Vietnam Journal of Agricultural Sciences

ISSN 2588-1299

VJAS 2024; 7(4): 2294-2304
https://doi.org/10.31817/vjas.2024.7.4.04

Numerical Comparison of PI and Neural Network-Based Controllers for the Hydrostatic Unit in Hydro-mechanical Transmissions of Self-propelled Vehicles

Dang Ngoc Danh[1*], Bui Viet Duc[2], Nguyen Thi Hue[2] & Vu Cong Canh[2]

[1]Faculty of Engineering, Vietnam National University of Agriculture, Hanoi 12400, Vietnam

[2]Institute of Engineering Technology Development, Vietnam National University of Agriculture, Hanoi 12400, Vietnam

Abstract

The application of hydro-mechanical transmissions is recently the trend in agricultural vehicles where a continuously variable transmission ratio has advantages. Hydro-mechanical transmissions provide efficient power transfer while maneuverability is still maintained, and therefore, fuel efficiency is enhanced. Nevertheless, the main issue in their employment is a precise control of the hydrostatic unit, whose physical characteristics are highly nonlinear and affected by unknown disturbances. In order to exploit the advantages of the system, the transmission ratio of the hydrostatic unit needs to be controlled properly to maintain the optimal working point of the internal combustion engine (ICE). This article presents numerical comparison results of a proportional-integral (PI) and a neural network (NN) based controller applied to the hydrostatic unit of a hydro-mechanical transmission system, which was designed to be deployed on a self-propelled agricultural vehicle. The controls were established in a discrete-time domain aiming at a practical outcome, where the control algorithm could be implemented on an industrial computer to perform the control tasks.

Keywords

Hydrostatic transmissions, hydro-mechanical transmissions, transmission ratio control, model-free control, neural network-based control

Received: May 22, 2024
Accepted: November 28, 2024

Correspondence to
dndanh@vnua.edu.vn

ORCID
Dang Ngoc Danh
https://orcid.org/0000-0002-0481-1666

Introduction

Environmental policies regarding emissions from internal combustion engines (ICEs), including agricultural machinery, are imposed by regulations in automotive technology. These policies force manufacturers towards hybrid solutions (Marcor *et al*., 2017; Yu *et al*., 2019). Over the past few years, the development of a drivetrain

for agricultural machines has involved the implementation of continuously variable transmissions to keep the ICEs working at minimal fuel consumption points. Various types of continuously variable transmissions (CVTs) have been explored, among these, the hydro-mechanical transmission, sometimes referred to as power split gearbox, provides the balance between fuel efficiency and manufacturing cost (Marcor & Rossetti, 2011; Kwon & Ivantysynova, 2020), and therefore, has become the trend in many applications, including agricultural machinery.

Hydro-mechanical transmissions are comprised of two main components: a planetary gear set and a hydrostatic unit. The power, provided by the ICE, is transmitted to the drive axle along two parallel paths - the mechanical path and the hydrostatic path. Power transmitted in this way allows the engine to simultaneously exploit the advantages of the continuously variable characteristics of hydrostatic units and the efficiency of mechanical transmissions (Schulte & Gerland, 2011; Kwon & Ivantysynova, 2020). The structure principle of a hydro-mechanical transmission in a self-propelled vehicle is demonstrated in **Figure 1.**

Regarding the control aspect, the practical issue in the application of hydro-mechanical transmissions is the achievement of precise control for the hydrostatic unit, whose physical characteristics are highly nonlinear and affected by unknown disturbances such as kinematic viscosity, fluid temperature variations, leakage oil flow, and the elasticity of the hydraulic hoses (Nawrocka & Kwasniewski, 2006). Proportional - integral - derivative (PID) controllers are still predominant in industrial applications of hydrostatic transmissions. Their performance, however, is not adequate to provide accurate control throughout the whole working range of the system (Kwasniewski *et al.*, 2003). Current advanced control designs for hydrostatic transmissions are mostly based on an explicit mathematical model of the system and many advanced concepts can be found in the literature. In the work of Wu & Lee (1996), a self–tuning pole–placement adaptive control theory was implemented for different configurations of a hydrostatic transmission. A Linear-Quadratic-Gaussian control design was applied and the obtained regulator was implemented in different ways for the system in a study of Lennevi & Palmberg (1995) using non-linear and linearized models, among others.

The performance of model-based controllers obviously depends on the accuracy of the mathematical model of the system. Under the impact of unknown disturbances and uncertainties in the model, the control performance reduces. Model-free control designs are good solutions for this issue. In Dang & Aschemann (2020), neural network-based controllers were implemented for nonlinear compensation and an improvement of hydraulic motor velocity control performance. In this control scheme, a mathematical model was partly used, which reduced the accuracy of the employed dynamics model. The study of Danh & Aschemann (2021a) presented the application results of a sliding mode control for motor velocity tracking. The study exploited the advantages of an almost model-free control design that did not rely on a complete model of the system. In further work of Danh & Aschemann (2021b), a generalized proportional-integral controller for hydrostatic transmission was investigated, which was also applied to control the angular velocity of the hydraulic motor of the system.

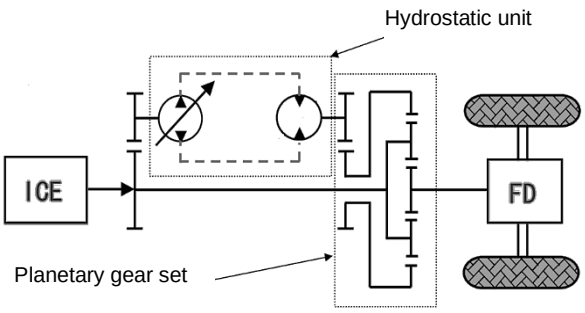

Figure 1. Principle of a hydro-mechanical transmission

In this article, a neural network-based model free controller was designed for transmission ratio regulation of the hydrostatic unit in a hydro-mechanical transmission system, which has not been addressed in the literature thus far. The addressed controller performed the task aiming to maintain a constant speed of the ICE at its optimal fuel working point regardless of the load change exerting on the hydraulic motor shaft. The controller was established in discrete-time form aiming at a practice-oriented outcome. A numerical comparison result with an industrial PI controller is also presented to provide a closer look at the performance of the designed controller.

Materials and Methods

The structure of the hydro-mechanical transmission in a self-propelled agricultural vehicle is shown in **Figure 1**. The motive power was supplied from an ICE, and this power was transmitted to the final drive (FD) of the vehicle through two parallel paths - the hydrostatic path and the planetary gear set path. The transmission ratio of the planetary gear set was fixed and the transmission ration of the whole system could be varied continuously by changing the transmission ratio of the hydrostatic unit using an automatic control device.

In the provided configuration, the hydrostatic unit was comprised of two components: a variable volumetric displacement pump and a fixed volumetric motor. They were connected in a closed hydraulic circuit. The pump was driven by the ICE, which partly supplied mechanical power to the hydraulic system. This power part was converted to hydraulic power in the form of pressurized fluid flow, which was transmitted to the hydraulic motor. At the motor, it was converted back to the mechanical rotational power of the motor shaft, then, added to the mechanical path. The principle structure of the hydrostatic drivetrain is presented in **Figure 2**. This system used a speed sensor to provide feedback for the control design.

The model was established using the Matlab/SIMULINK simulation toolbox with the utilization of the Simscape library. Based on the principle structure of the hydrostatic drivetrain in Figure 2, the model was comprised of the following three components.

The ICE model component This block represented a model of an internal combustion engine (**Figure 3**), which stood for the motive power of the vehicle. The model reacted to the load torque applied at the shaft in a similar way as a real engine does. This meant that the velocity of the engine crankshaft varied according to the value of the load. A controller was equipped to maintain a constant velocity of the engine despite load torque changes.

The load model component This model component simulated the load acting on the output shaft of the hydraulic motor (**Figure 4**). The load torque value was subjected to changes over time, which generated variations in the torque transmitted back to the engine through the hydrostatic transmission unit.

The hydrostatic unit model component The hydrostatic unit model consisted of several basic hydraulic subcomponents as shown in **Figure 5**. The Simscape library provided a realistic hydraulic element model, which allowed for a dependable result. The hydraulic pump and hydraulic motor models accounted for losses due to leakage flow and friction torque; the hydraulic pipes simulated the hydraulic hoses with resistance and elasticity; the fluid properties element evoked the physical characteristics of hydraulic oil such as density, viscosity, and compressibility; and the first order lag block provided the dynamic response of the servo valve of the hydraulic pump, which determines the transmission ratio of the hydraulic transmission unit.

The variation of load torque was the main source that caused the velocity change to the ICE due to the dynamic characteristics of the ICE. The controller was implemented to monitor the angular velocity of the ICE and regulate the transmission ratio of the hydrostatic unit to maintain a constant ICE velocity at its optimal working point. The whole system scheme is depicted in **Figure 6.**

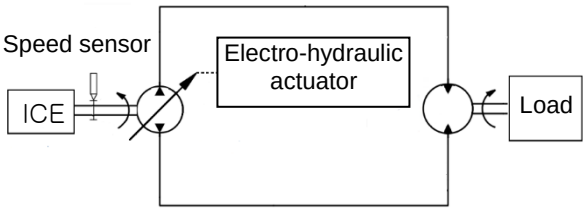

Figure 2. Principle structure of the hydrostatic drivetrain

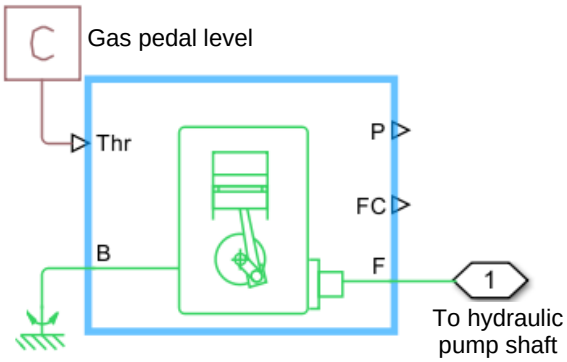

Figure 3. ICE model component

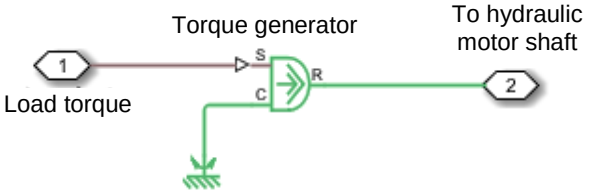

Figure 4. Load model component

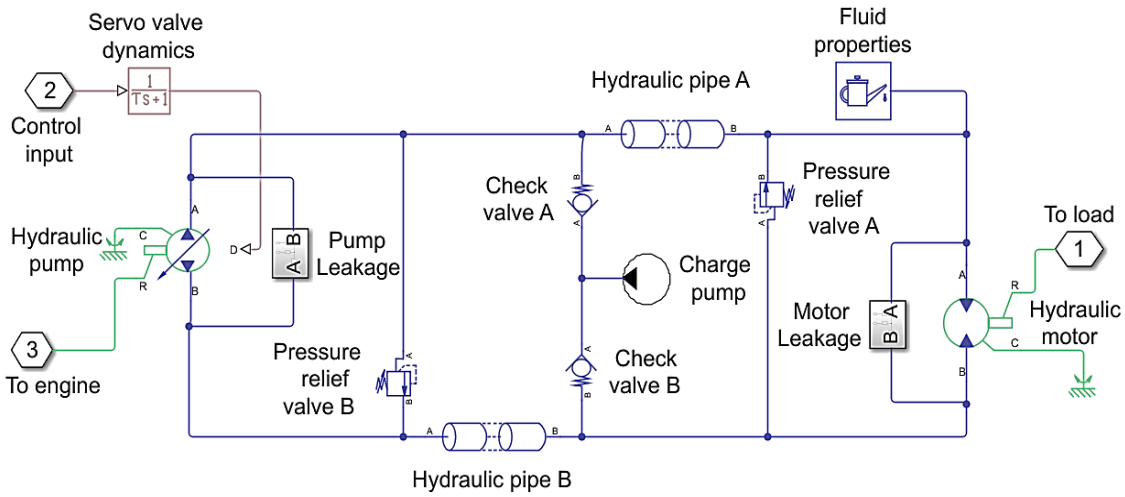

Figure 5. Hydrostatic unit model component

As demonstrated in **Figure 6**, the hydrostatic transmission unit was driven by the ICE and it generated a counterbalance to the load torque, τ_l. The controller monitored the angular velocity, ω_e, of the ICE and took the set point value of the ICE angular velocity, ω_d, as the input command. The controller produced a control signal, u, to regulate the transmission

ratio of the hydrostatic unit to keep the ICE velocity at its set point value despite changes to the load torque.

Results and Discussion

The PI control

Proportional-integral-derivative (PID) is the most widely used control strategy in industrial processes and it is still predominant in the control of hydrostatic transmissions. In time domain, the control law is stated as follows:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \quad (1)$$

where: $u(t)$ is the control signal generated by the PID controller; $e(t)$ stands for the control error, which is the difference between the output value of the controlled system and its desired (set point) value, $e(t) = \omega_e(t) - \omega_d$; and K_p, K_i, and K_d are the three gains of the proportional, integral, and derivative terms, respectively.

The proportional (P) term produces a control output signal that is proportional to the error of the system output response. When an equilibrium is reached, a steady-state error exists always. The integral (I) term accounts for the past value of the error. The term accumulates the error and produces a compensating signal to eliminate the steady-state error. The derivative (D) term provides an estimation of the future response of the system based on the current change rate of the output error; this term acts as a damping element in the control action.

In industrial implementations of the PID control, the D term is usually set to zero (K_d = 0), as seen in Knospe (2006), to avoid a noisy signal that is caused by the derivative of the measured output, and therefore, the popular form of the controller becomes PI control:

$$u(t) = K_p e(t) + K_i \int e(t)dt \quad (2)$$

In discrete-time form where the control strategy is implemented in a digital controller, the control law is established using rectangular quadrature numerical integration as follows:

$$u(n) = K_p e(n) + K_i T_s \sum_{k=0}^{n} e(k) \quad (3)$$

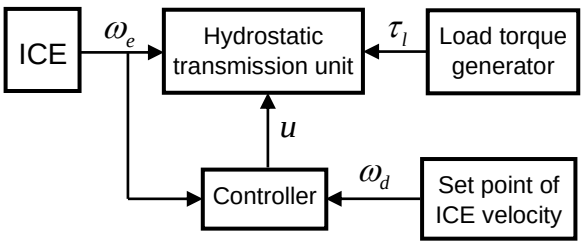

Figure 6. The block diagram of the control scheme

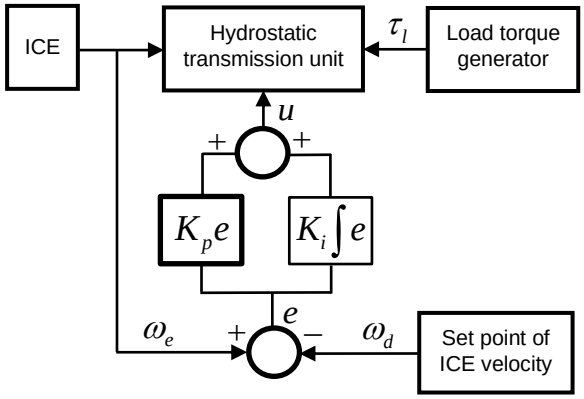

Figure 7. The implementation of PI control

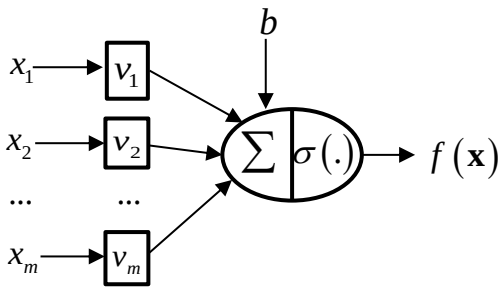

Figure 8. Principle structure of an artificial neuron

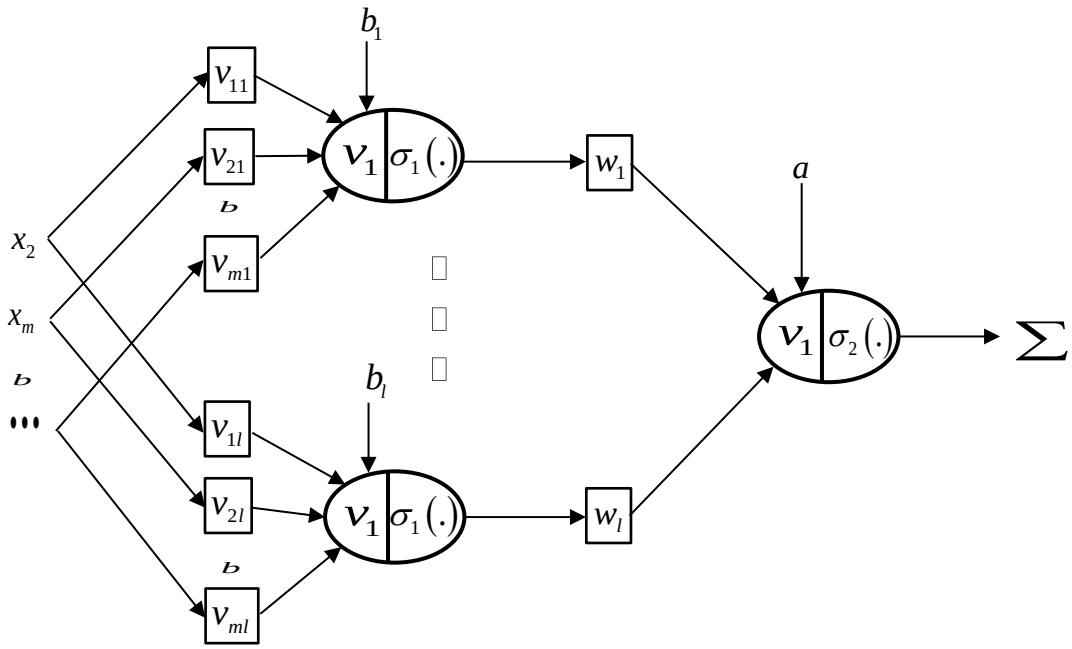

Figure 9. General structure of a neural network

Here, T_s presents the discrete time interval; n denotes the current time step index; and the index k expresses the discrete step k. In Matlab/Simulink simulations, the employment of PI control is easy by using the discrete-time integrator for the numerical integral term. The implemented PI control is shown in **Figure 7.**

The neural network-based control design

An artificial neuron is a function of the input, which is weighted by a factor referred to as the connection weight. A neuron generally consists of multiple inputs and weights, thus, the neuron function can be stated in vector form as follows:

$$f(\mathbf{x}) = \sigma\left(\mathbf{x}^T\mathbf{v} + b\right) \quad (4)$$

where $\mathbf{x}^T = [x_1, x_2,\ldots, x_m]$ presents the vector of m inputs; $\mathbf{v} = [v_1, v_2,\ldots, v_m]$ is the input weight vector; b denotes the bias of the neuron; and σ(.) is the activation function, of which, the sigmoid function is a popular form. The structure of a single neuron is demonstrated in **Figure 8**.

A neural network (NN), also called a multilayer perceptron, is a complex structure composed of several neurons in the form of neural layers, as depicted in **Figure 9**. As demonstrated, the neural network consists of two layers. The input layer has l neurons, each neuron takes m inputs as the input vector $\mathbf{x} = [x_1,\ldots, x_m]$ with its corresponding connection weight vectors $\mathbf{v_1} = [v_{11},\ldots, v_{m1}]$,…, $\mathbf{v}_l = [v_{1l},\ldots, v_{ml}]$, and they use the activation

function $\sigma_1(.)$ with the biases $b_1,\ldots, b_l$. The output layer is comprised of only one neuron corresponding to one output. It has weighting vector $w = [w_1,\ldots, w_l]$, the bias a, and uses the activation function $\sigma_1(.)$. Multilayer perceptron or neural networks have been applied successfully in the field of dynamic system identification and control. For a comprehensive overview of neural networks and their applications, see Hagan *et al.* (2002).

The output of a multilayer perceptron, as shown in Figure 9, is evaluated according to:

$$f(\mathbf{x}) = \sigma_2\left(a + \sum_1^l w_i.\sigma_1\left(\mathbf{x}^T.\mathbf{v}_i + b_i\right)\right) \quad (5)$$

In this study, a two-layer neural network was used to design a controller in a similar way proposed in the work of Danh & Aschemann (2021c) for the control task on a hydrostatic unit of a self-propelled vehicle. The designed control consisted of two components - a proportional (P) control and a feedforward compensator using a neural network, which produced the compensation signal that was trained by the current measurement error of the controlled output.

The P control

A P control was used to reduce the control error, which improved the learning performance of the neural network in the early training phase. The output signal from the P control was calculated according to:

$$u_P = k_P e \quad (6)$$

where the control error e is defined by $e = \omega_e - \omega_d$, and $k_P > 0$ is the positive control gain that is empirically tuned.

The neural network compensator

The control signal from the neural network compensator was evaluated according to:

$$u_{NN} = f(x) = a + \sum_1^l w_i.\sigma\left(x.v_i + b_i\right) \quad (7)$$

As implied in (7), the selected neural network consisted of two layers. The output layer had a single neuron with a linear activation function and connection weights $w_1,\ldots,w_l$. The input layer was composed of l neurons. They took $x = \omega_d$ as the single input with corresponding connection weights $v_1,\ldots, v_l$, and used the sigmoid activation function. The weights were trained by the control error e using the well-known back-propagation training technique. The implementation of the neural network-based control is shown in **Figure 10**.

Using the back-propagation training technique, the weights of the neural network were updated in such a way as to minimize the cost function of the control error:

$$E = \frac{1}{2}e^2 \quad (8)$$

and this led to a gradient descent updating law for each update step as follows:

$$\begin{aligned} v_{i,k+1} &= v_{i,k} - \eta\frac{\partial E}{\partial v_{i,k}}, \\ w_{i,k+1} &= w_{i,k} - \eta\frac{\partial E}{\partial w_{i,k}}, \\ a_{k+1} &= a_k - \eta\frac{\partial E}{\partial a_k}, \\ b_{i,k+1} &= b_{i,k} - \eta\frac{\partial E}{\partial b_{i,k}}. \end{aligned} \quad (9)$$

Here, η specifies the learning rate. The gradient of the cost function with respect to the network weights was evaluated using the chain rule:

$$\begin{aligned} \frac{\partial E}{\partial v_i} &= \frac{\partial E}{\partial e}\times\frac{\partial e}{\partial \omega_e}\times\frac{\partial \omega_e}{\partial u_{NN}}\times\frac{\partial u_{NN}}{\partial v_i}, \\ \frac{\partial E}{\partial w_i} &= \frac{\partial E}{\partial e}\times\frac{\partial e}{\partial \omega_e}\times\frac{\partial \omega_e}{\partial u_{NN}}\times\frac{\partial u_{NN}}{\partial w_i}. \end{aligned} \quad (10)$$

The gradients with respect to the biases a and b_i were applied similarly. According to mathematical definition, some terms in (10) can be determined as follows:

$$\begin{aligned} \frac{\partial E}{\partial e} &= e, \frac{\partial e}{\partial \omega_e} = 1, \\ \frac{\partial u_{NN}}{\partial w_i} &= \sigma\left(v_i.\omega_d + b_i\right).\sigma', \\ \frac{\partial u_{NN}}{\partial v_i} &= \omega_e.\sigma''.w_i.\sigma', \end{aligned} \quad (11)$$

where σ' and σ'' are the partial derivatives of the activation function σ, which are defined by:

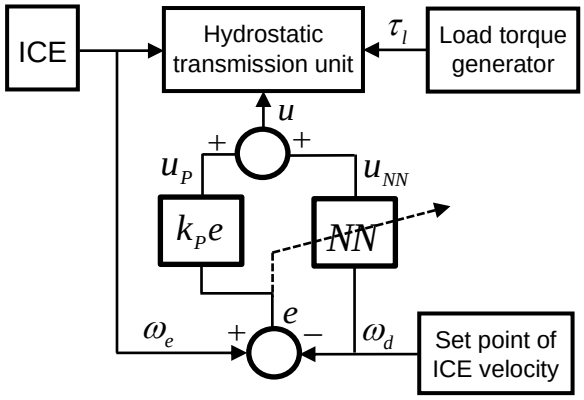

Figure 10. The implementation of the neural network-based control

$$\sigma' = \frac{\partial}{\partial v_i}\sigma\left(a + \sum_{i=1}^{l} w_i.\sigma\left(v_i.\omega_d + b_i\right)\right),$$
$$\sigma'' = \frac{\partial}{\partial w_i}\sigma\left(v_i.\omega_d + b_i\right). \quad (12)$$

The remaining term in (10), $\frac{\partial \omega_e}{\partial u_{NN}}$, is unknown. It can be, however, replaced by its sign without knowing the true value. The sign can be deduced from experimentation on the real system. With the given configuration of the hydrostatic transmission unit, when the transmission ratio increases, namely u_{NN} increases, the angular velocity of ICE reduces, implying:

$$\frac{\partial \omega_e}{\partial u_{NN}} = -1 . \quad (13)$$

Substituting (10), (11), and (13) into (9) results in the overall training rule as follows:

$$\begin{aligned} v_{i,k+1} &= v_{i,k} + \eta.e.\omega_e.\sigma''.w_{i,k}.\sigma' \\ w_{i,k+1} &= w_{i,k} + \eta.e.\sigma\left(v_{i,k}.\omega_d + b_{i,k}\right).\sigma' \\ b_{i,k+1} &= b_{i,k} + \eta.e.\sigma''.w_{i,k}.\sigma' \\ a_{k+1} &= a_k + \eta.e.\sigma' \end{aligned} \quad (14)$$

Numerical results

The proposed controllers were implemented in the Matlab/Simulink simulation environment according to the control schemes shown in **Figure 7** and **Figure 10**. The set point value of ICE was supplied to the controllers. The angular velocity of ICE was measurable and used as the feedback signal to calculate the control error, and measurement noise was also added for a realistic simulation result. The load torque generator produced a varying resistance torque applied to the output shaft of the hydrostatic transmission unit.

The parameters of both the PI and NN-based controllers were manually tuned for the best control results in the same simulation conditions. For the PI control, K_P = 40, K_i = 400. For the NN based control, k_P = 40, the number of neurons was l = 10, and the learning rate was η = 0.3. According to the ICE performance characteristics, the set point value was set to 2600 rpm at the optimal working point. The discrete time step was set to T_s = 1 ms.

In the first simulation test, the value of the load toque was varied from 135 Nm to 165 Nm in the form of a slow sinusoidal oscillation as shown in **Figure 11**. The comparison of the control results for both the PI and NN-based controls are shown in **Figure 12** and **Figure 13**.

As can be seen, the controlled output followed the set point very well with small errors for both controllers. The statistical data of the control performance is summarized in **Table 1**.

The statistics showed that, in the case of a slow load oscillation, both controls performed very well with small differences in terms of the control results. The NN-based control results were 11.2% smaller than the maximal control error and 14.8% smaller than the RMS error in comparison to the PI controller.

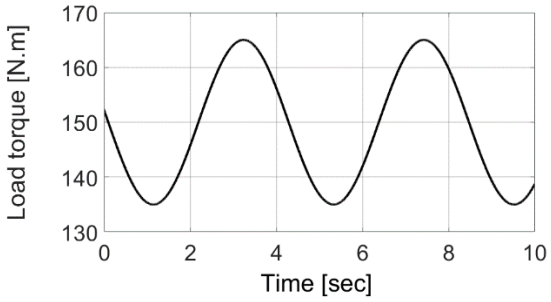

Figure 11. Slow varying load torque

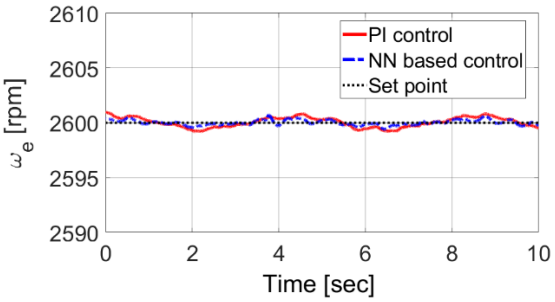

Figure 12. Control results comparison of the PI and NN-based controls for a slow varying load

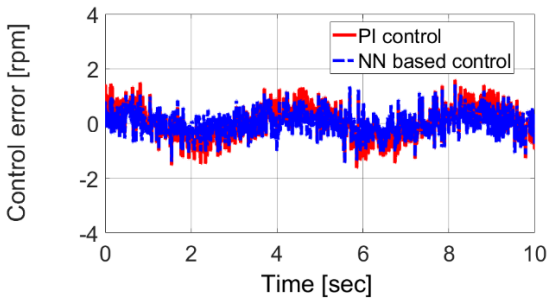

Figure 13. Control error comparison for a slow varying load

Table 1. Statistics comparison of control error for a slow varying load

	Max. Error [rpm]	RMS error [rpm]
PI control	1.60	0.54
NN control	1.42	0.46

The second test case was performed with a pulse load larger magnitude in comparison to the first case. Here, the value of the load torque varied from 120Nm to 180Nm in the form of a pulse signal as shown in **Figure 14.** The results of this test case are presented in **Figure 15** and **Figure 16**.

As presented in **Figure 15** and **Figure 16**, in the case of a pulse load, the NN-based controller performed better than the PI controller, with the difference being remarkable. The statistical data of control performance for a pulse load is summarized in **Table 2.**

The statistics show that, in the case of a large magnitude and fast varying load torque, both controls were still capable of keeping the set point quite well. The NN-based control, however, had superior results 21.6% smaller that the maximal control error and 27.4% smaller that the RMS error in comparison to the PI results.

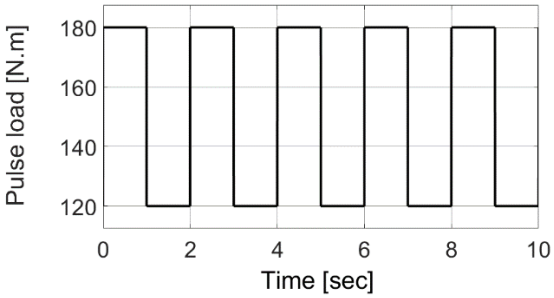

Figure 14. The pulse load torque

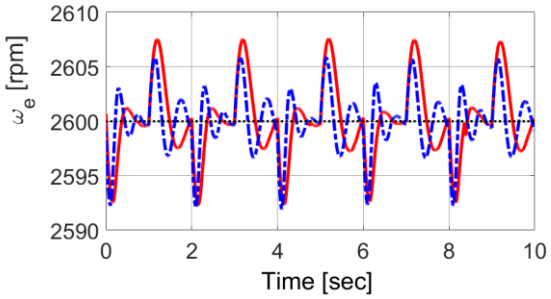

Figure 15. Control results of the PI and NN based controls for a pulse load

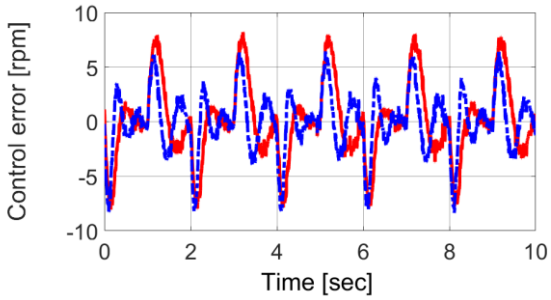

Figure 16. Control error comparison in the case of a pulse load

Table 2. Statistics comparison of control error for pulse load

	Max. Error [rpm]	RMS error [rpm]
PI control	8.20	3.50
NN control	6.43	2.54

Conclusions

PI is a well-known control approach in practice, while neural network-based control is an advanced method that provides accuracy and robustness for applications in uncertain dynamic systems. The simulation results in this study showed that the deployment of a neural network-based control for a hydro-mechanical transmission provided a robust and fast response control solution. The designed controller did not require a complete mathematical model of the hydro-mechanical transmission system, which allowed for simplicity of the design process and guaranteed robustness for the control structure. The designed neural network-based control promises a good result in practice.

Acknowledgments

This research was a part of the project “Investigation and design of hydro-mechanical

transmission system for research and practice development on self-propelled vehicle", coded T2023-04-06TĐ, funded by Vietnam National University of Agriculture.

References

Dang N. D. & Aschemann H. (2020). Neural network control by error-feedback learning for hydrostatic transmissions with disturbances and uncertainties. Proceedings of KKA 2020 - The 20th Polish Control Conference, Łódź, Poland, 2020. 436-448.

Danh D. N. & Aschemann H. (2021a). Tracking differentiator-based sliding mode velocity control of a hydrostatic transmission. Proceedings of 25th International Conference on Methods and Models in Automation and Robotics (MMAR). 269-274. DOI: 10.1109/MMAR49549.2021.9528444.

Danh D. N. & Aschemann H. (2021b). Design and Comparison of Active Disturbance Rejection Approaches for the Velocity Tracking Control of Hydrostatic Transmissions. Proceedings of European Control Conference (ECC). 2157-2162. DOI: 10.23919/ECC54610.2021.9655074.

Danh D. N. & Aschemann H. (2021c). Adaptive Feedforward Compensation Using a Neural Network for Velocity Control of Hydrostatic Transmissions. Proceedings of 29th Mediterranean Conference on Control and Automation (MED). 354-359. DOI: 10.1109/MED51440.2021.9480220.

Hagan M. T., Demuth H. B. & Jesús O.D. (2002). An introduction to the use of neural networks in control systems. International Journal of Robust and Nonlinear Control. 12(11): 959-985.

Knospe C. (2006). PID control. IEEE Control Systems. 26(1): 30-31.

Kwasniewski J., Pluta J. & Piotrowska A. (2003). Research on the properties of a hydrostatic transmission with different controllers. Acta Montanistica Slovaca. 8: 240-244.

Kwon H. & Ivantysynova M. (2020). System Characteristics Analysis for Energy Management of Power-Split Hydraulic Hybrids. Energies. 1837. doi:10.3390/en13071837.

Lennevi J. & Palmberg J. O. (1995). Application and implementation of LQ design method for the velocity control of hydrostatic transmissions. Proceedings of Institution of Mechanical Engineers. 209: 255-268. doi: 10.1243/PIME_PROC_1995_209_393_02.

Marcor A. & Rossetti A. (2011). Optimization of hydro-mechanical power split transmissions. Mechanism and Machine Theory. 46: 1901-1919. DOI: 10.1016/j.mechmachtheory.2011.07.007.

Marcor A., Benato A., Rossetti A., & Bettio Z. (2017). Study and simulation of a hydraulic hybrid powertrain. Energy procedia. 126: 1131-1138. doi: 10.1016/j.egypro.2017.08.279.

Nawrocka A. & Kwasniewski J. (2006). Advanced algorithm for stabilization rotational speed in hydrostatic transmission. International Journal of Mechanics. 25(4): 164-168.

Schulte H. & Gerland P. (2011). Control-oriented modelling of hydrostatic power-split CVTs using Takagi-Sugeno fuzzy models. Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology. DOI: 10.2991/eusflat.2011.18.

Wu H.W. & Lee C. B. (1996). Self-tuning adaptive speed control for hydrostatic transmission systems. International Journal of Computer Applications in Technology. 9: 18-33. doi: 10.1504/IJCAT.1996.062296.

Yu J., Cao Z., Cheng M. & Pan R. (2019). Hydro-mechanical power split transmissions: Progress evolution and future trends. Journal of Automobile Engineering. 233(3): 727-739. doi: 10.1177/0954407017749734.