2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Neural Network Approach to Hoist Deceleration Control

Peter Beneš, Ivo Bukovsky

Abstract— **This paper introduces a neural network approach to hoist deceleration control of industrial hoist mechanisms, with particular focus to crane applications. The necessity for investigation in this field arises from the increasing demands in terms of safety within in the industry. Should the industrial hoist feature too high deceleration this can lead to overstressing of the hoist mechanism and structure, further, damaging of the load due to large dynamical forces. Furthermore, too low deceleration can lead to incompliance with industrial standards and thus being a safety issue, due to potential loss of load in the worst case. Till this day various solutions and devices have been proposed to achieve controlled deceleration of the industrial hoist braking. However, there still lies a necessity for deeper study into this problem, to achieve quicker response towards the desired behavior of the hoist deceleration as well as improved adherence with the desired behavior. Thus, this paper analyses the potentials of hoist deceleration control by neural network architectures as such the linear, quadratic and cubic neural units with real-time recurrent learning and back-propagation through time approach when real measured data are used for experimental analysis.**

I. INTRODUCTION

Till now hoist deceleration control is not such a widely studied area. Several theoretical studies have been published, which are focused primarily on mining hoist applications, as exampled in the recent works of [2] & [3]. Though featuring differences in their dynamics, due to vast hoisting distances and conditions, as compared with industrial crane applications. The principle structure of the studied hoist mechanisms are analogical, structured as single-input-single-output (SISO) systems, with the hoist motor speed or deceleration being the output of the system and frequency or voltage variation in the sense of AC induction hoist motors, as input. Though these reviewed studies feature promising results, they are rather more theoretically focused. Thus, a necessity into experimentation with real industrial hoist mechanisms are a key step for further research into this problem, along with the necessity for further optimization of control, to achieve quicker response towards the desired behavior of the hoist deceleration as well as improved adherence with the desired behavior. An earlier study from 2003 based on higher order neural units for adaptive identification and fast state feedback control, of unknown non-linear systems [4], shows that the use of higher order neural units in state feedback control, can achieve even faster response for unknown non-linear systems, like the hoist mechanism in [2]. Thus this paper aims to provide a new study into the potentials of adaptive control via neural network architectures as such the linear, quadratic and cubic neural units with real-time-recurrent-learning (RTRL) and variation of back-propagation-through-time (BPTT). This investigation will be undertaken via application of the adaptive identification and control software application as presented in the work [5]. Investigation via means of an adaptive neural network based approach, arises from the already promising theoretical studies of higher-order neural units (HONUs), especially the quadratic neural unit for engineering problems [4][7][8]. These studies are focused on the use of supervised learning based approaches for polynomial structured neural units, also known as a class of HONUs, for adaptive identification and control of real engineering systems. We may recall from the work [6], successful implementation of a quadratic neural unit controller (neuro-controller), used for control of a bathyscaphe system located in the automatic control laboratories of FME at CTU. Where, here, such controller adhered more closely to the desired behavior of the system than the conventionally used PID controller. An extension on this result may also be recalled in the work [5]. Where, further study was made, via introduction of a new software application for adaptive identification and control, along with further testing on both a theoretical system and the previously mentioned bathyscaphe system. Given these results of real implementation, we thus aim to investigate via a similar approach the use of such adaptive linear, quadratic and cubic neural unit based architectures for control of the hoist deceleration problem. The results presented in this paper, are based on measured data from a smaller scale industrial hoist mechanism which is that of the Gude GSZ 200. Due to the Gude GSZ 200/400 being a mechanically braked hoist, the only option for an electrical means of control of this hoist is via an extended single phase variable frequency drive. Thus, this investigated hoist mechanism is set up with variable frequency as its input, corresponding to a produced deceleration characteristic at the output. This study highlights that although the initial characteristic for a standard braking process may be quite undesirable, the investigated form of adaptive control is still able to achieve desirable control of the hoist deceleration as is later shown in Figure 5 & Figure 6 of this paper.

II. PRELIMINARIES

A. Industrial Problem Description

Within the industry, industrial hoist mechanisms must comply with a prescribed minimum and maximum braking torque. This braking torque is directly influenced by the inertias of the hoist mechanism and more importantly their

Ivo Bukovsky and Peter Beneš are with Czech Technical University in Prague, Department of Instrumentation and Control Engineering, Technicka 4, 166 07, Prague, Czech Republic, email: {ivo.bukovsky;petermark.benes}@fs.cvut.cz. This work has been supported by the grant SGS12/177/OHK2/3T/12.

deceleration characteristic. Thus, the braking torque is in fact the torque or rotational force required to suspend the load of the hoist in the air, should the braking mechanism be disregarded i.e.; the torque necessary to substitute for the braking mechanism, in order to keep the hoist load suspended.

The true braking torque of the industrial hoist mechanism, is given via the following relation, where here we see its direct relation to the hoist deceleration, as follows;

$$B_T + \zeta_{mech.} = (\sum J^{eq.}_{rot.} + \sum J^{eq.}_{lin.}).\varepsilon\,, \qquad (1)$$

Here B_T [N.m] denotes the real braking torque, where $\zeta_{mech.}$ represents the total mechanical torque such to add to the braking torque of the whole hoist mechanism and ε, representing the angular acceleration [rad/s^2]. Furthermore, $\sum J^{eq.}_{rot.}$ & $\sum J^{eq.}_{lin.}$ correspond to the equivalent rotational and linear moving inertias, respectively.

According to industrial crane hoist standards, as such that of the AS1418.1 -2002 Clause 7.12.8 [1]. Braking systems shall comply with a minimum braking torque of 1.6 times the rated capacity in a static condition and arrest a minimum of 1.2 times the rated hoist capacity, from the maximum lowering speed, in a dynamical sense. Should the hoist mechanism suspend less than 1.6 times the rated capacity of the hoist mechanism, the load would be at serious risk of being lost by the brake pads, due to insufficient grasping at the brake lining. This condition also corresponds to too low deceleration of the hoist mechanism. Furthermore, should the hoist mechanism also feature too high braking torque, this would correspond to too high deceleration of the hoist mechanism. The consequence of which, leads to overstressing of the hoist mechanism and structure, furthermore the risk of damage at the load. Thus it is also a key consideration in determining the desired performance of the hoist mechanism, to ensure that braking torque values are with the permissible limits of the mechanism.

Till this day various solutions and devices have been proposed to achieve controlled deceleration of the industrial hoist braking. Such methods till this date include the conventional PID controller, where the set point deceleration and actual deceleration of the hoist are used as the input value of the control system. Or further a fuzzy logic based controller, with a rule based system defined for given limits of the deceleration as input and necessary force for output to manipulate the deceleration. A more recent method may be reviewed in the work [2], where a theoretical study into a fuzzy neural network based controller was proposed to achieve improved behavior of the hoist deceleration. The results of this study were compared to the conventional PID controller and fuzzy controller, with tests on the hoist as an identified model, showing better behavior via the proposed fuzzy-neural adaptive approach. However till now, hoist deceleration control has not been such a widely studied area. With certain solutions to this problem, in terms of produced devices, being in some cases uneconomical especially, for use on smaller sized hoist applications, or not being the most optimal solution. Similarly with the above mentioned controllers investigated in the industry, here too is a necessity for further study to either further optimize the proposed methods of control for more adequate functionality and also investigation into other alternative computational methods for control.

B. Applied Neural Architectures and Algorithms

Higher-order nonlinear neural units (HONU) have been shown as promising polynomial neural architectures for adaptive identification and control on engineering systems [4][6] [8].

Linear predictors, i.e., linear neural units (LNUs) are considered to be the first–order neural units. The second–order neural unit is called the quadratic neural unit (QNU) and the third–order one can be called the cubic neural unit (CNU). The general long–vector form of such higher-order neural units, is thus expressed as follows

$$\tilde{y}(k+p) = \mathbf{w} \cdot \mathbf{colx}\,, \qquad (2)$$

where $\tilde{y}(k+p)$ is the predicted value at prediction horizon of p samples, **w** denotes the row vector of all neural weights, and **colx** stands for the column vector of polynomial terms including the neural inputs and feedbacks. Particularly for LNUs, the predictor's output is calculated as

$$\tilde{y}(k+p) = \sum_{i=0}^{n} w_i x_i = \mathbf{w} \cdot \mathbf{x};\ \text{for LNU } \mathbf{x} = \mathbf{colx}\,, \qquad (3)$$

where n is the length of signal history at predictor's input so

$$\mathbf{x} = \begin{bmatrix} 1 = x_0 & y(k) & y(k-1) & \dots & y(k-n+1) \end{bmatrix}^T, \qquad (4)$$

where x_0=1 allows for neural bias in case of LNU and it also allows LNU be subset of HONUs in case of higher orders. Then, QNU can be expressed as follows

$$\tilde{y}(k+p) = \sum_{i=0}^{n} \sum_{j=i}^{n} w_{i,j} x_{i,j} = \mathbf{w} \cdot \mathbf{colx}\,, \qquad (5)$$

where the all quadratic polynomial terms are in **colx** as

$$\mathbf{colx} = \left[\left\{ x_i \cdot x_j\ ;\ i = 0 \dots n,\ j = i \dots n \right\}\right], \qquad (6)$$

and the weight vector becomes as follows

$$\mathbf{w} = \left[\left\{ w_{i,j}\ ;\ i = 0 \dots n,\ j = i \dots n \right\}\right]. \qquad (7)$$

Similarly the long vectors can be extended for CNU and higher orders, while the simple vector notation of HONU (2) can be maintained. The long-vector representation of HONU (2) also clearly highlights the fact that HONUs are non-linear mappings that are linear in parameters. This has important connotations to the learning of HONUs as their optimization is a linear problem and thus there are no local minima for a given training data.

In the sense of adaptive identification of dynamical systems, these neural architectures as such LNU, QNU & CNU may be updated incrementally via the gradient descent rule, as follows

$$\mathbf{w}(\mathrm{k}+1) = \mathbf{w}(k) + \mu\, e(k) \frac{\partial y(k)}{\partial \mathbf{w}}. \qquad (8)$$

Equation (8), depicts the gradient descent rule, for incrementally updating the LNU and HONUs neural weights, respectively. Where μ represents the learning rate of the GD algorithm, $e(k)$ (k representing the number of the sample) represents the current error between the real and calculated output of the model and the final term corresponds to the partial derivatives of the neural unit output, with respect to the individual neural weights. For this scope of this paper, these updates will be performed on dynamic neural models and thus these above forms denote a real-time-recurrent-learning (RTRL) learning method. However, in certain engineering applications, it is more suitable to train these neural weights rather than over sample-by-sample, in the form of batch training and thus, an extension of the gradient decent rule with the famous Levenberg-Marquardt equation may be used as follows

$$\Delta \mathbf{w} = (\mathbf{J}^T \mathbf{J} + \frac{1}{\mu}\mathbf{I})^{-1} \mathbf{J}^T \mathbf{e} \,. \tag{9}$$

Here **J** represents the Jacobian matrix of derivatives for the neural unit. This may be the complete partial derivatives of the neural model with respect to each neural weights, or in practical applications it seems useful to simply introduce this Jacobian matrix as the input vector or matrix itself, being **x** and **colx** for LNU and HONUs respectively. Thus, the neural weight update itself may be given in following way **w**=**w**+**Δw**.

Often in such adaptive neural units, it is apparent that a modification of the normalized learning rate may be used to solve issues of instability of learning. In practice it is possible to employ the simplified normalized learning as follows;

$$\eta = \frac{\mu}{\mathbf{x}(\mathrm{k})\mathbf{x}(\mathrm{k})^T + 1} \,. \tag{10}$$

III. Adaptive Control Approach

In this paper, an offline tuning approach is investigated to propose the potentials of the presented neural network based architectures for hoist deceleration control. Following adaptive identification of the hoist mechanism, a neural network based controller (neuro-controller) may then be extended as a state feedback configuration.

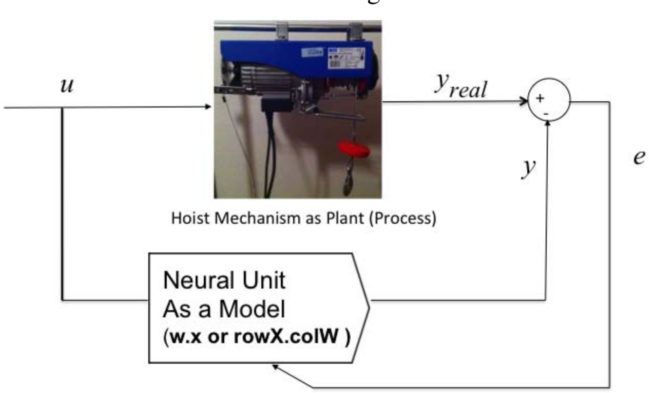

Figure 1: Adaptive Identification with supervised learning of neural networks (**w.x**=LNU, **rowX.colW** =QNU/CNU)

Thus, Figure 1 depicts the adaptive identification scheme via supervised learning of the LNU and HONUs (as such QNU & CNU) neural units. Here, u represents the control variable of the hoist (i.e.; frequency, voltage or current supply to the hoist motor or external braking system). The output y_{real}, corresponds to the hoist deceleration over time, and y, being the output of the neural unit, with the difference being the error, e. An extension of the neural unit as a model, with neuro-controller is depicted in Figure 2. Here, Figure 2 introduces a vector that comprises of a combination of outputs from the neural unit as a model and the difference between the desired behavior (in our case desired speed or deceleration of the hoist) and the output of the neural model. v. or collectively, the variable q, thus serves as a manipulator for the newly fed samples of the neural unit as an identified model. Here the GD algorithms are employed in the following manner to achieve sample-by-sample adaptation of the neural weights for the controller, as follows

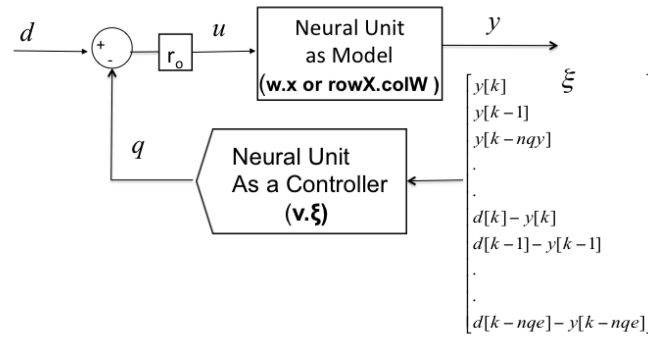

Figure 2: Adaptive control loop for experimental study of a neural network controller (**w.x**=LNU, **rowX.colW**=QNU/CNU) (modification of Figure 2 [5])

$$v_{i+1} = v_i + \mu . ereg(k) . \frac{\partial y(k)}{\partial v_i} \,. \tag{11}$$

Where v_i, are adaptable neural weights of the neural unit as a controller and $e_{reg}(k)$ is the error between the desired value of the real system (in this case the hoist, where the desired value will be denoted as d) and the real system output value at sample k (in this case the identified neural model output). The final term denotes the partial derivative of the output of the neural unit as a model, with respect to the individual adaptive neural weights of the neural unit as a controller. An extension of this weight update scheme for BPTT training would result in the following form **v**=**v**+**Δv**, where the change of neural weights for each batch would be analogical to equation (9).

IV. Experimental Analysis

A. Neural Network Model of Hoist Mechanism Deceleration Control

Following experimental analysis for a full loading condition, at lowering of the Gude GSZ 200/400, hoist mechanism. It was shown that for the first 48 measured samples of braking on the hoist deceleration characteristic, too high deceleration was present at the load, for this hoist mechanism. The following onset of samples, were thus within the permissible region of deceleration as dictated by the industrial standards [1], featuring less than 2 m/s^2 deceleration at the load, whilst still being above its permissible minimum for both static and dynamic braking torque. Thus, this paper considers control of the hoist deceleration only for the first 48 samples, of the investigated hoist deceleration characteristic. For this component, control via the QNU and CNU architecture neuro-controllers, are investigated. With the remaining portion of the deceleration characteristic delivered with the standard mechanical braking mechanism. Figure 3, depicts the adaptive identification of the investigated hoist data, with the DLNU with RTRL training. From this figure we may note that the adaptive model (green line) follows the hoist real data (blue) almost exactly. The central plot depicts the decreasing sum of square errors throughout the learning process. Its steady decay implies stable learning however this is more so evident in the weights, plotted on the lower graph. This graph depicts the learning of each individual adaptive weight in the neural model, here we may note the gradual increase of each weight for the linear neural model until its stable value is shown, this stable value corresponds to the trained coefficients of the polynomial model as seen in equation (3), represented by vector **w**. Figure 4, depicts the adaptive identification of the hoist data with DQNU trained via the BPTT learning algorithm. What we may notice is that the identification is even more precise to the real behavior of the hoist data and with faster learning, achieved over just a few epochs or runs of the learning algorithm. The BPTT algorithm is advantageous in learning data affected by noise, as such here where a minor disturbance is present. The model thus focuses on the main governing law as opposed to learning this behavior. Following these results of adaptive identification, it was found that the DQNU with BPTT training featured the best performance in identification of the investigation hoist deceleration data.

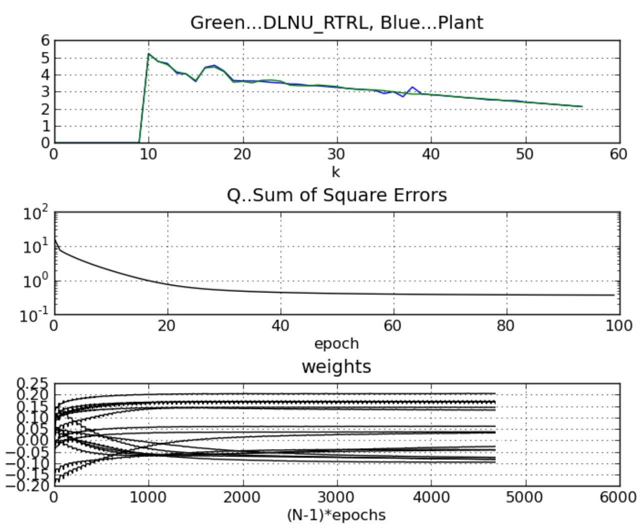

Figure 3: Adaptive Identification of Hoist Mechanism Deceleration at Load, with DLNU and RTRL Training, ny=9 (No of previous samples of neural model output), nu=5(No. of previous samples of real process input), mu (learning rate)=0.2, epochs (runs of algorithm)=100

A neuro-controller is thus extended for control of the hoist deceleration characteristic, with the neural unit as an identified model being of that presented in Figure 4. Figure 5 & Figure 6, show the extension of a neuro-controller in state feedback as applied to the hoist deceleration data, with the black line depicting the desired behavior, the blue being the real system and the magenta line being the neuro-controller output. Figure 5 & Figure 6, thus depicts the application of the QNU & CNU neuro-controller with BPTT training methods respectfully. As we may notice, the QNU controller achieved adequate control, following closely to the desired behavior after several initial samples at application. However, at the initial moment of braking, the CNU delivers a closer value to the desired behavior than the QNU. Where the QNU features a peak of almost 3 m/s^2, corresponding to the initial moment of application during the braking phase.

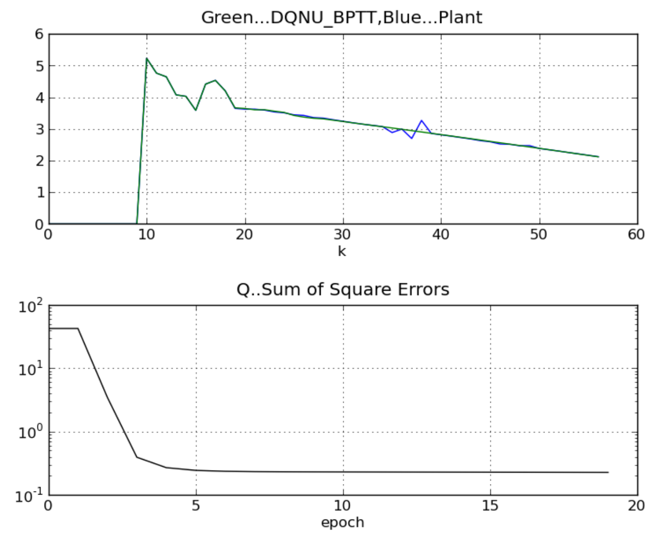

Figure 4: Adaptive Identification of Hoist Mechanism Deceleration at Load, with DQNU and BPTT Training, ny=9 (No of previous samples of neural model output), nu=5 (No. of previous samples of real process input), mu (learning rate)=0.2, epochs (runs of algorithm)=20

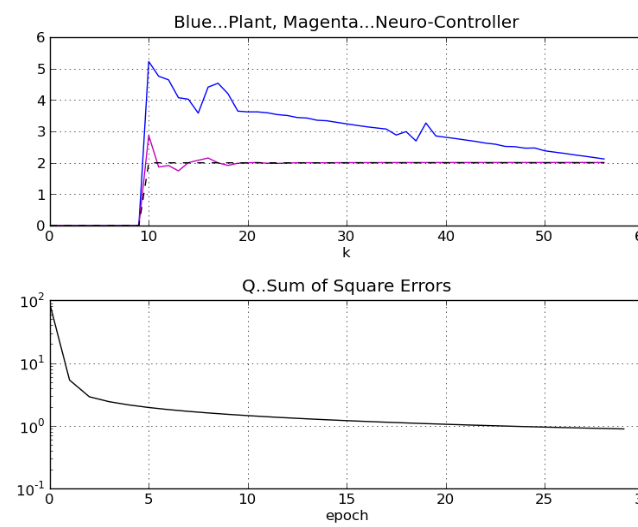

Figure 5: Adaptive Control of Hoist Mechanism Deceleration at Load, with QNU BPTT Training, nqy=9 (No of previous samples of neural model output), nqe=0 (No. of previous samples of difference between desired and output of the neural model), mu (learning rate)=1, epochs (runs of algorithm)=30 (Trained with DQNU BPTT as per Figure 4)

However, the CNU featured 2.6 m/s^2, which is more desirable. Following this initial value at braking, all other points in both the QNU and CNU featured close values to the desired behavior of the hoist deceleration. Thus, from these results in Figure 5 & Figure 6, we may see the potentials that the investigated neural architectures have in controlling the hoist deceleration to achieve close response to the desired deceleration value. Another key consideration is that for the above tested case, a large variation is present from the real deceleration of the hoist and the desired behavior. In spite this, the neuro-controller is still able to quickly respond to the desired behavior of the system within the first few samples, and maintain a steady desirable output value even when a change in the dynamics of the system, or large disturbance may occur.

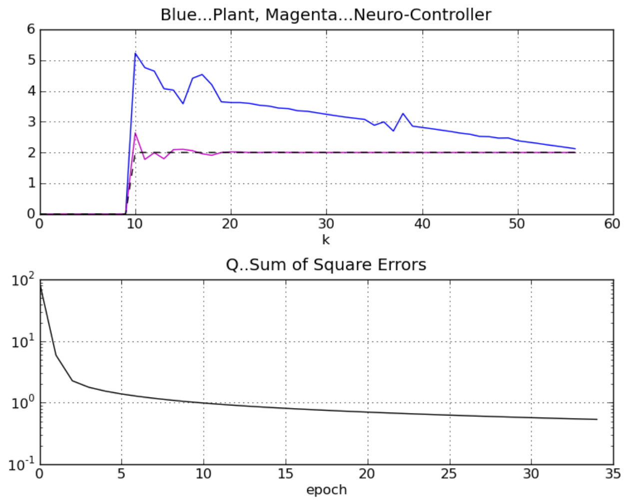

Figure 6: Adaptive Control of Hoist Mechanism Deceleration at Load, with CNU BPTT Training, nqy=9 (No of previous samples of neural model output), nqe=0 (No. of previous samples of difference between desired and output of the neural model), mu (learning rate)=1, epochs (runs of algorithm)=35 (Trained with DQNU BPTT as per Figure 4)

B. *Analysis of Braking Torque Following Control*

Following the above investigation of the neuro-controller as applied to the investigated hoist deceleration data (Figure 5 & Figure 6), further analysis into the resulting braking torque of the hoist mechanism is considered in Figure 7 & Figure 8. Figure 7, depicts the static braking torque analysis before and after the application of the neuro-controller (CNU with BPTT) to the hoist mechanism deceleration data. Here the original static braking torque value (blue), calculated from the averaged deceleration over the first 53 samples of data (for the purpose of concluding an overall braking torque value of the hoist mechanism) is 3.8 N.m, which is too high, corresponding to too high rate of deceleration of the hoist mechanism in its initial stages of braking. During the initial stages of control, the neuro-controller (magenta) featured a few samples of deviation from the desired behavior of the hoist mechanism and correspondingly the braking torque values too, represent this component of the behavior.

However, in spite this, the calculated values of static braking torque are far more desirable, considering the average deceleration produced by the neuro-controller, after settling. Figure 8, depicts the calculated dynamic braking torque for the investigated hoist mechanism before and after application of the neuro-controller. Here a reflection in the values of the braking torque characteristic can be seen, with respect to the controlled deceleration values.

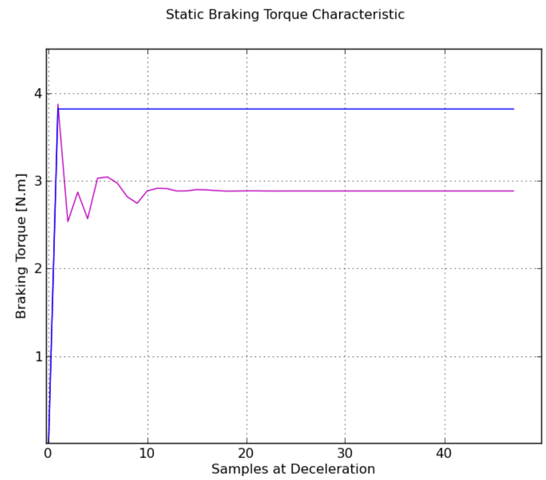

Figure 7: Static Braking Torque Analysis - Adaptively Tuned Neuro-Controller (CNU with BPTT) with Hoist Mechanism data (magenta), and Original Hoist Mechanism Data (blue)

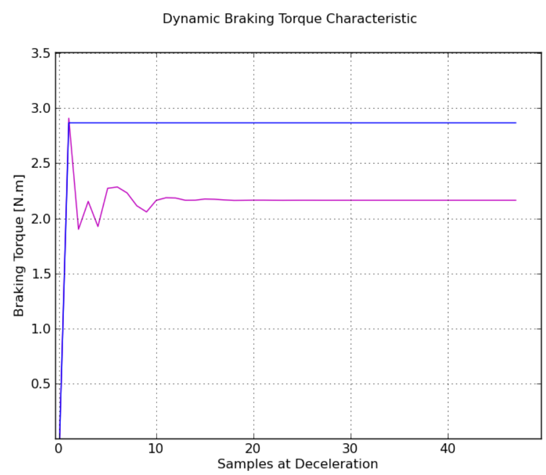

Figure 8: Dynamic Braking Torque Analysis - Adaptively Tuned Neuro-Controller (CNU with BPTT) with Hoist Mechanism data (magenta), and Original Hoist Mechanism Data (blue)

During the initial stages of control, the neuro-controller featured a few samples of deviation from the desired behavior of the hoist mechanism and correspondingly the braking torque values too, represent this component of the behavior. However, in spite this, the calculated values of dynamical braking torque are far more desirable, considering the average deceleration produced by the neuro-controller, after settling. With its final value being 2.2 N.m and thus deviating from the minimum prescribed value by 1.23 N.m which is desirable.

V. Discussion

Following experimental analysis of the proposed neural architectures (i.e. QNU and CNU) for hoist deceleration control, the produced results showed that although quite large deviation in the output data of the hoist mechanism with respect to its desired behavior was shown, the adaptive identification via a neural network based model, particularly the DQNU with BPTT, proved to model the behavior of the hoist deceleration characteristic data quite well. We may also note the excellent speed of convergence of the neuro-controller, as seen in Figure 5 & Figure 6 where the

QNU and CNU with BPTT achieved fast convergence in its sum of square errors. Following this an extension of the neuro-controller, via QNU and CNU with BPTT, quite desirably followed the behavior of the desired hoist mechanism output during simulation. As seen in Figure 5 & Figure 6, there was a small deviation in its initial values, from the desired value, it is thus an important future direction to test behavior of such proposed neural network setup (particularly the QNU & CNU with BPTT training as a neuro-controller) on the real hoist mechanism, as may be recalled in work [6], the results of the neuro-controller after real application indeed proved to be of better performance as compared with its simulation. Following experimentation of such control on the real system, one may also identify any potential for further optimization. Another consideration is that with larger hoist applications as such considered in works [2] & [3], the overall deceleration time is several folds higher, in the order of several seconds, as compared to the tested hoist mechanism. Thus, testing regarding the capability of actuation for control with the real application is the next stage of research, following these promising results. Furthermore, should such a hoist mechanism setup already feature an initial means of control (i.e; conventional PID or fuzzy control), the proposed neural network control also features potentials for further optimization and thus, arises even further motivation for research on other hoist mechanisms in the field of this study.

VI. CONCLUSIONS

From this application study, we may see the potentials that especially the QNU and CNU with BPTT training has in providing desirable behaving control to the hoist mechanism data. With the BPTT training method, being able to identify the process data with better performance, particularly with the noised regions within the data being ignored and the model focusing only on the main governing law of the hoist mechanism output data. Following this QNU with BPTT identification, it was shown that with only several runs of the neuro-controller algorithm, the neuro-controller was tuned to provide desirable control of the hoist mechanism, following closely to the desired behavior almost exactly with exception of the first several samples at the beginning of the controller's application. A future goal following these results is implementation of this proposed neural network setup for real time control of an industrial hoist mechanism. As may be recalled in the work [6], the real application of the tuned neuro-controller provided better performance than that of its simulation, following proper setup of the adaptive control algorithm. With real application on an industrial hoist mechanism, further optimization on the real system may also be necessary and thus also leading to a further point in research.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Ladislav Novák, former Maintenance Manager/Inspector for Montáže s.r.o. for providing us with conditions for experimental analysis and also to Mr. Jan Beneš, the specialist in crane design from IBCE Pty.Ltd., for his valuable advice and expertise.

REFERENCES

[1] Australia Standards Association, *AS 1418.1-2002 – Cranes, hoists and winches Part 1 –General requirements*, Fourth Edition, Sydney, Australia ISBN 0 7337 4372 2, 2002

[2] Liu Jingyan, Wang Fuzhong: "The Fuzzy Neural Network Control of Hoist Constant Deceleration Braking System Based on Genetic Algorithm," *ICDMA*, vol. 2, pp.687-689, 2010 International Conference on Digital Manufacturing & Automation, 2010

[3] Wu Yanxiang, Tian Zhanfei, Xue Minjie, "Study and Simulation on Variable Frequency Speed-Regulating System of Mine Hoist Based on PLC Fuzzy Control," *CISP 2010, 3rd International Congress on Image and Signal Processing*, 2010

[4] Bukovsky I., S. Redlapalli and M. M. Gupta (2003): Quadratic and Cubic Neural Units for Identification and Fast State Feedback Control of Unknown Non-Linear Dynamic Systems, *Fourth International Symposium on Uncertainty Modeling and Analysis ISUMA* 2003, IEEE Computer Society, , Maryland USA, ISBN 0-7695-1997-0, p.p.330-334

[5] Peter M. Beneš (2013): Software Application for Adaptive Identification and Controller Tuning Student's Conference STC, Faculty of Mechanical Engineering, CTU in Prague. 2013 (Awarded paper: STČ + Bosch + STOČ Zlín)

[6] Ladislav Smetana: *Nonlinear Neuro-Controller for Automatic Control Laboratory System*, Master's Thesis (sup. Ivo Bukovsky), Czech Tech. Univ. in Prague, 2008

[7] Gupta, M., M., Bukovsky, I., Homma, N., Solo M. G. A., Hou Z.-G. (2012): "Fundamentals of Higher Order Neural Networks for Modeling and Simulation", in *Artificial Higher Order Neural Networks for Modeling and Simulation*, ed. M. Zhang, IGI Global.

[8] Bukovsky, I., Homma, N., Smetana, L., Rodriguez, R., Mironovova M., Vrana S. (2010): "Quadratic Neural Unit is a Good Compromise between Linear Models and Neural Networks for Industrial Applications", ICCI 2010 The *9th IEEE International Conference on Cognitive Informatics*, Tsinghua University, Beijing, China, July 7-9.