v+f\right] + \varepsilon = b^{-1}\left[\dot{F}_{Hd} - K_p\tilde{F}_H + f\right](1+\delta) \tag{22}$$

In addition, by substituting each proposed control law into the dynamics of its corresponding subsystem (Eq. (11) into Eq. (1), Eq. (17) into Eq. (14)), the behavior of the closed-loop system can be described in terms of the following auxiliary functions:

$$M\dot{z} = -K_d z + \tilde{F}_H + F_D \tag{23}$$

$$\dot{\tilde{F}}_H = -K_p\tilde{F} + \delta\left[\dot{F}_{Hd} - K_p\tilde{F}_H + f\right] \tag{24}$$

With the aid of these expressions, the stability properties of the closed-loop system are proven as follows.

Main result – Consider the closed-loop system whose dynamics is described by expressions (23) and (24), subject to an unmodeled disturbance force $\bar{F}_D$, upper-bounded by $\bar{F}_D$. When the model of the system is subject to uncertainties whose combined effect can be represented as a percentage factor δ with an upper bound $\bar{\delta}$, given an arbitrary initial condition, the controller gains can be chosen so as to ensure that the trajectory-tracking error vector $\boldsymbol{\rho}$ converges to a limited residual set R as $t \to \infty$. The amplitude of such set depends on $\bar{F}_D$, $\bar{\delta}$ and the controller's gains. Moreover, if $F_D = 0$ and the output of the neural network used in the hydraulic subsystem control law cancels uncertainty effects, then $\|\boldsymbol{\rho}\| \to 0$ as $t \to \infty$.

Proof: Consider the Lyapunov candidate function.

$$V = \frac{1}{2}\left(HMz^2 + P\tilde{y}^2 + \tilde{F}_H^2\right) = \frac{1}{2}\left[\tilde{y}\ \dot{\tilde{y}}\ \tilde{F}_H\right]\begin{bmatrix} HM\lambda^2+P & \lambda HM & 0 \\ \lambda HM & HM & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{y} \\ \dot{\tilde{y}} \\ \tilde{F}_H \end{bmatrix} \tag{25}$$

where P and H are positive constants. By taking the time derivative of Eq. (25), substituting into it the auxiliary expressions (23) and (24) and defining $P = 2\lambda K_d H$ one obtains:

$$\dot{V} = -\left(\begin{bmatrix} \tilde{y} & \dot{\tilde{y}} & \tilde{F}_H \end{bmatrix} \begin{bmatrix} \lambda^2 H K_d & 0 & -\frac{1}{2}H\lambda - \frac{\delta\alpha_1}{2} \\ 0 & HK_d & -\frac{1}{2}H - \frac{\delta\alpha_2}{2} \\ -\frac{1}{2}H\lambda - \frac{\delta\alpha_1}{2} & -\frac{1}{2}H - \frac{\delta\alpha_2}{2} & K_p - \delta\alpha_3 \end{bmatrix} \begin{bmatrix} \tilde{y} \\ \dot{\tilde{y}} \\ \tilde{F}_H \end{bmatrix} + \begin{bmatrix} \tilde{y} & \dot{\tilde{y}} & \tilde{F}_H \end{bmatrix} \begin{bmatrix} \lambda H F_D \\ H F_D \\ \Psi_d \delta \end{bmatrix} \right) = -\left(\boldsymbol{\rho}^T \left(\mathbf{N}_2\right)\boldsymbol{\rho} + \boldsymbol{\rho}^{\mathbf{T}}\boldsymbol{\Delta}\right) \tag{26}$$

where $\alpha_1, \alpha_2, \alpha_3$ and Ψ_d are the following bounded auxiliary terms, computed considering the following expressions:

$$\alpha_1 = \left(\left(M\lambda + K_d\right)\frac{\lambda K_d}{M} - \frac{BK_d}{M}\lambda \right)$$

$$\alpha_2 = \left(\left(M\lambda + K_d\right)\left(\lambda + \frac{K_d}{M}\right) - \lambda K_d - \frac{B}{M}(K_d + \lambda M) + (A_1^2 f_1 + A_2^2 f_2) \right)$$

$$\alpha_3 = \left(\left(B - K_d\right)\frac{1}{M} - (\lambda + K_p) \right) \tag{27}$$

$$\dot{F}_{Hd} = \left(-K_d - M\lambda + B\right)\ddot{\tilde{y}} - K_d\lambda\dot{\tilde{y}} + B\ddot{y}_d + M\dddot{y}_d$$

$$\ddot{\tilde{y}} = -\lambda\dot{\tilde{y}} - M^{-1}K_d\left(\dot{\tilde{y}} + \lambda\tilde{y}\right) + M^{-1}\tilde{F}_H - M^{-1}F_D.$$

Note that, as v_1 and v_2 are greater than zero, from Eq. (12), f_1 and f_2 are bounded functions. From Eq. (26), if $F_D = \delta = 0$, it can be readily verified by applying the Sylvester criterion that $K_d K_p > \frac{1}{2}H$ is a sufficient condition to ensure that $\dot{V}(t)$ is negative definite. Therefore, $\|\rho\| \to 0$ as $t \to \infty$. On the other hands, if $\delta \neq 0$ and/or $F_D \neq 0$, by the same criterion,N_2 can be made symmetric and positive definite if

$$K_p > \frac{1}{(\lambda^2 K_d H)(1+\delta)}\left(\left(\frac{1}{2}H\lambda + \frac{\delta\alpha_1}{2}\right)^2 + \left(\frac{1}{2}H + \frac{\delta\alpha_2}{2}\right)^2\lambda^2 + \lambda^2 K_d H\delta(B - K_d - \lambda M)\frac{1}{M} \right) \tag{28}$$

which is satisfied by choosing K_p as an appropriate value. With the feedback gains and parameter values presented in Sect. 6, and using $H = 9 \times 10^5$, the criterion defined in Eq. (28) holds for $\bar{\delta} = 0.25$, which means that the stability condition holds for estimation errors up to ±25% in the output given by the neural network. Within the region, where this condition is met, application of the Rayleigh–Ritz theorem combined with the Cauchy–Schwartz inequality to Eq. (26) yields:

$$\dot{V} \leq -\lambda 2^T \|\rho\| \|\Delta\|_{min} \tag{29}$$

where $\lambda 2_{min}$ is the minimum eigenvalue of N_2. Under the assumption that both F_D and δ are upper-bounded, and since Ψ_d has also an upper limit $\bar{\Omega}_d$ because it is derived from the desired trajectory, we have that $\|\Delta\|$ is upper-bounded by $\bar{\Delta} = \sqrt{\lambda^2 H^2 \bar{F}_D^2 + H^2 \bar{F}_D^2 + \bar{\delta}^2 \bar{\Psi}_D^2}$. Therefore, the condition $\dot{V}(t) < 0$ is attained if

$$\|\rho\| > \frac{\bar{\Delta}}{\lambda 2_{min}} \tag{30}$$

From Eq. (30), it is clear that any system trajectory with initial condition $\rho(0)$ lying outside a ball with radius $\bar{\Delta}/(\lambda 2_{\min})$ must converge and remain confined to such ball as $t \to \infty$, thus ensuring that $\|\rho(t)\|$ is a limited quantity. Moreover, if there are no disturbances and the action of the neural network can overcome parametric uncertainties, we have $F_D = \delta = 0$ and the closed-loop tracking errors converge asymptotically to zero. This completes the proof.

6 Experimental evaluation

Experimental evaluation was carried out by means of two tests. First, the actuator piston was led to track the polynomial trajectory depicted in Fig. 7. In the second test, a sinusoid with amplitude 0.08 m and period 8.3 s was tracked.

We employed two inversion methods for obtaining the control input applied to the actuator valve: the analytical function with nominal parameters and the output of the neural network Ω taking temperature into account. The feedback gains used in the proposed controller were kept fixed with the values of $K_d = 5000\ \text{s}^{-1}$, $\lambda = 150\ \text{s}^{-1}$ and $K_p = 200\ \text{s}^{-1}$.

For comparison purposes, we also performed experiments using a classical PID controller, which was applied both with and without a feedforward term considering the desired velocity trajectory. Such controller is described by Eqs (31) and (32).

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} + \dot{y}_d F_f \tag{31}$$

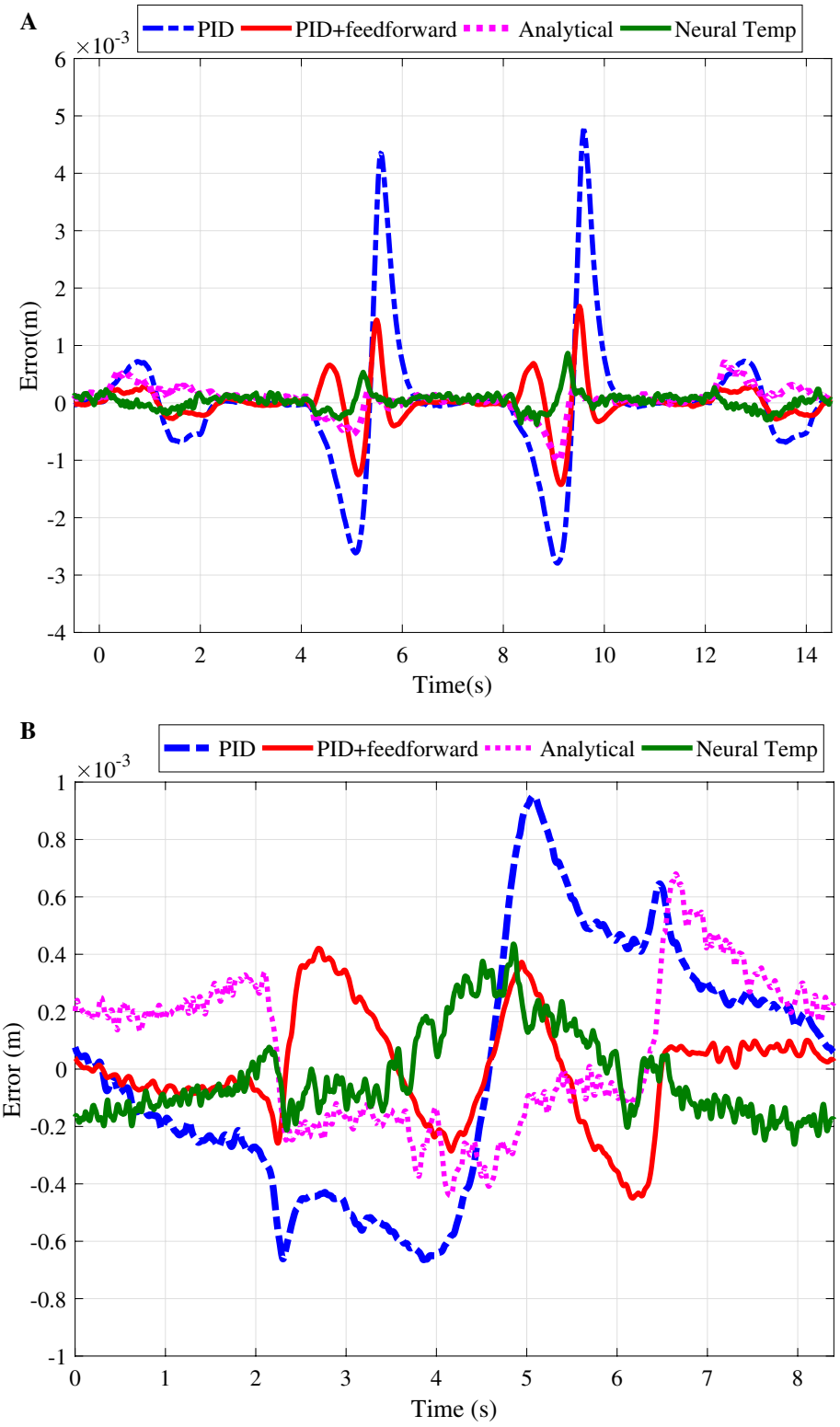

Fig. 11 Position trajectory-tracking errors for hydraulic fluid at 20 °C ± 1.7 °C

$$F_f = \begin{cases} K_{fp}, \ \dot{y}_d \geq 0 \\ K_{fn}, \ \dot{y}_d < 0 \end{cases} \quad (32)$$

where $e(t) = y_d - y$.

The feedback gains values are $K_p = 420$, $K_i = 2018$, $K_d = 0.9$, $K_{fp} = 14$ and $K_{fn} = 40$. All gains values were obtained according the methodology used in [19]. Experimental results are shown in Figs. 11, 12, 13 for both the polynomial and sinusoidal trajectories. Tables 7 and 8 present

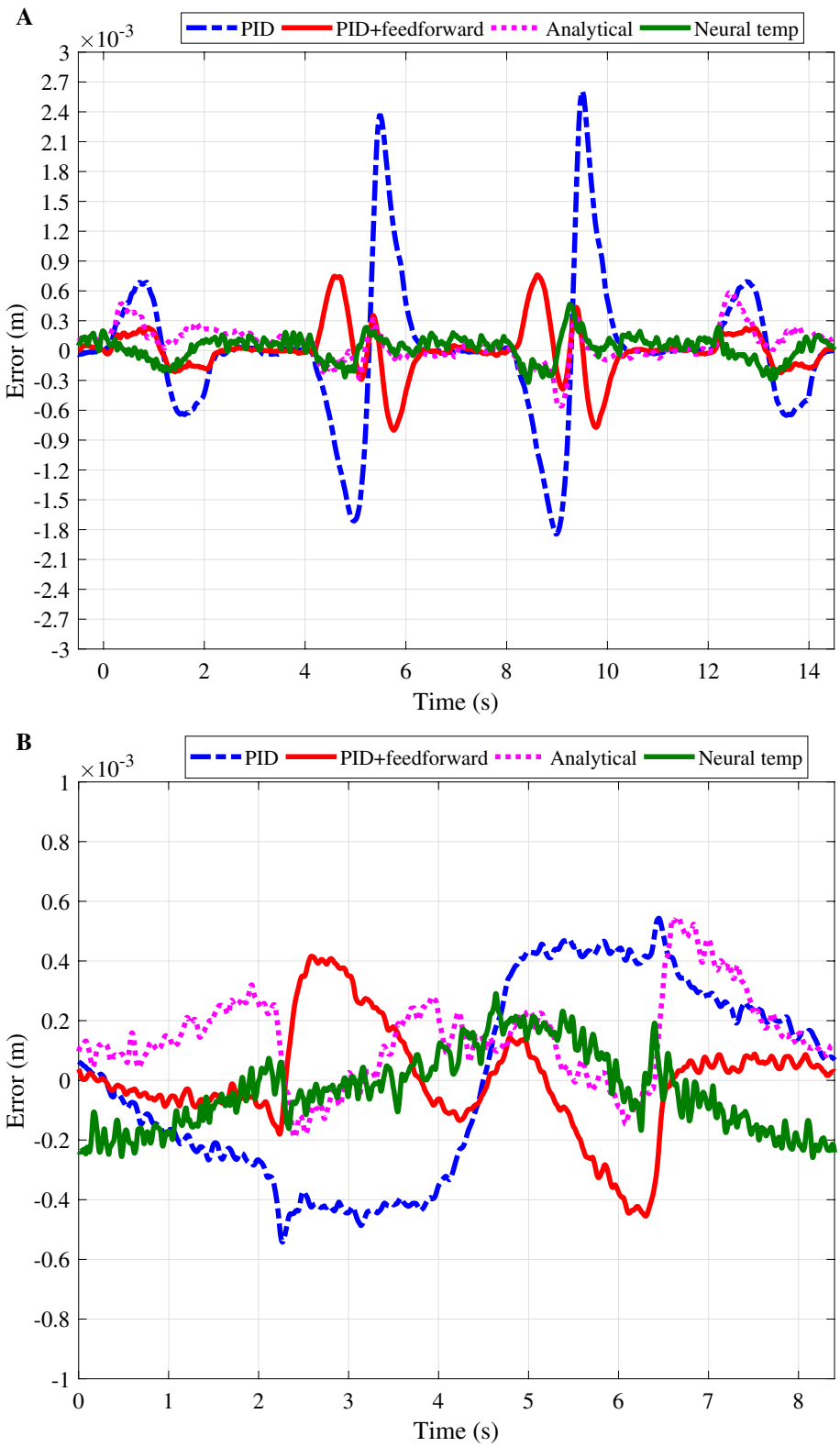

Fig. 12 Position trajectory-tracking errors for hydraulic fluid at 30 °C ± 1.7 °C

the RMSE results and the highest values of the position error.

Results show lower RMSE and peak errors for the proposed controller using neural network-based inversion method. The main objective of the proposed approach is to facilitate its practical application. Therefore, the improvement of the trajectory-tracking performance of a feedback

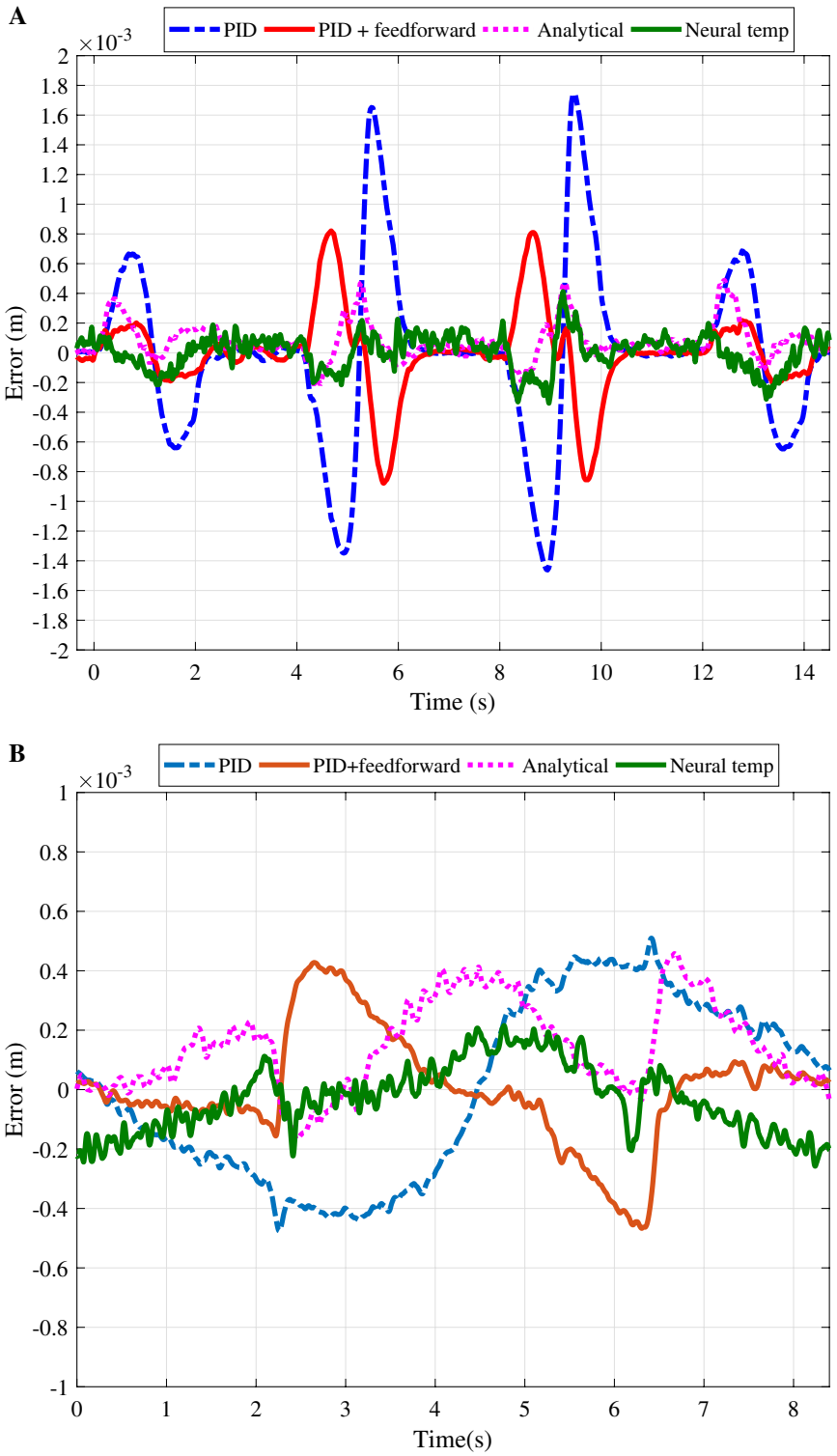

Fig. 13 Position trajectory-tracking errors for hydraulic fluid at 40 °C ± 1.7 °C

linearization-based controller can be considered an additional advantage of the proposed control algorithm.

7 Conclusions

In the present work, we propose the use of a feedforward multi-layer neural network to replace an analytical function in the model of a hydraulic actuator, usually applied to model-based synthesis of controllers for such systems.

Table 7 RMSE for the polynomial position trajectory error

Temperature (°C)	PID		PID + FF		Analytical		Neural Temp	
	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)
20	11.0	47.6	3.61	16.7	2.47	10.0	1.37	8.6
30	7.05	25.6	2.53	7.8	1.45	5.6	1.17	4.6
40	5.77	17.4	2.81	8.6	1.30	4.46	1.1	4.0

Table 8 RMSE for the sinusoidal position trajectory error

Temperature (°C)	PID		PID + FF		Analytical		Neural Temp	
	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)	(RMSE) (10^{-4} m)	Highest error (10^{-4} m)
20	4.20	9.4	1.83	4.2	2.67	6.7	1.5	4.3
30	3.12	5.4	1.76	4.5	1.63	5.4	1.26	2.8
40	2.88	5.1	1.83	4.5	1.8	4.5	1.08	2.4

This strategy aims at to facilitate the application of feedback linearization-based control schemes to such systems. When compared with traditional procedures, this strategy allows avoidance of the extensive experimental work and the high levels of uncertainty involved in developing the models for such control schemes in high- precision applications. A rigorous Lyapunov-based theoretical stability analysis was presented and corresponding experimental results obtained clearly indicate that besides requiring less experimental effort to implement, the proposed inversion scheme yields smaller trajectory-tracking errors than these obtained by the traditional methods. For these reasons, the proposed inversion method can be considered as an effective way for experimental implementation of model-based strategies in the control of hydraulic drives, also comprising industrial-scale application. Future work will focus the expansion of the proposed method to encompass other nonlinear phenomena involved in the operation of hydraulic actuators that were not addressed in the present work, such as dry friction effects analysis and the effects of unknown external load disturbances.

Table 9 Neural network normalization

Neural Network input	Maximal value	Minimal value
y	0,1 m	−0,1 m
$\dot{y}$	0,15 m/s	−0,1 m/s
p_1	2,5 MPa	0,01 MPa
p_2	5,0 MPa	0,01 MPa
T	70 °C	10 °C
Out(V/(N/s))	4,0 10^{-5}	0

Appendix

(1) Normalization Equations.

Table 10 Weights Layer 1 to Layer 2

Layer 2	Layer 1					
	Input 1	Input 2	Input 3	Input 4	Input 5	Thersholds
N1	−6.36597341310150	16.5077714974687	−3.13779679024344	−18.1313423228545	19.1678700752719	−0.562026919937644
N2	−8.69429863342924	5.06033243223119	−0.615825836638685	−6.13822845938630	2.80387447130696	0.00488997425278474
N3	−7.66496868816365	16.5239629829962	−5.11711223652556	−15.0572384944819	19.0518216575182	−0.0679496314731981
N4	−6.41247416253936	17.2924893198281	−3.36056571552519	−19.4346657132758	19.6012638862174	−0.281950537943531
N5	−5.79233491418532	4.41568405731381	−0.420033366932498	−7.21026446844022	1.07152165510392	0.196430969159193
N6	−0.930281391513413	28.0409574736910	−0.388212865023555	−34.0339838237914	−4.84403394446628	−0.110381597293962
N7	1.78930671633738	21.0034364853193	−4.64830365969991	−34.4160538913726	−4.20112842679759	−0.00738739915449935
N8	−6.62337953930139	16.6612381565198	−5.06788445087522	−16.8754755605125	19.5519620634922	−0.241323765403628
N9	−10.8475174854601	5.89194790864145	−0.184718373359893	−7.01898496394809	1.28152168925314	0.00951768135588512
N10	−11.1830702693174	16.0102049670144	−11.9250317919372	−14.5481595661690	18.8026442348017	−0.943139460065714

Table 11 Weights Layer 2 to Layer 3 (neuron 1–6)

Layer 3	Layer 2					
	N1	N2	N3	N4	N5	N6
N1	0.126459525225218	0.491210577717582	0.129063869159268	0.125353706236741	−0.748042080670698	0.935113469523096
N2	0.126458998206672	0.491210866192207	0.129063742388102	0.125353195040675	−0.747898913992743	0.935110878706768
N3	0.107957110349282	0.575557626698085	0.111998455699491	0.107516586202091	0.495684426003983	0.571118617040776
N4	0.107956918435619	0.575560530041035	0.111998280748945	0.107516309769334	0.495686738439458	0.571128405157357
N5	0.107957041934123	0.575558667350142	0.111998393292618	0.107516487482378	0.495685254334811	0.571122106810099
N6	0.126462765975914	0.491208807843965	0.129064642281605	0.125356849711821	−0.748922234967505	0.935129400805221
N7	0.107956943476562	0.575560150390586	0.111998303772925	0.107516345982491	0.495686435128470	0.571127112479047
N8	0.126479950526594	0.491199451158912	0.129068721598002	0.125373521969876	−0.753589571052748	0.935213683815015
N9	0.126457478855718	0.491211694556218	0.129063379566586	0.125351720453222	−0.747486208600631	0.935103404585167
N10	0.126463217351332	0.491208560154631	0.129064749949050	0.125357287380284	−0.749044776971014	0.935131617300847

Table 12 Weights Layer 2 to Layer 3 (neuron 7–10)

Layer 3	Layer 2				
	N7	N8	N9	N10	Thersholds
N1	.373295396644175	0.131877348347851	1.17181119280113	0.151884569649175	−0.250622440695391
N2	0.373295394294990	0.131876920101171	1.17181254419206	0.151884477647797	−0.250622442381323
N3	−1.15370297821913	0.109561783132085	1.73642578734396	0.257335742637122	−0.206963275665140
N4	−1.15375165443010	0.109561618687375	1.73644313287530	0.257326739232410	−0.206963287846359
N5	−1.15372034574972	0.109561724507814	1.73643198794539	0.257332543418107	−0.206963279208723
N6	0.373295407927704	0.131879980051229	1.17180285651655	0.151885136526690	−0.250622443187820
N7	−1.15374524114792	0.109561640031112	1.73644085841113	0.257327933558995	−0.206963285074935
N8	0.373295469311495	0.131893917657162	1.17175879879176	0.151888134015839	−0.250622437585197
N9	0.373295388669728	0.131875685776040	1.17181646015918	0.151884212121376	−0.250622443278652
N10	0.373295409477423	0.131880346427270	1.17180169772232	0.151885215143812	−0.250622442413081

Table 13 Weights Layer 3 to Layer 4

Layer 4	
Layer 3	N1
N1	−0.172473808663452
N2	−0.172473801742085
N3	−0.171374979647380
N3	−0.171373895519672
N5	−0.171374591608419
N6	−0.172473852776407
N7	−0.171374037480287
N8	−0.172474084268685
N9	−0.172473781053205
N10	−0.172473858679202
Threshold	−0.125522864338090

Input normalization function: Table 9

$$N(x) = \frac{0,9-0,1}{X_{\max}-X_{\min}}\left(x - X_{\min}\right) + 0,1 \tag{33}$$

Output normalization function:

$$D(y) = \frac{y-0,1}{0,9-0,1}\left(X_{\max}-X_{\min}\right) + X_{\min} \tag{34}$$

(2) Weights for neural network when $\dot{y} \geq 0$. Tables 10, 11, 12, 13
(3) Weights for neural network when $\dot{y} < 0$. Tables 14, 15

Table 14 Weights Layer 1 to Layer 2

Layer 2	Layer 1					
	Input 1	Input 2	Input 3	Input 4	Input 5	Thershold
N1	-2.58565349314786	6.38641893169279	−0.812147514511006	−2.39895192054532	1.22251870196674	−0.262485933404903
N2	−0.00881485378641554	49.1290075678509	−0.0877690540511511	−13.5134325232682	21.7830558149263	−5.07872856447161
N3	−0.439861518090415	38.4805377432664	−0.0885462014060751	−5.40040530757173	5.89991619113764	−4.59461094484537
N4	−2.02884345887898	6.38634356601623	−1.25491718592026	−2.44379111677041	1.22303021586865	−0.262798151425950
N5	−0.164164317019509	38.5390950634192	−0.0882040845171557	−5.40989303243027	5.99861819265197	−4.64352988862444
N6	−2.86532674415012	6.38642003413566	−0.690577695808889	−2.39910201296318	1.21892373429239	−0.262236687291465
N7	−0.00947121815164545	39.8585653030216	−0.0849164573625043	−6.89377228642218	8.88733178449677	−4.76113093820530
N8	−0.446640954539744	41.8140135117459	−0.0852216500547128	−6.15428308564890	5.83815727108116	−4.77528661738925
N9	−0.716182224046497	42.7028582773536	−0.116780472645975	−5.48915958422523	3.34780873615430	−4.78179666417069
N10	−2.81709749656707	6.38641457960294	−0.748161473952788	−2.39886216473928	1.22005337430874	−0.262250354437234

Table 15 Weights Layer 2 to Layer 3

Layer 3	
Layer 2	N1
N1	−0.285065864572983
N2	−0.698319822137451
N3	−0.323048773141637
N4	−0.408130365794524
N5	−0.343272981789387
N6	−0.278965150363627
N7	−0.334836542148793
N8	−0.342043667078291
N9	−0.312234386322982
N10	−0.281832400763363
Threshold	1.26790601121090

References

1. Watton J (2009) Fundamentals of fluid power control. Cambridge University Press, Cambridge
2. Guo Q, Zhang Y, Celler BG, Su SW (2016) Backstepping control of electro-hydraulic system based on extended-state-observer with plant dynamics largely unknown. IEEE Trans Ind Electron 63:6909–6920. https://doi.org/10.1109/TIE.2016.2585080
3. Yang X, Zheng X, Chen Y (2018) Position tracking control law for an electro-hydraulic servo system based on backstepping and extended differentiator. IEEE/ASME Trans Mechatronics 23:132–140. https://doi.org/10.1109/TMECH.2017.2746142
4. Sun C, Fang J, Wei J, Hu BO (2018) Nonlinear Motion Control of a Hydraulic Press Based on an Extended Disturbance Observer. IEEE Access 6:18502–18510. https://doi.org/10.1109/ACCESS.2018.2813317
5. Yao J, Deng W, Jiao Z (2017) RISE-based adaptive control of hydraulic systems with asymptotic tracking. IEEE Trans Autom Sci Eng 14:1524–1531. https://doi.org/10.1109/TASE.2015.2434393
6. Yao J, Deng W (2017) Active disturbance rejection adaptive control of hydraulic servo systems. IEEE Trans Ind Electron 64:8023–8032. https://doi.org/10.1109/TIE.2017.2694382
7. Angue Mintsa H, Venugopal R, Kenné JP, Belleau C (2012) Feedback linearization-based position control of an electrohydraulic servo system with supply pressure uncertainty. IEEE Trans Control Syst Technol 20:1092–1099. https://doi.org/10.1109/TCST.2011.2158101
8. Fales R, Kelkar A (2009) Robust control design for a wheel loader using H∞ and feedback linearization based methods. ISA Trans 48:312–320. https://doi.org/10.1016/j.isatra.2009.01.007
9. Seoa J, Venugopal R, Kennéa JP (2007) Feedback linearization based control of a rotational hydraulic drive. IFAC Proc 7:940–945. https://doi.org/10.1016/j.conengprac.2007.02.009
10. Xinliang L, Wang X (2013) Adaptive sliding mode position control of electro-hydraulic servo system with single-rod actuators. IEEE Int Symp Robot Sensors Environ 2013:220–225. https://doi.org/10.1109/ROSE.2013.6698446
11. Ghazali R, Ngadengon R, Sam YM et al (2011) Chaotic trajectory tracking of an electro-hydraulic actuator system using discrete sliding mode control. IEEE Int Conf Control Syst Comput Eng 2011:500–506. https://doi.org/10.1109/ICCSCE.2011.6190577
12. Zhu L, Wang Z, Liu Y, Song W (2016) Sliding-mode dynamic surface control for MDF continuous hot pressing hydraulic system. Chin Control Decis Conf 2016:2509–2514. https://doi.org/10.1109/CCDC.2016.7531407
13. Coelho LDS, Cunha MAB (2011) Adaptive cascade control of a hydraulic actuator with an adaptive dead-zone compensation and optimization based on evolutionary algorithms. Expert Syst Appl 38:12262–12269. https://doi.org/10.1016/j.eswa.2011.04.004
14. Ji Y, Huang GH, Sun W (2015) Risk assessment of hydropower stations through an integrated fuzzy entropy-weight multiple criteria decision making method: a case study of the Xiangxi River. Expert Syst Appl 42:5380–5389. https://doi.org/10.1016/j.eswa.2014.12.026
15. Na J, Li Y, Huang Y et al (2020) Output feedback control of uncertain hydraulic servo systems. IEEE Trans Ind Electron 67:490–500. https://doi.org/10.1109/TIE.2019.2897545
16. Borges FAP, Perondi EA, Cunha MAB, Sobczyk MR (2016) A hybrid feedback linearization + neural network control algorithm applied to a hydraulic actuator. In: 9th FPNI Ph.D. Symposium on Fluid Power, FPNI 2016. ASME, Florianópolis, SC, Brazil, p V001T01A037
17. Pedro J, Ekoru J (2013) NARMA-L2 control of a nonlinear half-car servo-hydraulic vehicle suspension system. Acta Polytech Hungarica 10:5–26
18. Borges FAP (2017) Controle em cascata de um atuador hidráulico utilizando redes neurais. Universidade Federal do Rio Grande do Sul
19. Yao Z, Yao J, Yao F et al (2020) Model reference adaptive tracking control for hydraulic servo systems with nonlinear

neural-networks. ISA Trans 100:396–404. https://doi.org/10.1016/j.isatra.2019.11.027
20. Yang Y, Balakrishnan SN, Tang L, Landers RG (2013) Electro-hydraulic piston control using neural MRAC based on a modified state observer. IEEE/ASME Trans Mechatron 18:867–877. https://doi.org/10.1109/TMECH.2012.2193592
21. Zhu L, Wang Z, Zhou Y (2018) Adaptive neural network saturated control for MDF continuous hot pressing hydraulic system with uncertainties. IEEE Access 6:2266–2273. https://doi.org/10.1109/ACCESS.2017.2782727
22. Lin YC, Chen D, Chen M et al (2018) A precise BP neural network-based online model predictive control strategy for die forging hydraulic press machine. Neural Comput Appl 29:585–596. https://doi.org/10.1007/s00521-016-2556-5
23. Yao Z, Yao J, Sun W (2019) Adaptive RISE control of hydraulic systems with multilayer neural-networks. IEEE Trans Ind Electron 66:8638–8647. https://doi.org/10.1109/TIE.2018.2886773
24. Lewis FL, Yeşildirek A, Liu K (1996) Multilayer neural-net robot controller with guaranteed tracking performance. IEEE Trans Neural Networks 7:388–399. https://doi.org/10.1109/72.485674
25. Moreno-Valenzuela J, Aguilar-Avelar C (2018) Adaptive neural network control of the Furuta pendulum. Intell Syst Control Autom Sci Eng 88:93–118. https://doi.org/10.1007/978-3-319-58319-8_6
26. Moreno-Valenzuela J, Montoya-Cháirez J, Santibáñez V (2020) Robust trajectory tracking control of an underactuated control moment gyroscope via neural network–based feedback linearization. Neurocomputing 403:314–324. https://doi.org/10.1016/j.neucom.2020.04.019
27. Simon Haykin (1999) Neural networks: a comprehensive foundation, 2nd ed. Pearson
28. Prabel R, Aschemann H (2014) Nonlinear adaptive backstepping control of two coupled hydraulic servo cylinders. In: 2014 American Control Conference. IEEE, pp 1310–1315
29. Narendra KS, Parthasarathy K (1990) Identification and control of dynamical systems using neural networks. IEEE Trans neural networks 1:4–27. https://doi.org/10.1109/72.80202
30. Slotine J-JE, Li W (1991) Applied nonlinear control. Prentice Hall, New Jersey
31. Khalil HK (2014) Nonlinear Systems, 3rd ed. Pearson
32. Cunha MAB, Guenther R, De Pieri ER, De Negri VJ (2002) Design of cascade controllers for a hydraulic actuator. Int J Fluid Power 3:35–46. https://doi.org/10.1080/14399776.2002.10781136
33. Slotine JJE, Weiping L (1988) Adaptive manipulator control: a case study. IEEE Trans Automat Contr 33:995–1003. https://doi.org/10.1109/9.14411
34. Valdiero AC (2005) Control of hydraulic robots with friction compensation. Universidade Federal de Santa Catarina
35. Wang J, Guofang Gong, Huayong Yang (2008) Control of bulk modulus of oil in hydraulic systems. In: 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, pp 1390–1395
36. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4:251–257. https://doi.org/10.1016/0893-6080(91)90009-T
37. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals, Syst 2:303–314. https://doi.org/10.1007/BF02551274
38. Chi-Chung Cheung, Ng S-C, Lui AK (2012) Improving the Quickprop algorithm. In: The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE, pp 1–6