

# Neural State Estimator for Complex Mechanical Part of Electrical Drive: Neural Network Size and Performance of State Estimation

Research Article

Dominik Łuczak, Adrian Wójcik

*Poznan University of Technology, Institute of Control, Robotics and Information Engineering,  
5 M. Skłodowska-Curie Square 60-965 Poznan, Poland  
Tel.: +48 61 665-2365  
Fax: +48 61 665-2563*

Received June 07, 2018; Accepted September 30, 2018

**Abstract:** This paper presents the results of simulation research of an off-line-trained, feedforward neural-network-based state estimator. The investigated system is the mechanical part of an electrical drive characterised by elastic coupling with a working machine, modelled as a dual-mass system. The aim of the research was to find a set of neural network structures giving useful and repeatable results of the estimation. The mechanical resonance frequency of the system has been adopted at the level of 9.3–10.3 Hz. The selected state variables of the mechanical system are load, speed and stiffness torque of the shaft.

**Keywords:** *two-mass drive • neural network • simulation studies • non-linear state estimation*

## 1. Introduction

Electrical drives represent a dominant source of mechanical power in various industrial applications. In order to meet the requirements for modern electrical drives in terms of both static and dynamic characteristics (Orłowska-Kowalska and Szabat, 2007b), it is vital to use control algorithms based on full knowledge about the process. Often, complete state vectors cannot be provided by the readily available measurement systems. Hence, there is a growing need for state variable-estimating algorithms.

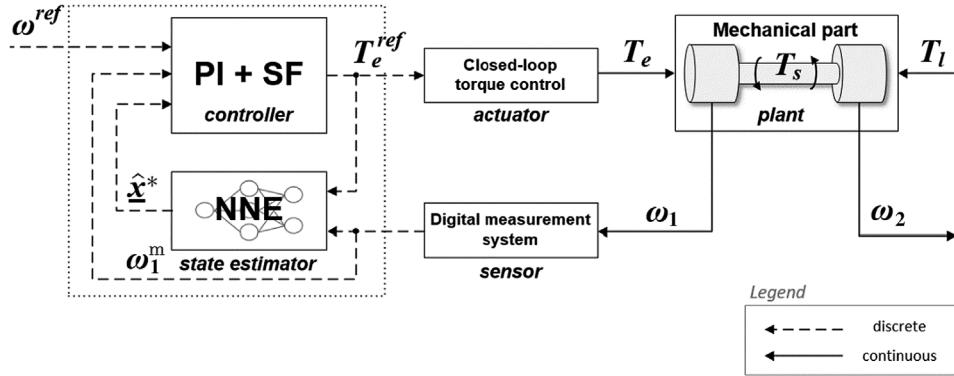
Machine learning methods – similar to artificial neural networks – play an increasingly important role in designing control algorithms. The following paper presents the state variables estimation algorithm based on a set of off-line-trained, feedforward, sigmoid neural networks (Kamiński, 2013; Orłowska-Kowalska and Kaminski, 2008; Orłowska-Kowalska and Kowalski, 1997; Yadaiah and Sowmya, 2006). The main advantages of this technique include the lack of necessity for developing a precise model of the considered system and the possibility of data generalisation during the training process.

One of the main problems related to the use of neural networks in control systems is the lack of a procedure for selecting the size and structure of a network (Kazmierkowski and Orłowska-Kowalska, 2002; Orłowska-Kowalska and Szabat, 2007a; Simoes and Bose, 1995). The paper presents statistical simulation research on estimation quality of the state variable of the mechanical part of a drive in relation to the size and structure of the neural network. On the basis of the obtained results, the scope of useful structures of neural state estimators (NSEs) for a complex mechanical part of drive is indicated.

\* Email: [Dominik.luczak@put.poznan.pl](mailto:Dominik.luczak@put.poznan.pl), [Adrian.wojcik@put.poznan.pl](mailto:Adrian.wojcik@put.poznan.pl)

## 2. Structure of the drive system

The schematic diagram of the considered drive system is presented in Fig. 1. The drive is a classic example of a cascade control system, with angular velocity control as the primary and motor torque control as the inner control loops. This section describes the mathematical models of the plant, the speed sensor, the speed controller and the actuator. The next section discusses in detail the structure of the proposed NSE.



**Fig. 1.** Schematic diagram of the considered drive system where  $\omega^{ref}$  [rad/s] – reference angular velocity,  $\omega_1$  [rad/s] – angular velocity on motor side,  $\omega_1^m$  [rad/s] – measured angular velocity (of motor),  $\omega_2$  [rad/s] – angular velocity on load side,  $T_e^{ref}$  [Nm] – reference motor torque,  $T_e$  [Nm] – motor torque,  $T_l$  [Nm] – load torque,  $T_s$  [Nm] – stiffness torque and  $\hat{x}^* = [\hat{\omega}_1, \hat{T}_s]^T$  – vector of unmeasurable state variable.

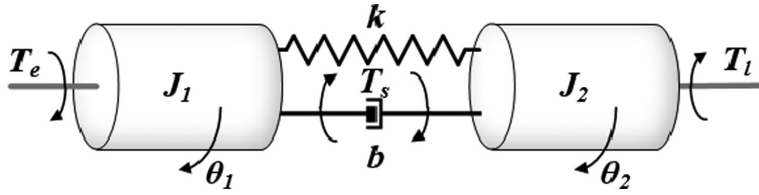
### 2.1. Mechanical Part of the Drive (Plant)

A conventional model of the complex mechanical system of the drive is the dual-mass system (Fig. 2), i.e. a system of two rigid bodies that share a common axis of rotation, each characterised by moments of inertia  $J_1$  and  $J_2$ . It is assumed that connections between the rigid bodies are characterised by stiffness  $k$  and coefficient of viscous friction  $b$  (Łuczak, 2014; Łuczak and Nowopolski, 2014; Orłowska-Kowalska and Szabat, 2007b). The equations of motion for the considered system, represented by Eq. (1), were derived from d'Alembert's principle (Łuczak, 2014).

$$\frac{d}{dt} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ T_s(t) \end{bmatrix} = \begin{bmatrix} -\frac{b}{J_1} & \frac{b}{J_1} & -\frac{1}{J_1} \\ \frac{b}{J_2} & -\frac{b}{J_2} & \frac{1}{J_2} \\ k & -k & 0 \end{bmatrix} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ T_s(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{J_1} & 0 \\ 0 & -\frac{1}{J_2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_e(t) \\ T_l(t) \end{bmatrix} \quad (1)$$

where  $J_1$  [kg·m<sup>2</sup>] – moment of inertia on motor side,  
 $J_2$  [kg·m<sup>2</sup>] – moment of inertia on load side,  
 $b$  [Nm·s/rad] – coefficient of viscous friction and  
 $k$  [Nm/rad] – stiffness.

The linear time-invariant (LTI) model of the mechanical part of the drive is based on the system of expressions in Eq. (1). Motor torque  $T_e$  and load torque  $T_l$  were adopted as the input signals and motor angular velocity  $\omega_1$  as the output, assuming that it is the only state variable used directly in the control process.



**Fig. 2.** Schematic diagram of the model of the mechanical part of the drive – a dual-mass system

## 2.2. Torque Control Loop (Actuator)

A closed-loop torque control system includes three elements of the electrical drive: an active current  $i_A$  digital controller, a power inverter and the electromagnetic part of the motor (Fig. 3). Optimal tuning of the current controller and proportional dependency between current  $i_A$  and motor torque  $T_e$  was assumed (Zawirski et al., 2012). Delay and time constants of the torque control loop for the considered system are at least two orders of magnitude smaller than the resonance period (period corresponding to the natural frequency) of the mechanical system. This allowed for usage of a simplified model as shown in Eq. (2).

$$G_T(s) = \frac{T_e(s)}{T_e^{ref}(s)} = K_e \frac{i_A(s)}{T_e^{ref}(s)} = \left( \frac{1 - e^{-\tau_s s}}{s} \right) \frac{1}{\tau_T s + 1} e^{-\tau_{rd} s} \quad (2)$$

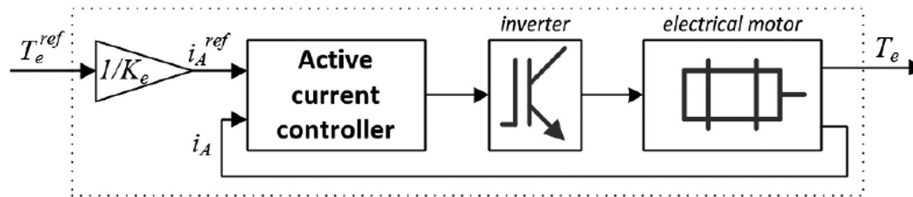
where  $\tau_T$  [s] – torque control loop time constant,

$\tau_{rd}$  [s] – torque control loop transport delay,

$\tau_s$  [s] – control system sample time and

$K_e$  [Nm/A] – motor torque constant.

This model abstracts from the particular type of electric motor. The available laboratory station is equipped with permanent magnet synchronous motors (PMSMs) – for this case, the active component of the current  $i_A$  is the current in the axis  $i_q$ .



**Fig. 3.** Schematic diagram of the torque control loop model

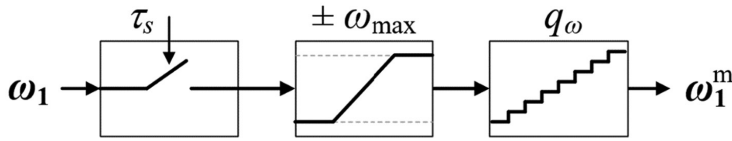
## 2.3. Digital Angular Velocity Measurement (Sensor)

The angular velocity measurement system model (Fig. 4) does not represent a specific sensor or technique, but instead, it takes into account the general parameters of measurement devices: bit resolution (quantisation), range (saturation) and sample time (discretisation), as shown in Eq. (3).

$$\omega_1^m(k) = q \cdot \left\lfloor \frac{1}{q} \text{sat}(\omega_1(t) \cdot \delta(t - k\tau_s), \omega_{\max}) + \frac{1}{2} \right\rfloor, \quad (3)$$

where  $q$  [rad/s] – sensor quantization step size and

$\omega_{\max}$  [rad/s] – sensor saturation.



**Fig. 4.** Schematic diagram of measurement system model

## 2.4. Digital Angular Velocity Control (Controller)

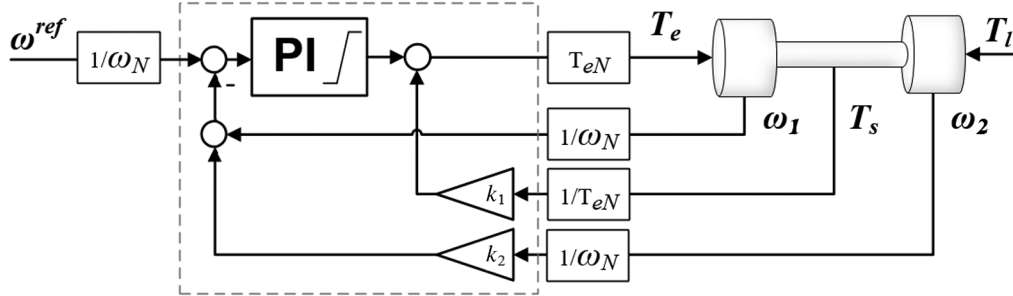
A discrete, parallel PI controller (proportional–integral controller), represented in Eq. (4), with an additional feedback (Fig. 5) was used in the angular velocity control method. Tuning was done based on a previous paper (Orlowska-Kowalska and Szabat, 2007b). The characteristic polynomial of the closed system is assumed to be in form. All input signals are normalised (divided by nominal values), and the output is denormalised (multiplied by nominal value). In Table 1, the controller gain for three feedback structures are presented: no additional feedback, stiffness torque feedback and load speed feedback. During simulation research, additional feedback was used alternatively in order to determinate estimation error in a closed loop for a single variable.

$$G_R(z) = K_P + K_I \cdot \frac{\tau_s}{z-1} \quad (4)$$

where  $K_p$  [-] – proportional gain and  $K_I$  [-] – integral gain.

$$\left(s^2 + 2\xi\omega_0 s + \omega_0^2\right)\left(s^2 + 2\xi\omega_0 s + \omega_0^2\right) = 0 \quad (5)$$

where  $\xi$  [-] – closed-loop system damping and  $\omega_0$  [rad/s] – closed-loop system natural radial frequency.



**Fig. 5.** Speed controller – PI controller with additional state feedback

where  $k_1$  [-] – stiffness torque feedback gain,

$k_2$  [-] – load speed feedback gain,

$\omega_N$  [rad/s] – nominal angular velocity of the drive and

$T_{eN}$  [Nm] – nominal torque of the drive.

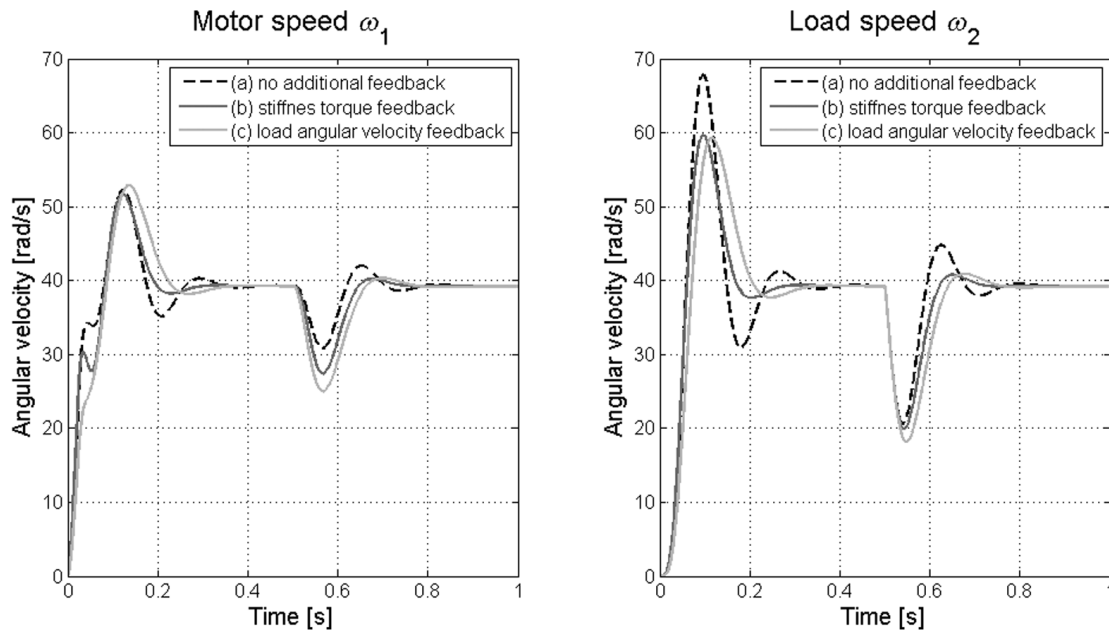
After including the additional state feedback in the closed-loop control system, the system damping  $\xi$  can be selected during the tuning procedure as  $\xi^{ref}$ . In Fig 6, an example step response of the drive system with and without additional state feedback and  $\xi^{ref} = 0.7$  is presented. Introduction of both additional state feedbacks reduces load speed overshoot.

The adopted model allowed for the inclusion of phenomena vital from the point of view of training procedures for artificial neural networks, such as measurement and control delays, quantisation noise and sampling period. At the same time, in order to simplify the calculations, a simple model of the electromagnetic part of motor and power inverter was adopted.

**Table 1.** Speed controller gains, closed-loop system damping and natural radial frequency

	No additional feedback	Stiffness torque feedback	Load speed feedback
$K_p$	$K_p = 2\sqrt{\frac{\tau_1}{\tau_c}}$	$K_p = 2\sqrt{\frac{\tau_1(1+k_1)}{\tau_c}}$	$K_p = 4\xi^{ref}\tau_1\sqrt{\frac{1+k_2}{\tau_2\tau_c}}$
$K_I$	$K_I = \frac{\tau_1}{\tau_2\tau_c}$	$K_I = \frac{\tau_1}{\tau_2\tau_c}$	$K_I = \frac{\tau_1(1+k_2)}{\tau_2\tau_c}$
$k_1$	0	$k_1 = \frac{4\left(\xi^{ref}\right)^2\tau_1}{\tau_2} - 1$	0
$k_2$	0	0	$k_2 = \frac{\tau_1 + \tau_2}{\tau_1\left(4\left(\xi^{ref}\right)^2 + 1\right)} - 1$
$\xi$	$\xi = \frac{1}{2}\sqrt{\frac{\tau_2}{\tau_1}}$	$\xi^{ref}$	$\xi^{ref}$
$\omega_0$	$\omega_0 = \sqrt{\frac{1}{\tau_2\tau_c}}$	$\omega_0 = \sqrt{\frac{1}{\tau_2\tau_c}}$	$\omega_0 = \sqrt{\frac{1+k_2}{\tau_2\tau_c}}$

where  $\tau_1 = (\omega_N/T_{eN})J_1$  [s] – motor mechanical time constant,  
 $\tau_2 = (\omega_N/T_{eN})J_2$  [s] – load mechanical time constant and  
 $\tau_c = (T_{eN}/\omega_N)k$  [s] – connection stiffness time constant.


**Fig. 6.** Comparison of step response of drive system with and without state feedback (ideal state estimation): motor angular velocity (left) and load angular velocity (right)

### 3. Proposed NSE

The proposed NSE is a variation of the neural network autoregressive with exogenous input model (NNARX), represented by Eq. (6) (Orlowska-Kowalska and Kowalski, 1997; Orlowska-Kowalska and Szabat, 2007a; Yadaiah and Sowmya, 2006). The task of the NSE system is to predict the value of an unmeasured state variable of the plant, represented by Eq. (7).

$$y_{k+1} = NNARX([y_k \ \dots \ y_{k-d1} \ u_k \ \dots \ u_{k-d2}]) \quad (6)$$

$$\hat{x}_{k+1} = NSE([y_k \ \dots \ y_{k-d1} \ u_k \ \dots \ u_{k-d2}]) \quad (7)$$

where  $y$  [-] – output signal of plant,

$u$  [-] – control signal (input signal of plant) and

$\hat{x}$  [-] – estimation of selected state variable of plant.

#### 3.1. Structure of the NSE

The NSE consists of three primary components (Fig. 7): data pre-processing, a set of neural algorithms (Fig. 8) and data post-processing. Estimator inputs are the current plant output and control signal, namely, – motor speed  $\omega$ , and reference motor torque  $T_e^{ref}$ , respectively. Estimator output is the prediction of the unmeasurable state variables of the plant, namely, load speed  $\omega_2$  and stiffness torque  $T_s$ .

The presented method assumes the use of a pre-processing procedure common for each of the estimated state variables. The pre-processing procedure consists of the following steps:

1. Input data normalisation to  $\langle -1, 1 \rangle$  range; requires assumption of maximum values of angular velocities and torques occurring in the system.
2. Obtaining the data vector, containing samples of the reference and measurement signals. The number of used samples and the sampling method (varying/constant sample time) constitute the discussed method's degrees of freedom.
3. Low-pass filtering of the obtained signals with digital, second-order filters (optional).

The vector obtained in the pre-processing step is then passed to the set of artificial neural networks. Each one of the networks is optimised through the supervised off-line learning process for estimating a single state variable, i.e. each one of the variables is estimated independently. This, in turn, allows for choosing a different network structure

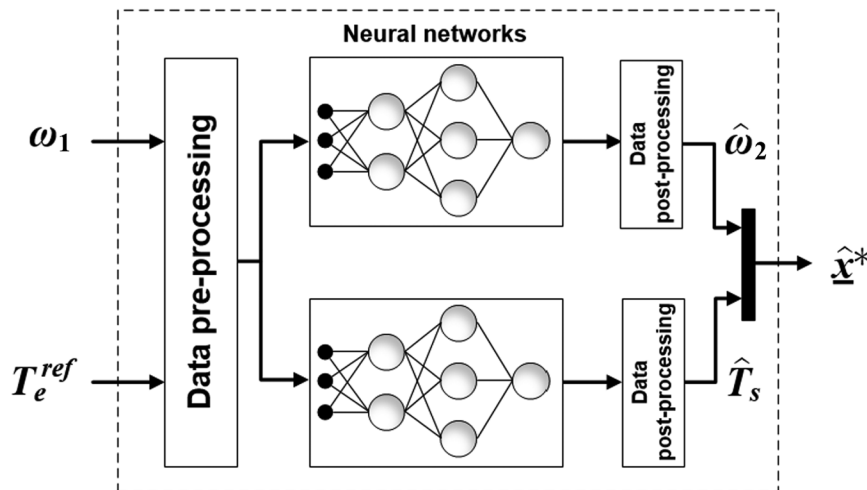
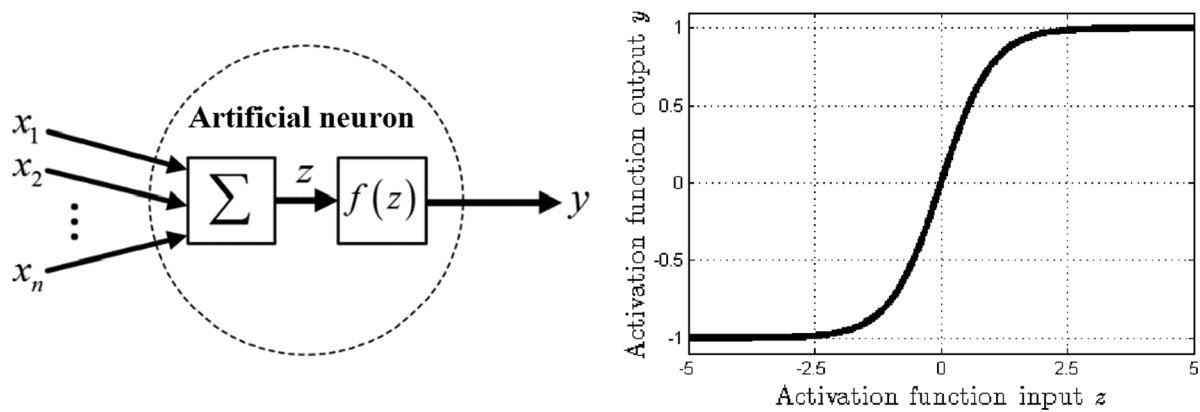


Fig. 7. Schematic diagram of NSE for selected variables of dual-mass system



**Fig. 8.** Schematic diagram of sigmoid neuron (right) and activation function (left)

for each of the state variables. The simulation research was primarily designed to identify a set of neural structures useful for the considered task.

The procedure of the post-processing of the output data consists of denormalisation of the estimation results and optional low-pass filtering, respectively, for load angular speed and stiffness torque. An important detail of the implementation of the algorithm is the imposition of saturation on the output of the estimator. The restriction, however, was imposed in the neural algorithm itself (non-linear output activation function).

### 3.2. Training Procedure of the NSE

The procedure of training of neural networks constituting the NSE consists of the following steps:

1. Generating the input signal vectors of the reference angular velocity  $\omega^{ref}$  and load torque  $T_l$ . This task requires a preliminary knowledge of the drive system, as proper training results will only be obtained under the condition that the spectrum of input signals will cover a wide range of bandwidth of the system. Particular attention should be paid to excite the resonance and anti-resonance frequencies of the mechanical part of the drive.
2. Obtaining the input vectors of the neural networks. This step requires the signals of the angular velocity of the motor and the given electromagnetic torque. This can be achieved by using either the simulation model described earlier or the drive system that satisfies the above-mentioned assumptions. Then, the signals must be pre-processed as describe above.
3. Obtaining the output vectors (targets). The algorithm assumes the availability of dataset of the unknown variables of state. This data can be obtained in a simple way by using a simulation model or the drive system, which – at least in certain circumstances – allows for measurement of the velocity and the stiffness torques of each of the shaft inertias. The second solution is achievable under laboratory conditions; hence, the procedure of training of the estimator is reduced to calibrating the controller with the use of additional sensors.
4. Distribution of training data into two groups: actual training data and validation data. Due to the dynamic nature of the data and the static nature of the estimator, it was decided to distribute the data into continuous blocks, which is assumed to improve the level of data generalisation.
5. Carrying out the training process using the chosen optimisation algorithm. Mean squared normalised error function was used as the performance indicator. The Levenberg–Marquardt backpropagation algorithm was adopted as the primary method of optimisation.

Weights update for the selected method is expressed as in Eq. (8). The computation process for Jacobian matrix can be organised according to the traditional backpropagation computation in first-order algorithms (such as the error backpropagation [BP] algorithm), with minor modification. First of all, for every pattern, in the BP algorithm, only one backpropagation process is needed, while in the Levenberg–Marquardt algorithm, the backpropagation process has to be repeated for every output separately in order to obtain consecutive rows of the Jacobian matrix (Yu and Wilamowski, 2011).

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \left( \mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I} \right)^{-1} \mathbf{J}_k \mathbf{e}_k \quad (8)$$

where  $\mathbf{w}_k$  [-] – network weights and biases vector in the  $k$ -th iteration,  
 $\mathbf{e}_k$  [-] – network error vector (for all neurons) in the  $k$ -th iteration,  
 $\mathbf{J}_k$  [-] – network Jacobian (partial derivative of error vector with respect to network weights) in the  $k$ -th iteration,  
 $\mu_k$  [-] – combination coefficient in the  $k$ -th iteration (always positive) and  
 $\mathbf{I}$  [-] – identity matrix.

This method is a modification of the Gauss–Newton method, with estimation of the Hessian matrix, as in Eq. (9), and the error gradient, as in Eq. (10).

$$\mathbf{H} \approx \mathbf{J}^T \mathbf{J} + \mu \mathbf{I} \quad (9)$$

$$\mathbf{g} = \mathbf{J} \mathbf{e} \quad (10)$$

where  $\mathbf{H}$  [-] – network Hessian (second-order partial derivative of error vector with respect to network weights) and  
 $\mathbf{g}$  [-] – network error gradient.

As the combination of the steepest descent algorithm and the Gauss–Newton algorithm, the Levenberg–Marquardt algorithm switches between the two algorithms during the training process. When the combination coefficient  $\mu$  is very small (nearly zero), the Gauss–Newton algorithm is used. When the combination coefficient  $\mu$  is very large, the steepest descent method is used (Yu and Wilamowski, 2011).

6. Test of performance of the NSE on a different set of input signals. Test is a two-step process, the first being evaluation of the estimation error exclusively, without including the estimated variables in the regulation. The second step is to evaluate the estimation error and control quality after closing the estimator feedback.

After reaching satisfactory results at the stages of training and testing, the adopted set of data and network structure can be considered useful for the investigated drive system.

## 4. Simulation results

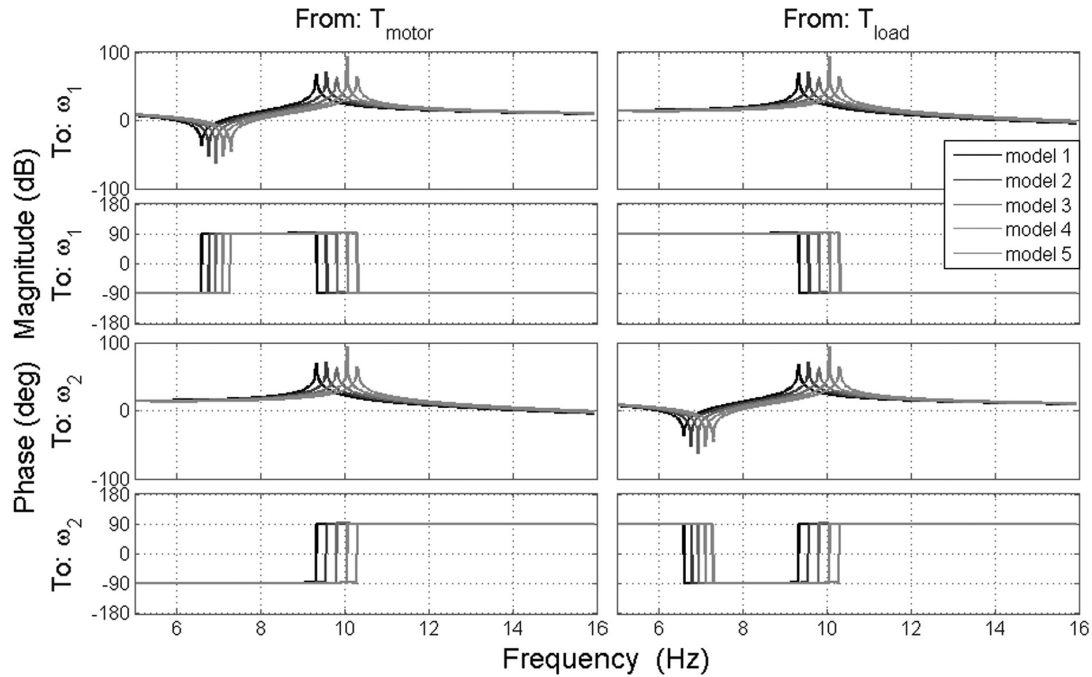
The conducted simulation research consisted of optimisation and evaluation of 272 neural network structures as NSE for the dual-mass system. Training and testing were led for the dual-mass system as a model of the mechanical part, with five different values of shaft stiffness (Table 2). Sample time was chosen as  $\tau_s = 500 \mu\text{s}$ . Speed sensor bit resolution was chosen as 16 bits with range of  $2\omega_N$ . In Fig. 9, the Bode plot of the considered system models is shown.

**Table 2.** Drive model parameters

Moment of inertia on motor side	Moment of inertia on load side	Ratio of the moments of inertia	Coefficient of viscous friction	Nominal motor speed	Nominal motor torque
$J_1$	$J_2$	$R = J_2/J_1$	$b$	$n_N$	$T_N$
[kg·m <sup>2</sup> ]	[kg·m <sup>2</sup> ]	[-]	[Nm·s/rad]	[rpm]	[Nm]
0.0041	0.0041	1.0	0.0	1,500.0	3.1831
Stiffness [Nm/rad]					
model 1	model 2	model 3	model 4	model 5	
7.0340	7.4091	7.7939	8.1885	8.592	

Neural networks with fixed-size inputs and outputs were subjected to the study. As the input vector, due to the order of the investigated system, based on a previous study (Orlowska-Kowalska and Kaminski, 2008), current samples and two previous samples of measured speed ( $d_1 = 2$ ) and reference torque ( $d_2 = 2$ ) were adopted – a total of six





**Fig. 9.** Bode plots of plant models – resonant frequencies in the range of 9.3–10.3 Hz

elements. Thus, 16 networks with a single hidden layer and 256 networks with two hidden layers were examined. Training data for all cases were identical and consisted of 500,000 samples. However, to prevent overfitting for small neural networks and ensure good generalisation for bigger networks, a simple heuristic, shown in Eq. (11), was adopted for selection of the training data from a common set.

$$(N_{wb})^2 \cong p \quad (11)$$

where  $N_{wb}$  [-] – number of weights and biases in network and  $p$  [-] – number of learning samples.

The data was obtained by conducting a simulation in the MatLab/Simulink environment. A pseudorandom binary sequence (PRBS) signal within the range of nominal values and the sample time  $\tau_{PRBS} = 100 \cdot \tau_s$  were used as the procedure inputs (reference speed  $\omega^{ref}$  and load torque  $T_l$ ). The results of the training and validation for each structure were satisfactory –  $<0.5\% \omega_N$  for the speed estimator and  $0.05\% T_N$  for the torque estimator. This allows to state that the training data have been selected correctly. Evaluation of the usefulness of the structure was based on the quality of state estimation for a separate set of data, which was carried out in a system with feedback from the state estimate. The collected results, averaged for five sets of model parameters (Table 2), are shown in Figs. 10 and 11. Root mean square error (RMSE) of the estimate of the state variable, expressed in physical units, was selected as the indicator.

Test results with error  $>0.5\%$  of nominal value are marked black and are not presented by the exact value but as constant, denoting unsatisfactory error – 1.0 for load speed and 0.05 for stiffness torque. Test results with error  $<120\%$  of global minimum are marked white. NSEs with a single hidden layer return the best results of load speed estimation for seven and eight neurons. The best result of stiffness torque estimations is obtained for a network with 7 and 11 neurons. NSEs with two hidden layers return the best result of load speed estimation for the {7,5} and {2,7} structures. The general relation between neuron number and estimation error suggests that for this particular task, layers with  $>12$  neurons tend to overfit the training data. The best result of stiffness torque estimation is obtained for a network with {9,6} and {16,3} neuron structure. Most of the test failures occur for networks with  $<4$  neurons in the hidden layer.

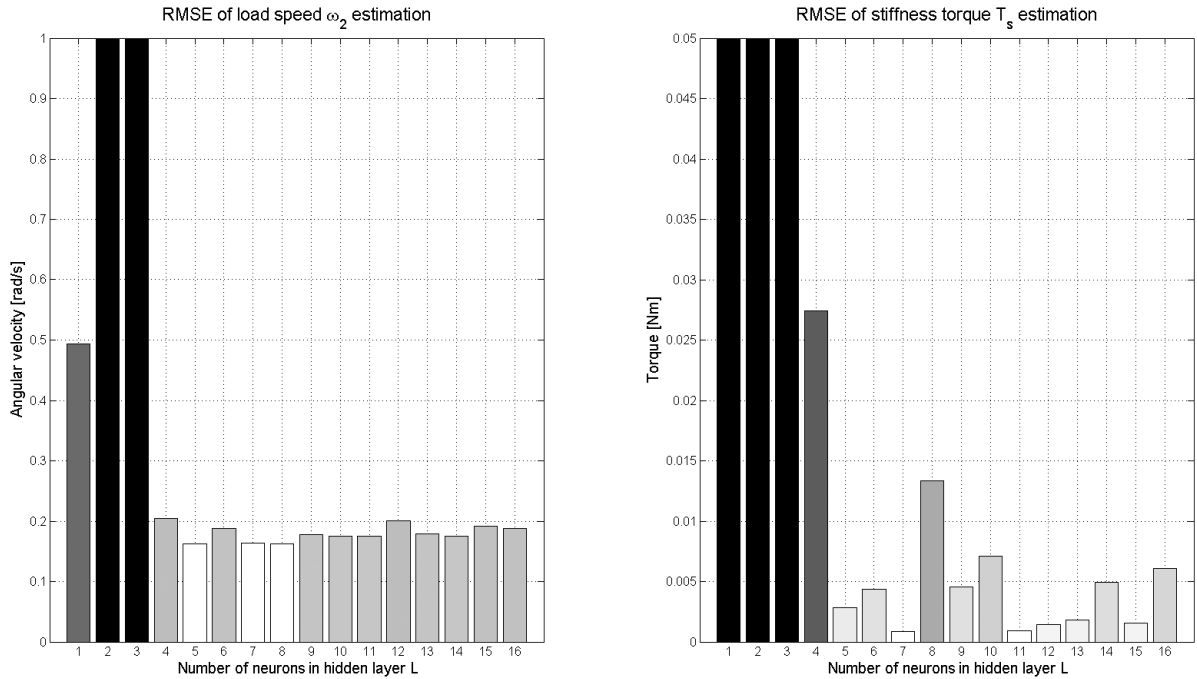


Fig. 10. Closed-loop test results for single hidden layer NSE: RMSE of estimation of selected state variables of dual-mass system

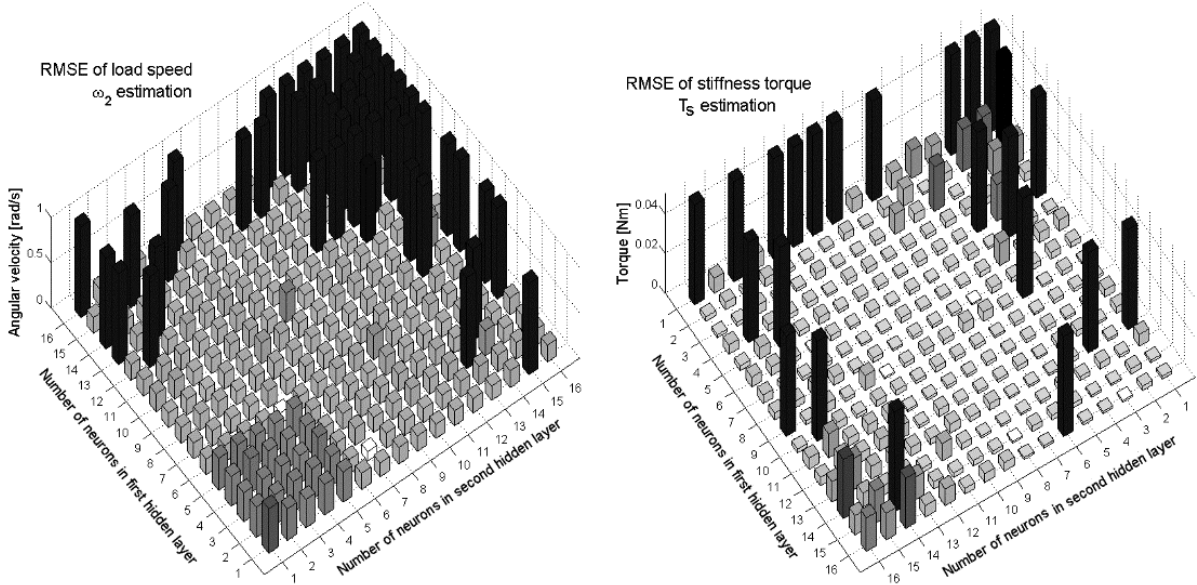


Fig. 11. Closed-loop test results for NSE with two hidden layers: RMSE of estimation of selected state variables of dual-mass system

Simulation research was originally conducted in the MatLab/Simulink environment, allowing the evaluation of the accuracy of both the training procedure and the estimation algorithm. In addition, tests were repeated in the CrossCore Embedded Studio environment, using previously obtained data, in order to verify the correctness of the authorial method implementation. The programme code was written in C++, using the object-oriented technique (Łuczak and Wójcik, 2017).

## 5. Conclusion

Based on the obtained results, it can be concluded that the proposed training procedure for NSE was correct. All the selected networks produced satisfactory estimation results for the training data and the open-loop validation test; therefore, the range of the tested structure of neural networks was also selected correctly. The level of estimation error in the best cases shows that the algorithm is applicable in the case of the considered drive system. The proposed training method gives useful and repeatable results. The results demonstrate that neural networks with fewer neurons – less than 12 per hidden layer – deal with the task of load speed of dual-mass system much better than networks with greater numbers of neurons. Even a single, well-trained neuron with appropriate input size is a useful model, allowing to reduce drive speed overshoot. Estimation of the stiffness torque requires a neural network with neuron number  $>6$  neurons per hidden layer. Further work is oriented towards determining the upper limit of the resonance frequency of the mechanical drive, for which static neural models allow for state estimation.

## References

- Isermann, R. (1989). State controller and state observer. In: B.W. Dickinson, E.D. Sontag, eds., *Digital Control Systems: Volume 1: Fundamentals, Deterministic Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 180–227.
- Jobling, C. P., Grant, P. W., Barker, H. A. and Townsend, P. (1994). Object-oriented programming in control system design: a survey. *Automatica*, 30(8), pp. 1221–1261.
- Kamiński, M. (2013). Estymacja zmiennych stanu układu dwumasowego za pomocą modeli neuronowych. *Prace Naukowe Instytutu Maszyn, Napędów i Pomiarów Elektrycznych Politechniki Wrocławskiej*, 69(33), pp. 222–238.
- Kazmierkowski, M. P. and Orłowska-Kowalska, T. (2002). NN state estimation and control in converter-fed induction motor drives, Ch. 2. In: Seppo J. Ovaska, Les M. Sztandera, eds. *Soft Computing in Industrial Electronics*. Heidelberg, Germany: Physica-Verlag, pp. 45–94.
- Kincaid, D. and Cheney, W. (2006). Arytmetyka Komputerowa. In: Stefan Paszkowski, ed., *Analiza numeryczna*, Warszawa: Wydawnictwo Naukowo-Techniczne, pp. 46–54.
- Łuczak, D. (2014). Mathematical model of multi-mass electric drive system with flexible connection. In: *2014 19th International Conference on Methods and Models in Automation and Robotics MMAR*, Miedzyzdroje, Poland, 2–5 September 2014, IEEE, pp. 590–595.
- Łuczak, D. and Nowopolski, K. (2014). Identification of multi-mass mechanical systems in electrical drives. In: *2014 16th International Conference on Mechatronics - Mechatronika*, Brno, Czech Republic, 3–5 December 2014, IEEE, pp. 275–282.
- Łuczak, D. and Wójcik, A. (2017). Object-oriented DSP implementation of neural state estimator for electrical drive with elastic coupling. *Poznan University of Technology Academic Journals: Electrical Engineering*, 91, pp. 395–406.
- Orłowska-Kowalska, T. and Kaminski, M. (2008). Optimization of neural state estimators of the two-mass system using OBD method. In: *IEEE International Symposium on Industrial Electronics*, Cambridge, UK, 30 June–2 July 2008, ISIE, pp. 461–466.
- Orłowska-Kowalska, T. and Kowalski, C. T. (1997). Neural network application for flux and speed estimation in the sensorless induction motor drive. In: *Proceedings of the IEEE International Symposium on Industrial Electronics*, vol. 3, Guimaraes, Portugal, 7-11 July 1997, Portugal: ISIE, pp. 1253–1258.
- Orłowska-Kowalska, T. and Szabat, K. (2007a). Neural-Network Application for Mechanical Variables Estimation of a Two-Mass Drive System. *IEEE Transactions on Industrial Electronics*, 54(3), pp. 1352–1364.
- Orłowska-Kowalska, T. and Szabat, K. (2007b). Vibration Suppression in a Two-Mass Drive System Using PI Speed Controller and Additional Feedbacks -Comparative Study. *IEEE Transactions on Industrial Electronics*, 54(2,3), pp. 1193–1206.
- Reynaldi, A., Lukas, S. and Margaretha, H. (2012). Backpropagation and Levenberg-Marquardt algorithm for training finite element neural network. In: *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation*, Valetta, Malta, 14–16 November 2012, EMS, pp. 89–94.
- Rojas, R. (1996). *Neural Networks – A Systematic Introduction*, [Online]. Available at: <http://page>.

- mi.fu-berlin.de/rojas/neural/ [Accessed 19 Feb. 2016].
- Simoes, M. G. and Bose, B. K. (1995). Neural network based estimation of feedback signals for a vector controlled induction motor drive. *IEEE Transactions on Industry Applications*, 31(3), pp. 620–629.
- Szabat, K. and Orłowska-Kowalska, T. (2008). Optimization of the two-mass drive dynamics using different compensation feedbacks. In: *11th International Conference on Optimization of Electrical and Electronic Equipment, OPTIM*, Brasov, Romania, 22–24 May 2008, IEEE, pp. 19–24.
- Yadaiah, N. and Sowmya, G. (2006). Neural network based state estimation of dynamical systems. In: *International Joint Conference on Neural Networks, IJCNN*, Vancouver, BC, Canada, 16–21 July 2006, IEEE, pp. 1042–1049.
- Yu, H. and Wilamowski, B. M. (2011). Levenberg-Marquardt training. In: Bogdan M. Wilamowski, J. David Irwin, eds., *Industrial Electronics Handbook – vol. 5 Intelligent Systems*, 2nd ed. pp. 12–1 to 12–15.
- Zawirski, K., Deskur, J. and Kaczmarek, T. (2012). *Automatyka Napędu Elektrycznego*. Poznań: Wydawnictwo Politechniki Poznańskiej, pp. 76–100.