JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
3, **40**, 2002

DEDICATED NEURAL NETWORK DESIGN FOR FRICTION COMPENSATION IN ROBOT DRIVES

Zbigniew Korendo

ABB Corporate Research Cracow
e-mail: Zbigniew.Korendo@pl.abb.com

Tadeusz Uhl

Department of Robotics and Machine Dynamics, University of Mining and Metallurgy, Krakow
e-mail: tuhl@rob.wibro.agh.edu.pl

In the paper we demonstrate a neural network-based controller design and prototyping following the mechatronic approach. A unified treatment of all system components (mechanical, electrical and computational) is made possible thanks to the integrated software-hardware platform. The neural network in the presented approach is used to provide a linearising feedback loop for friction compensation in a robot drive. The efficiency of the experimental friction identification is improved thanks to dedicated network architecture. The proposed solution is implemented in DSP hardware and the simulation results are verified through laboratory experiments.

Key words: friction modelling, mechatronics, neural networks for control

1. Introduction

Increasing demand for systems and devices meeting stringent criteria regarding high precision positioning as well as trajectory tracking lead to growing research activity in the field of motion resistance compensation. It is reflected by numerous publications in many scientific journals and also friction modelling and compensation issues being the key thematic sessions at the most significant control and robotics conferences. Friction can hardly be neglected it introduces highly non-linear and discontinuous dynamics. It is also very difficult to identify and model analytically. A comprehensive survey of friction effects in control tasks and compensation techniques can be found in Armstrong-Helouvry et al. (1994).

In the field of nonlinear control systems many researchers use neural networks as advanced nonlinear controllers. Some properties of neural networks, such as their inherent computational parallelism, fuzzification of knowledge represented by particular patterns along with the effective neural net training algorithms made this new tool attractive for control of robotic manipulators. Among neural control techniques for manipulation robots the most often are reported: controller parameters tuning (gain scheduling) via neural look-up table (Hunt eta al., 1992), adaptive control with neural parameters adaptation (Khemaissia and Morris, 1993), inverse dynamics modelling with a neural model (Kawato and Wada, 1993), inverse dynamics and PD controller realised as neural network (Uhl and Szymkat, 1992), model predictive control scheme with neural prediction (Kawato and Wada, 1993) and other.

Neural network-based friction compensation has attracted increasing interest. Many researchers report successful experimental implementation of neural-based controllers. Neural networks for friction compensation are used to control current pulse width (Yang and Tomizuka, 1988), dither-like signal generation (Deweerth et al., 1991), as a compensation look-up table based on CMAC architecture (Larsen et al., 1995) and also as nonlinearity identifiers (Ciliz and Tomizuka, 19995; Du and Nair, 1999). Funahashi (1989) theorem guarantees for any nonlinear function the existence of a nonlinear, multilayer neural network with a sufficient number of neurones able to approximate it with arbitrary precision. The general problem, however, with doing neural-based identification is that Funahashi theorem is not constructive in the sense that it does not state necessary and sufficient conditions for the optimal network structure design for a given task. A usual approach involves substantial trial-and-error efforts (based on past experience) with training set selection and training process convergence as the principal problem areas.

2. Problem formulation

The effect of friction upon mechanical systems can be assessed from manifold perspectives: mechanical design, wear processes, kinematics, dynamics, simulation, control, etc. Its analysis requires diverse analytic treatment and software tools usually hindering a synergetic approach. The mechatronic methodology largely applied in this study offers a unified treatment of all system aspects – mechanics, drives, control electronics, sensors and numerical processing – within common analysis and simulation environment of Matlab-Simulink-dSPACE. Its effective use requires diversity of research competencies ranging from robotics, through sensors, theory of drives, digital signal processing, control systems, optimisation methods, systems identification, digital simulations, until systems modelling, programming and artificial intelligence.

This study aims at the development of a theoretical concept and technical implementation of an intelligent controller for a typical industrial robot drive. The controller uses artificial intelligence tools to provide friction modelling and compensation. This goal is achieved with the use of mechatronic design methodology consisting in iterative execution of the following research tasks:

- problem formulation, draft solution proposal, CAD design, sensors and actuators selection
- classical friction modelling and identification (reference data)
- experimental model verification and validation
- dedicated neural architecture design and training for friction compensation
- controller synthesis (neural-based and reference classical) – analysis and optimisation using simulation
- DSP-based prototyping – experimental testing of the neural controller, control quality and robustness assessment, iterative parameters fine-tuning
- final hardware implementation – dynamic performance verification of actuators.

Another mechatronic aspect of this work consists in the use of tribology foundations for design of control algorithms for mechanical systems with friction. It features marginal scope of analytic treatment for friction modelling but instead significant contribution of experimental knowledge about the modelled phenomenon. A dedicated neural network architecture was developed using known qualitative properties of the friction model. The neural network is specifically designed to yield qualitative characteristics of the modelled nonlinearity. This way, the neural network does not operate as a "black-box", but rather as a "gray-box" i.e. it makes explicit use of the knowledge about the physical properties of the controlled object.

3. Dedicated network design for friction compensation

In general, robot dynamics can be expressed in the form

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{F}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{\tau} \tag{3.1}$$

where $\boldsymbol{\theta}$ denotes angular position, $\mathbf{M}$ is inertia matrix and $\boldsymbol{F}$ denotes all nonlinear torque components, including friction and $\boldsymbol{\tau}$ is the driving torque.

The proposed control algorithm is based on the nonlinear feedback linearisation (NFL) scheme (Uhl et al., 1994). Two control loops are used: the inner loop with a model of nonlinearities to cancel out the system nonlinearities and the external loop with the standard linear PD control to cancel uncertainty of model parameters and unmodelled disturbances. The overall control signal is given by means of the following equation

$$\boldsymbol{\tau} = \widehat{\mathbf{M}}(\boldsymbol{\theta})[\ddot{\boldsymbol{\theta}}_d + \mathbf{K}_V\dot{\boldsymbol{e}} + \mathbf{K}_P\boldsymbol{e}] + \widehat{\boldsymbol{F}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{3.2}$$

where dashed symbols denote estimations of the inertia and nonlinearities, respectively, $\boldsymbol{\theta}_d$ denotes the desired (set) angular position and $\mathbf{K}_P$, $\mathbf{K}_V$ are proportional and derivative gains of the PD controller. Assuming that $\widehat{\mathbf{M}}(\boldsymbol{\theta}) = \mathbf{M}(\boldsymbol{\theta})$, $\widehat{\boldsymbol{F}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{F}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ (i.e. there are no modelling errors), the control error dynamics is governed by (where $\boldsymbol{e} = \boldsymbol{\theta}_d - \boldsymbol{\theta}$)

$$\ddot{\boldsymbol{e}} + \mathbf{K}_V\dot{\boldsymbol{e}} + \mathbf{K}_P\boldsymbol{e} = \mathbf{0} \tag{3.3}$$

The applied control structure is depicted in a block diagram (Fig. 1).

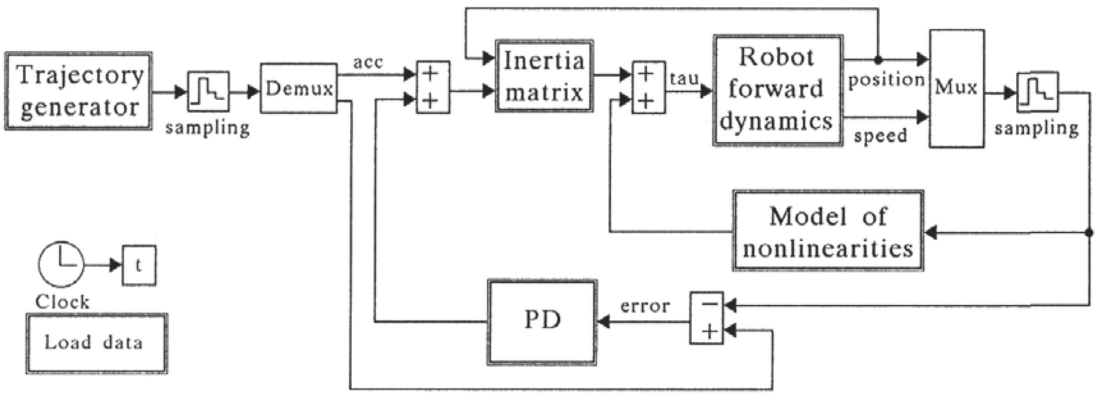

Fig. 1. Scheme of nonlinear feedback linearisation control

The role of the neural network is to provide a nonlinear friction model in the inner control loop. The proposed solution relies on the network architecture design according to the modelled nonlinearity properties.

In this study the exponential friction model (3.4) and (3.5) of Tustin-Armstrong-Helouvry was used (Armstrong-Helouvry et al., 1994). The neural model synthesis consists in embedding three basic properties of static friction characteristics (velocity vs. friction force) into the network architecture: friction force sign reversal with velocity sign change, Stribeck's effect (friction force drop at low velocity) and viscous friction component linear to velocity change beyond certain threshold velocity (Stribeck's velocity), which reads

$$\begin{aligned} F_+(\omega) &= F_{s+} + F_{Str}(\omega) + F_v\omega \qquad\qquad \omega > 0 \\ F_s(\omega) &= F_s + \operatorname{sgn}\omega \\ F_{Str}(\omega) &= (F_C - F_s)\Big\{1 - \exp\Big[-\Big(\frac{\omega}{\omega_s}\Big)^2\Big]\Big\}\operatorname{sgn}\omega \end{aligned} \tag{3.4}$$

and

$$F(\omega) = \Big[\operatorname{sgn}\omega,\ \Big\{1 - \exp\Big[-\Big(\frac{\omega}{\omega_s}\Big)^2\Big]\Big\},\ \omega\Big] \begin{bmatrix} F_s \\ F_C - F_s \\ F_v \end{bmatrix} = \Psi(\omega)\phi \tag{3.5}$$

where ω denotes the angular velocity, ω_s is Stribeck's velocity, F_v is the viscous friction coefficient, F_s and F_C denote the static and Coulomb friction torques, respectively.

The exponential model of Tustin-Armstrong-Helouvry was reformulated into a form linear with respect to certain parameters: a linear combination of a set of the nonlinear basis functions $\Psi(\omega)$ (corresponding to the three friction components) and a vector of the coefficients Ψ, Eq. (3.5). It can be shown that both Stribeck curve and linear viscous friction can be approximated with hyperbolic tangent functions with a proper selection of the parameters Ψ. For example, the linear viscous friction component is modelled by scaling the velocity input (by the input connection weight w_i, where the index i denotes ith neuron) to stay within a quasi-linear portion of the hyperbolic tangent characteristics

$$\forall i = 1,2,3 \qquad \Psi_i(\omega) \cong y_i(\omega) = \tanh(w_i\omega) \tag{3.6}$$

Model (3.4) and (3.5) can be realised as a nonlinear, unidirectional, two-layer neural network with hyperbolic tangent activation functions, given by Eq. (3.7), where $y_i(\omega)$ denotes the output of the ith hidden neuron that corresponds to the ith component of the basis function $\Psi(\omega)$

$$y(\omega) = \sum_{i=1}^{3} w_{2,i}y_i(\omega) = \sum_{i=1}^{3} w_{2,i}\tanh(w_{1,i}\omega) \approx \Psi(\omega)\phi \tag{3.7}$$

Thence, each hidden layer neuron is dedicated to model certain distinguished friction phenomenon: "sign neuron" (signum), "Stribeck neuron" and "viscous neuron". It should be noted that in general each basis function can be modelled by more than one neuron.

The proposed two-layer network architecture is shown in Fig. 2.

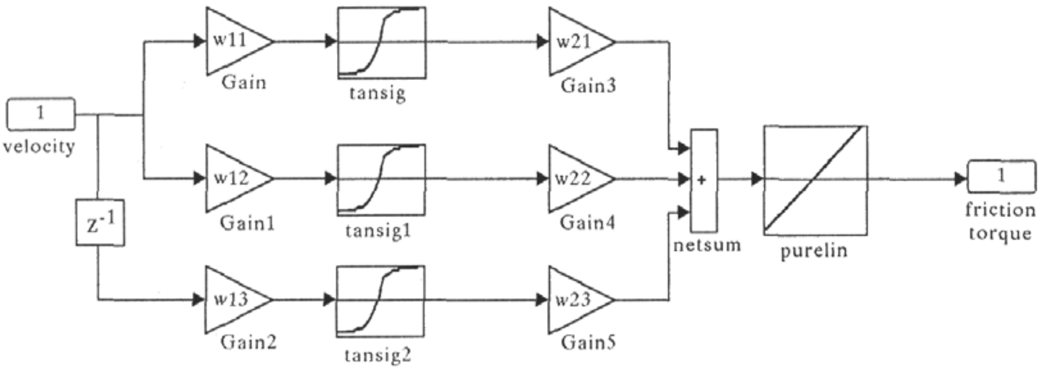

Fig. 2. Dedicated neural friction approximator

The proposed neural friction model has the advantage that the learning convergence is improved because the qualitative features (e.g. overall shape) of the modelled nonlinearity are embedded into the network architecture. For the same reason it was possible to a use smaller set of training data and the network did not need to detect the discontinuity because it was already there, embedded in its structure. Neural connection weights have a direct relation to qualitative features of the static friction characteristics – e.g. tuning one weight adjusts the slope of the linear portion related to viscous friction. As the network training progresses the changes of connection weights result in neural friction model adjustments as depicted graphically in Fig. 3.

Friction forces in real mechanical systems are not discontinuous function of time but show hysteretic behaviour. When the mass motion comes at rest, the friction forces counteract the external torque that is too small to sustain the motion (Dahl effect, see Armstrong-Helouvry et al. (1994)). This phenomenon is approximated in the proposed approach by a relay-type static friction dependence on the velocity. When the velocity crosses zero the static friction does not change the sign instantaneously but after a certain period of time, in proportion to the speed of the velocity change. To account for this phenomenon the second input to the network is the velocity signal delayed by one sample period. In this way the proposed network captures the hysteretic behaviour around zero velocity known from experimental work (Armstrong-Helouvry et al., 1994).

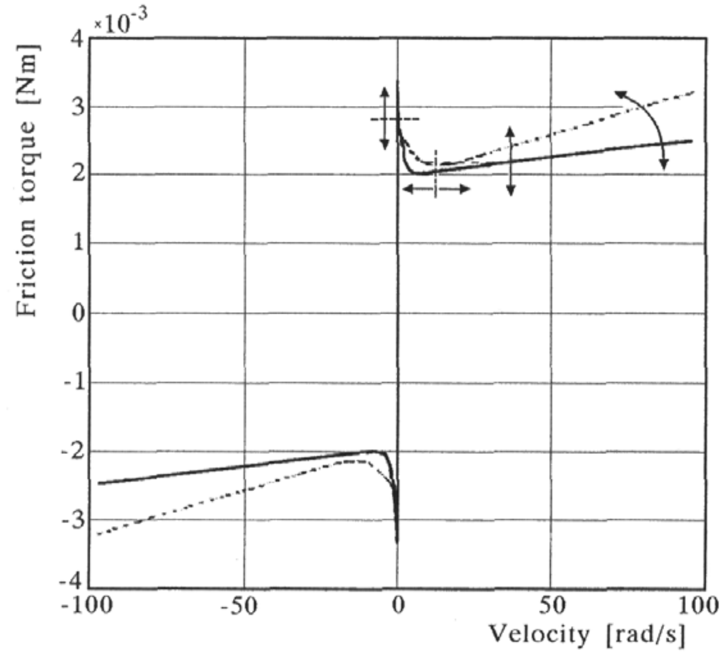

Fig. 3. Neural friction model adjustments with weights changing

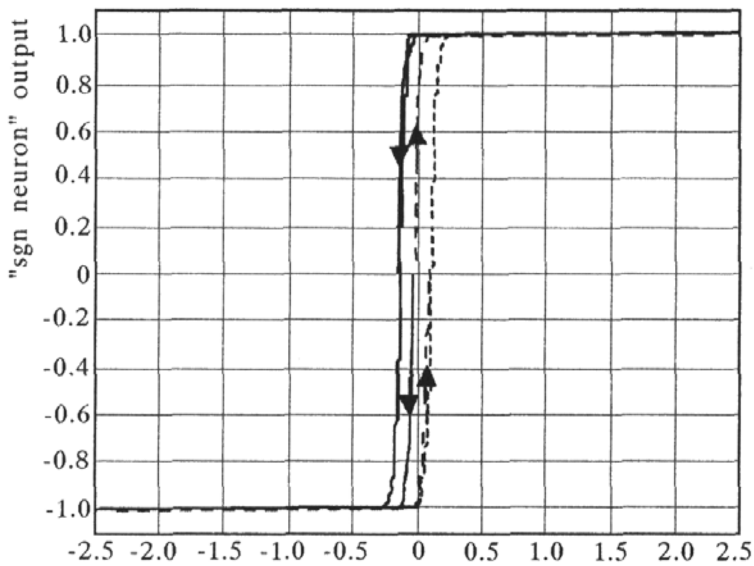

Fig. 4. "Sign neuron" output as input signal changes from positive to negative (arrows downwards) and reverse

This solution is in fact only an approximation of stiction dynamics as in general the friction force is known to be a function of the external force applied to the interface and, at the motion rest, may not necessarily drop down to zero. Besides when the motion stops for longer than one sample period then the actual function realised by "sign neuron" is different from the plain hysteresis and is shown in Fig. 4 (actual output magnified for demonstration purpose). Thence, the proposed neural friction model cannot yield the ideal friction characteristics at the motion start. Experiments (Korendo, 1999) revealed though that it is not a major constraint as eventual hardware implementation imposes actuator bandwidth constraints anyway.

Prior to actual network training it is essential to initialise weights with heuristic values known to yield the required qualitative network performance (e.g. velocity scaling factor for viscous friction modelling should be on the level of 1e-4). Then selective weights pruning is applied during the training (note incomplete network connections, Fig. 2). In this study it was implemented into standard training procedures.

The training data were collected experimentally during the identification phase. In practical implementation it can precede the actual trajectory tracking task (i.e. the system operates according to the sequence: *update nonlinearities model – perfom the task*). It consisted in driving the controlled object at a number of desired constant velocities (with a PI controller) to collect the corresponding motor torque (current) data (Fig. 4a). The motor current measurement was assumed to be the only source of friction torque data. Because the required velocity was kept constant the motor torque was approximately equal to the friction torque at this velocity. The thus obtained set of *velocity-torque* pairs describes the static friction characteristics to be modelled by the neural network. Because the network, by its structure, kept the quasi-linear viscous friction characteristics, a relatively small number of data samples was required to model viscous friction well (to adjust the slope). Additionally, the training set was extended manually with data points related to static friction (stiction torque approximation).

The network weights, Eq. (3.8), were adjusted by the training with a modified Levenberg-Marquardt algorithm (batch mode with pruning) on the training set obtained experimentally (136 bi-directional *velocity-torque* pairs – Fig. 5a)

$$W_1 = \begin{bmatrix} 3.12 & 0 \\ 0.0006 & 0 \\ 0 & 11308 \end{bmatrix} \qquad\qquad W_2 = \begin{bmatrix} -0.018 \\ 2.05 \\ 0.11 \end{bmatrix} \tag{3.8}$$

The network converged after 236 epochs and its performance was verified with the validation set of input data. Good quality of the neural model of static friction characteristics was confirmed with experimental data (Fig. 5b).

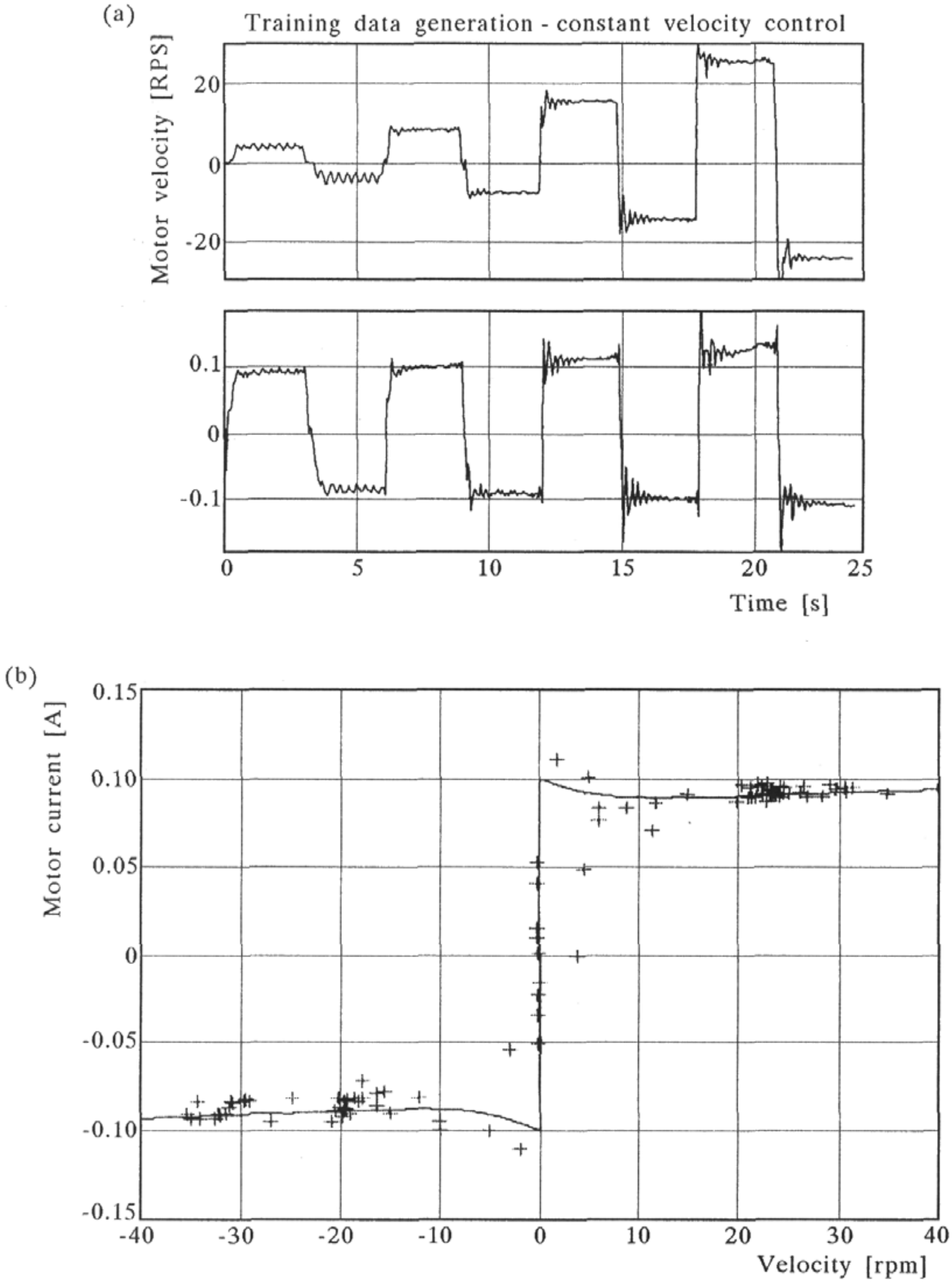

Fig. 5. Experimental training data set (a) and neural friction model after training (b) (model-solid, training data-cross)

4. Controller simulations and experimental prototyping

The proposed neural controller performance was tested by a numerical simulation in Simulink. The block diagram used for simulations is shown in Fig. 6a. Also the standard PID control scheme was simulated for comparison. The model of the controlled object used in the simulations including friction was obtained by an extensive identification process (Korendo, 1999). All simulations were carried out with the fixed-step Euler integration method (step value: 1e-4). The analytical friction model was simulated using Karnopp approach (Armstrong-Helouvry et al., 1994; Korendo, 1999) to simulate systems with discontinuities. The PID gains were adjusted with NCD Toolbox and were set to $K_P = 0.01$, $K_I = 0.07$, $K_D = 0.005$ for the standard PID and $K_P = 1000$, $K_I = 0$, $K_D = 500$ for the neural based NFL.

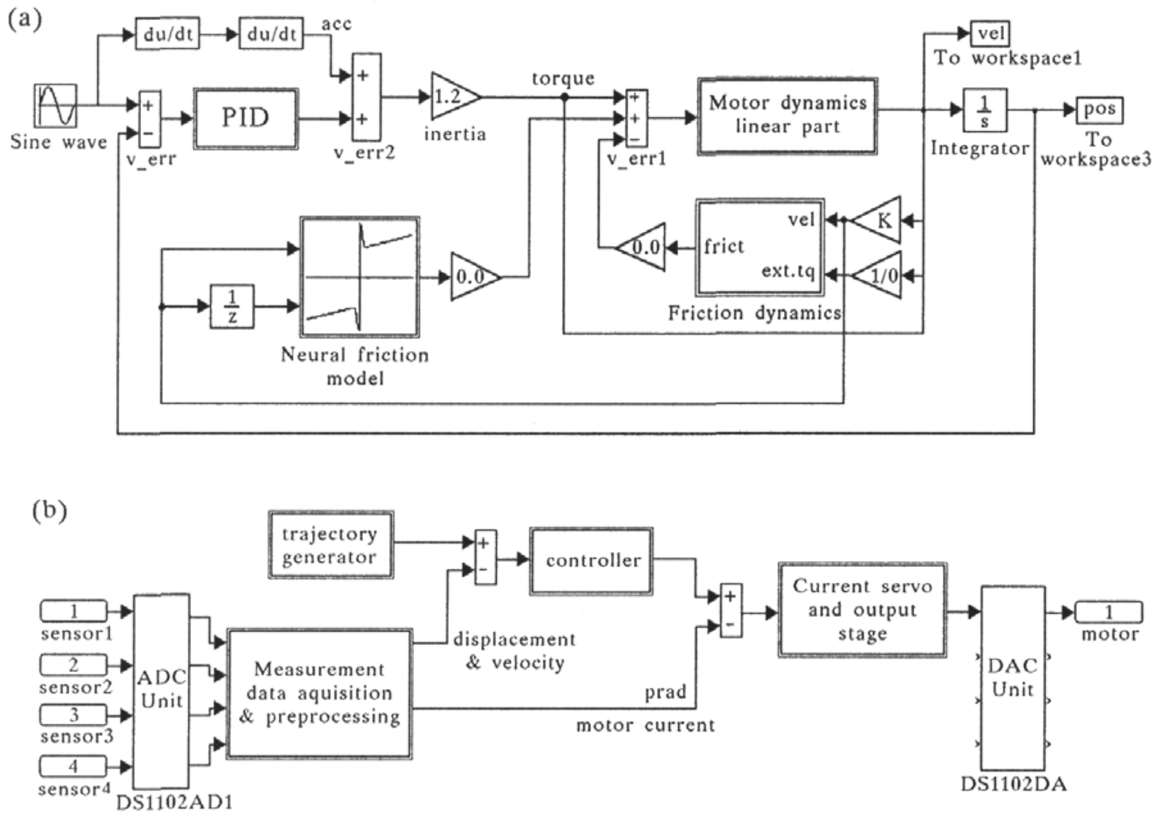

Fig. 6. Simulink diagram for controllers evaluation by simulations (a) and experimental DSP implementation (b)

The case study object was a prismatic link of the SCARA-type robot (built at Dept. of Robotics and Machine Dynamics, AGH, see Korendo (1999)), Fig. 7a. It consists of a mass (an end effector) sliding along a linear path over four stiff driving rods. The mass is driven via steel string transmission by a geared DC motor. The system is equipped with a position sensor (linear potentiometer), motor shaft rotary encoder and motor current measurement.

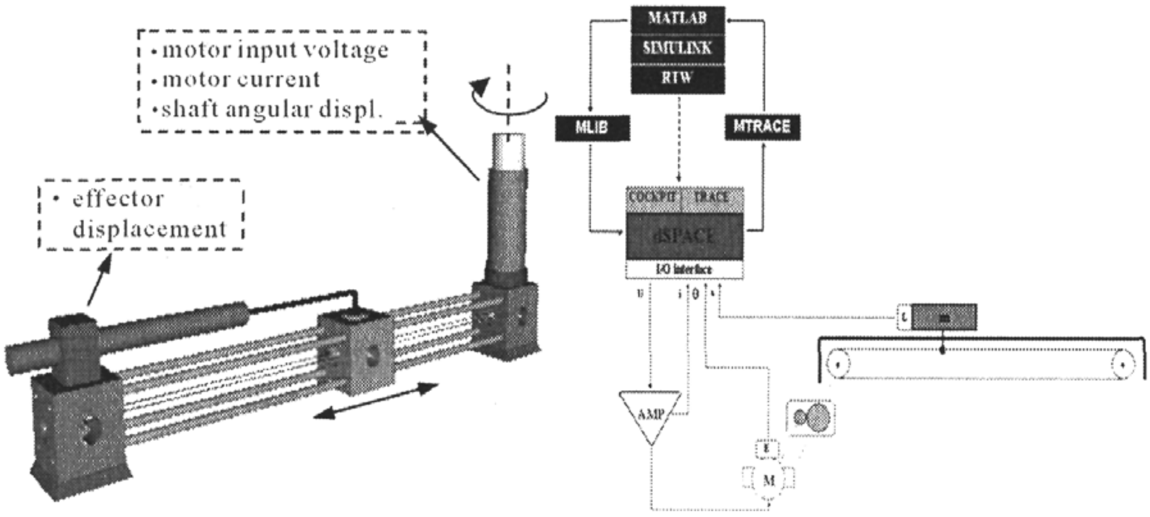

Fig. 7. Experimental setup – prismatic link of a SCARA robot (a) and DSP-based prototyping setup (b)

The effector was controlled to follow the commanded velocity trajectory given by the equation: $vel\_d = 12\sin(2.5t)$. According to expectations, the PID controller could not cope with the velocity control around zero velocity. The assumed criterion for the comparison of the control performance was RMS velocity error per one trajectory cycle, averaged over 20 cycles. The simulation results proved the expected superiority of the NFL controller over the standard PID. The RMS of the velocity error was equal to 3.17 rps (rotations per second) with the PID control whereas the NFL controller yielded 0.87 rps.

The motor was controlled in a torque control mode with the use of a high-gain current servo loop. The control and measurement feedback signals were provided by a DSP board (dSPACE DS1102) running at 40MHz. The DSP board was fully integrated with Matlab/Simulink environment from which it was controlled. A general model for DSP- based prototyping is shown in Fig. 6b. The controllers prototyping (simulations and experimental validation) was carried out with the use of Matlab/Simulink real-time environment RTW/RTI. Data acquisition and parameters tuning on-line during the experiments was performed using dSPACE COCKPIT and TRACE applications. The overall DSP-based laboratory setup for rapid prototyping is shown in Fig. 7b.

The performance of the new controller was verified experimentally in the laboratory with the use of the dedicated neural controller hardware implemen-

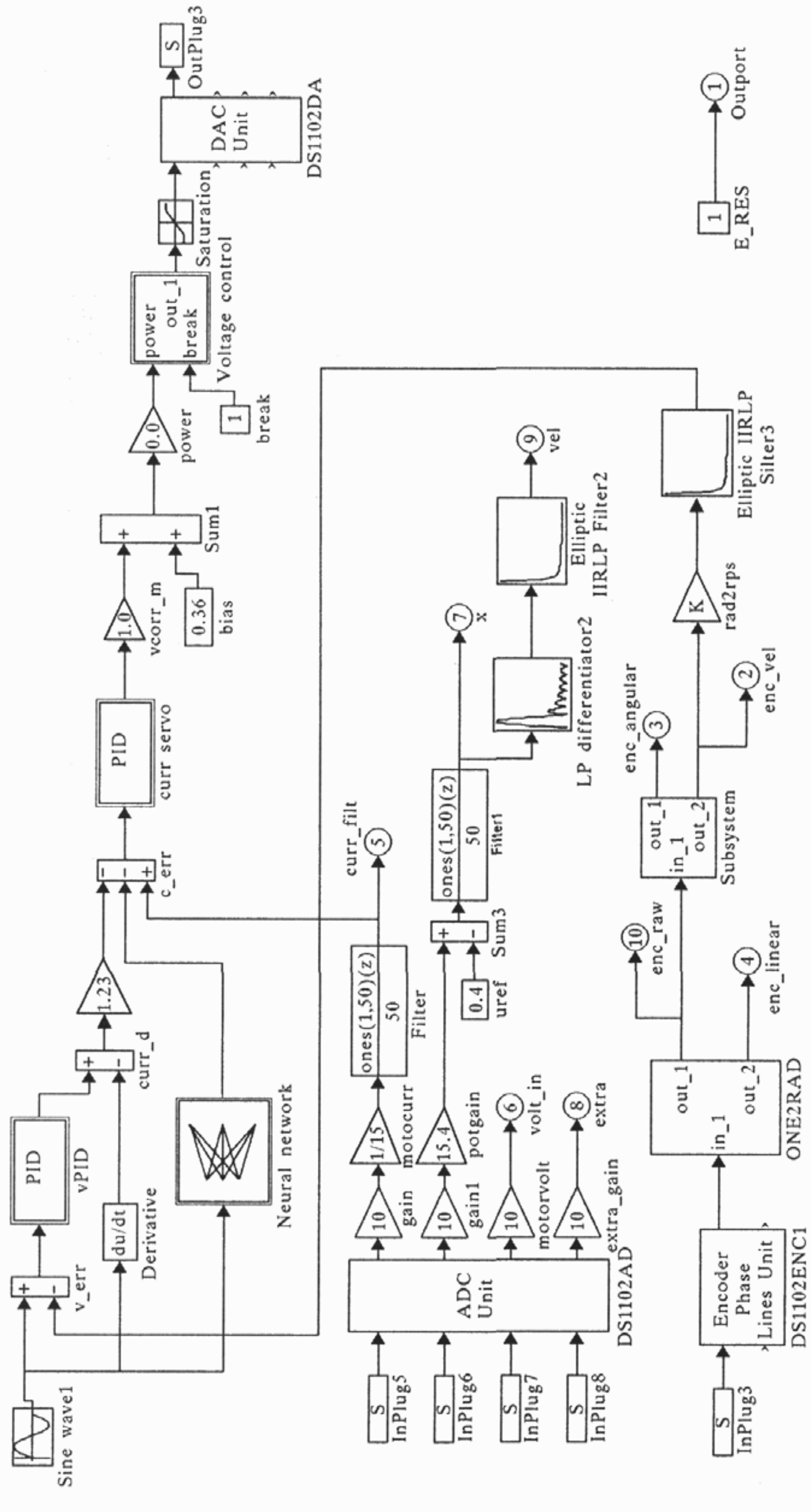

Fig. 8. dSPACE model of neural friction approximator

tation on a DSP board. The actual dSPACE model used for the experiments is shown in Fig. 8. Such neural network implementation allows through dSPACE-MLIB module for effective adaptation of synaptic weights in the training stage, because all parameters are available on-the-fly in DSP program memory.

The user interface was designed in COCKPIT environment to control the experiment status, adjust the system parameters on-line and monitor the performance (Fig. 9), including numeric displays and graphs as well as saving selected signals with TRACE module.

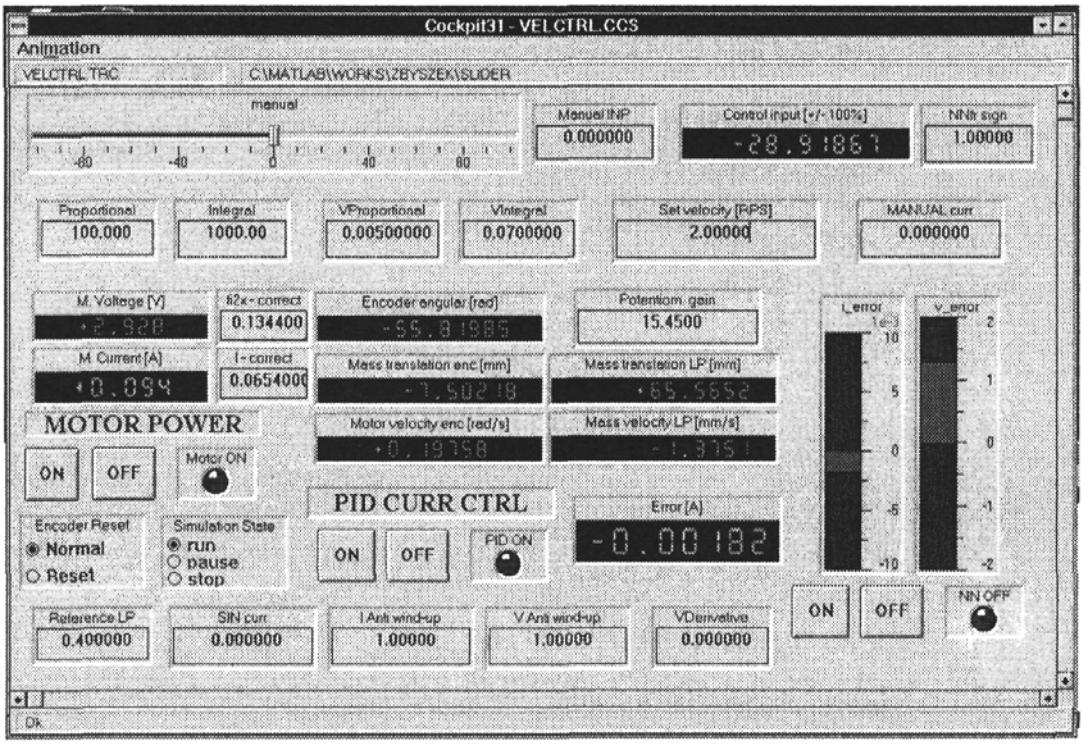

Fig. 9. COCKPIT experiment control panel

In both control schemes the PID gains that were used for simulation had to be significantly reduced for the experimental implementation. They were set to $K_P = 0.005$, $K_I = 0.04$, $K_D = 0.001$ for the standard PID and $K_P = 600$, $K_I = 0$, $K_D = 250$ for the neural based NFL.

The system performance was tested for the same velocity trajectory as the one used in the simulations. Figure 10 shows experimental results of both experiments (PID control and neural-based NFL).

The proposed neural-based nonlinear feedback linearisation scheme clearly outperforms the standard PID controller in the presence of friction. The experimental results obtained using the fast prototyping DSP based technique also confirmed the system behaviour predicted by simulations.

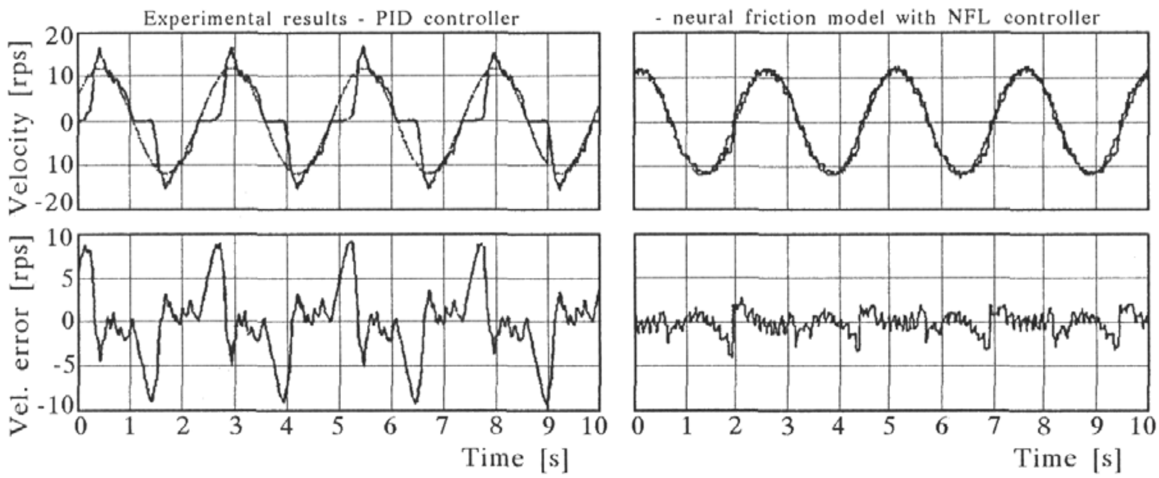

Fig. 10. Friction compensation by PID and NFL – experimental results (velocity control, dashed line: desired velocity)

Table 1. Experimental velocity control error – results summary — PID and NFL comparison.

Control algorithm	Experiment results	
	RMS [rps]	improvement [%]
PID	4.06	–
neural NFL	1.08	73.4

Again, the comparison criterion was RMS velocity error per one trajectory cycle, averaged over 20 cycles. In the velocity control experiment the PID controller yields the RMS of the velocity error equal to 4.06 rps, whereas with the neural network control it was possible to decrease the velocity error to 1.08 rps, which gives 73.4% improvement (Table 1).

5. Conclusions

In the paper a new neural-based friction identification and compensation scheme is proposed. The neural network structure was designed to obtain the qualitative features of friction characteristics based on empirical knowledge ("gray box" approach). In general, model-based friction compensation algorithms outperform the common linear PID control substantially. It should be noted, however, that practical friction modelling requires a very difficult and complex identification process, often infeasible in the industrial environment. The proposed friction compensation with a neural friction model allows for friction identification through training with experimental data that can be repetitively collected at the time of the desired task trajectory change or payload replacement (e.g. tool). The neural network allowed one to model friction on-line with modest requirements for the training data. The neural controller design and prototyping was very effective thanks to the applied mechatronic design methodology and DSP hardware. The laboratory experiments confirmed the suitability of the proposed approach.

It should be noted that the friction characteristics embedded into the network might not hold for any friction interface. In particular, viscous friction is modelled as a linear function of velocity, which does not apply to dry friction (e.g. friction gearbox). It is als assumed that neural friction model parameters do not change with the direction of motion i.e. the static friction characteristics is indeed odd-symmetrical. Furthermore, the proposed neural model does not capture the dynamics of frictional forces (e.g. hysteresis, static friction change). The using of static friction characteristics (i.e. without dynamics) might lead in certain cases to wrong results (e.g. with rapid desired velocity changes or velocity pulsation). The proposed controller performance should also be verified in other control tasks (e.g. low- velocity tracking) to prove its robustness. These problems will be addressed in the frame of the future research.

Refernces

1. ARMSTRONG-HELOUVRY B., DUPONT P., CANUDAS DE WIT C., 1994, A survey of models, analysis tools and compensation methods for the control of machines with friction, *Automatica*, **30**, 7

2. CILIZ M., TOMIZUKA M., 1995, Modeling and compensation of frictional uncertainties in robot control: a neural network-based approach, *Proc. ACC*, FA13, 3269-3273

3. DEWEERTH S., NIELSEN L., MEAD C.A., ASTROM K.J., 1991, A simple neuron servo, *IEEE Trans. on Neural Networks*, **2**, 2, 248-251

4. DU H., NAIR S., 1999, Modeling and compensation of low-velocity friction with bounds, *IEEE Trans. on Control Syst. Techn.*, **7**, 1, 110-121

5. FUNAHASHI K., 1989, On the approximate realisation of continuous mapping by neural networks, *Neural Networks*, **2**, 1

6. HUNT K.J., SABRARO D., ZBIKOWSKI R., 1992, Neural networks for control system – a survey, *Automatica*, **28**, 6,1083-1112

7. KAWATO M., WADA Y., 1993, A neural network model for arm trajectory forming using forward and inverse dynamics model, *Neural networks*, **6**, 7, 919-933

8. KHEMAISSIA S., MORRIS A.S., 19993, Neuro-adaptive control of robotic manipulators, *Robotica*, **11**, 5, 465-473

9. KORENDO Z., 1999, Zastosowanie sieci neuronowych i regulatorów rozmytych do sterowania układami mechanicznymi – podejście mechatroniczne, Doctoral Dissertation, Dept. Robotics and Machine Dynamics, UMM

10. LARSEN G., CETINKUNT S., DONMEZ A., 1995, CMAC neural network control for high precision motion control in the presence of large friction, *Journal of Dynamic Systems, Measurement and Control*, **117**, 415-420

11. UHL T., SZYMKAT M., 1992, A comparison of the classical and neural-based approach to control of manipulation robots, *XVII ISMA, Leuven*, 255-284

12. UHL T., SZYMKAT M., BOJKO T., KORENDO Z., 1994, Application of neural networks for robotic manipulators control – mechatronic approach, Copernicus Proj. CP93-10119 Tech. Rep.

13. YANG S., TOMIZUKA M., 1988, Adaptive pulse width control for precise positioning under the influence of stiction and Coulomb friction, *Journal of Dynamic Systems, Measurement and Control*, Sept., **110**

Projektowanie struktury sieci neuronowej dla celów eliminacji tarcia w napędach robotów

Streszczenie

W pracy przedstawiono oparty na sieciach neuronowych układ sterowania napędem robota. Przedstawiono proces projektowania i prototypowania oparty na podejściu mechatronicznym. Sieć neuronowa w proponowanym rozwiązaniu spełnia rolę lineryzującej pętli sprzężenia zwrotnego. Jej podstawowym zadaniem jest kompensacja wpływu tarcia w napędzie robota. Zaproponowano specjalizowaną architekturę sieci neuronowej dostosowaną do modelowania tarcia. Uczenie sieci odbywa się na podstawie danych eksperymentalnych. Zaproponowaną sieć neuronową zaimplementowano z zastosowaniem techniki szybkiego prototypowania z wykorzystaniem procesorów sygnałowych. Wyniki symulacji porównano z wynikami eksperymentu na rzeczywistym obiekcie. Przedstawione podejście, jak wykazały uzyskane rezultaty, daje dobre wyniki w zakresie linearyzacji układów sterowania robotami z uwzględnieniem tarcia.

Manuscript received July 30, 2001; accepted for print April 3, 2002