BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering and Communication

Department of Control and Instrumentation

Ing. Kamil Švancara

Adaptive Optimal Controller with Identification Based on Neural Networks

Adaptivní optimální regulátor s identifikací založenou na neuronových sítích

Short Version of Ph.D. Thesis

Study Field: Cybernetics, Control and Measurements

Supervisor: Prof. Ing. Petr Pivoňka, CSc.

Opponent: Prof. Ing. Jan Štecha, CSc.

Opponent: Prof. Ing. Mikuláš Alexík, CSc.

Presentation Date: 21 June 2004

Keywords: Adaptive control, A/D converters, Delta models, D/A converters, Identification algorithms, Linear quadratic regulators, MATLAB/Simulink, Neural networks, Programable logic controllers, Quantization errors, Real-time operating systems, Recursive algorithms, Sampling periods.

Klíčová slova: *adaptivní řízení, A/Č převodníky, delta modely, Č/A převodníky, identifikační algoritmy, lineárně kvadratické regulátory, MATLAB/ Simulink, neuronové sítě, programovatelné automaty, kvantizační chyby, operační systémy reálného času, rekurzivní algoritmy, perioda vzorkování.*

Thesis is available at research department of Faculty of Electrical Engineering and Communication, Brno University of Technology, Údolní 53, Brno, 602 00, Czech Republic.

ISBN 80-214-2709-4
ISSN 1213-4198

Contents

1 Introduction

The adaptive control theory follows in time classical control theory. The simple idea to adapt to the new control conditions (process parameters or even structure of the process) leads us to set controller e. g. in the classical way but moreover in the real-time. Naturally, the assumption to use adaptive control scheme brings us many advantages but also some problems.

Generally, the adaptive controller scheme is separable into two parts: identification and controller. It is not novelty that the "core" of each adaptive controller is in the very identification of process. If the identification correctly estimates the real controlled process any stable controller could be used. In this case real-time identification (as the identification is comprehended in adaptive controller) is very important and presented thesis focuses on the identification dealing with practical problems. In presented thesis, the adaptive optimal controller is investigated to fulfill following properties: feasible implementation into industrial controller and the real process control.

First, the overview of identification methods is presented in thesis. Next, several identification methods are implemented with purpose to overcome the quantization effect. The quantization problem arises when the sampling period in the real process control is short. The decreasing of the sampling period is simply provable in the faster controller reaction mainly for the disturbance rejection. Several identification methods working in real-time are investigated in comparison with the identification based on neural networks. The advantage of using of the identification based on neural networks is shown.

Instead of presenting purely simulation results, the real problems in the real process control are investigated. All algorithms are firstly developed in simulation environment, i. e. MATLAB/Simulink. The algorithms are verified on the real process (physical models) via new developed real-time communication toolbox afterwards. Finally, new direct implementation of control algorithms into industrial controller (PLC) is presented.

Motivation of Ph.D. Thesis

During my work on adaptive controllers I have realised that **crucial problem** in adaptive control is the quantization effect. Quantization effect is not properly described in the current literature in connection with adaptive control. That is why the thesis begins with quantization effect description (chapter 2). At the end of such description, the aims of the thesis have naturally arisen.

2 Quantization Effect

The quantization effect is more known for example in instrumentation theory or signal processing theory than in control theory. Furthermore, in control theory the phenomenon has been usually disregarded. It is due to the fact that the conditions used in process control allow the quantization effect to be ignored. Nowadays, when the sampling period is demanded to be very short and the requirements for the control precision are higher then before, the quantization effect plays considerable role in the practical control.

State of the Art

The quantization error given by using A/D and D/A converters are more known in instrumentation and measurement theory. Fowler's [21] and Händel et al. [27] papers published in

IEEE Transactions on Instrumentation and Measurement affirmed that. Next, an overview of quantization with historical background is published by Gray and Neuhoff [24].

The relationship between process control theory and quantization effect is investigated by Williamson [47] as the finite word-length considerations. Next, Åström and Wittenmark [5] together with Middleton and Goodwin [36] basically deal with problems with digital process control. Recently, papers [7, 22] deal with digital process control for short sampling period and different resolution of quantizers.

2.1 Quantization Error

The process control of continuous time system and the control of sampled continuous time system are two different fields. It happens that the controller design is created without precise knowledge of sampling, shaping and quantization effect. The A/D and D/A converters are

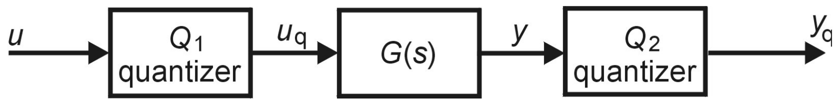

Fig. 1: The real model with A/D and D/A converters represented as quantizers.

necessary parts of each real-time system [21]. The basic feature of the converters is to convert analog signal to discrete values and back (see fig. 1).

$$
\begin{gathered}
y = G(s)u_q \\
\underbrace{y_q = Q_2(y) \quad u_q = Q_1(u)} \\
y_q = Q_2\{G(s)Q_1(u)\}
\end{gathered}
\tag{2.1}
$$

The quantization error e is limited to quantization band $\equiv 1$ LSB. The quantization range Q_{RANGE} and the quantization resolution Q_{RES} are basic parameters for definition of the quantization band. For example, for $Q_{\mathrm{RES}} = 8$ bits we get $2^{Q_{\mathrm{RES}}} = 2^8 = 256$ number of codes. Next, for bipolar converters ± 10 V we get the quantization band $Q_{\mathrm{BAND}} = 10/256 = 39.1$ mV $\approx$ 0.04 V. Therefore the value in finite word-length precision is numerically round off to the three valid places divisible by ≈ 0.04 V.

The quantization error may be modelled as deterministic or stochastic signal in linear analysis. In deterministic model, the error is modeled as constant having the size of quantization errors and with the resolution in the arithmetic calculation. In the stochastic model, the error introduced by rounding or quantization is then described as additive white noise with rectangular distribution (Åström and Wittenmark [5], pp. 479–480). Next, Williamson deals with quantization analysis and shows cases where after linearization the roundoff quantization error is uncorrelated with the quantizer input ([47], pp. 209–211).

Simple results where previous mentioned conclusions are not applicable [56] are presented in this section. Let us consider the modelling of quantizer. The model can be built from quantization effect description to show the disturbance properties of quantization effect. The model can be seen in fig. 2 where the linear part of value u_L is disturbed by non-linear part represented as quantization error e. This point of view is very simple, given from description of quantization effect and it gives us the beginning point for explanation of quantization effect.

The quantization error e **is not independent** from quantizer input u and hence cannot be treated as the independent additive noise [56]. Next, the quantization error cannot be treated as the Gaussian or even white noise because it is directly derived from quantizer input. It means

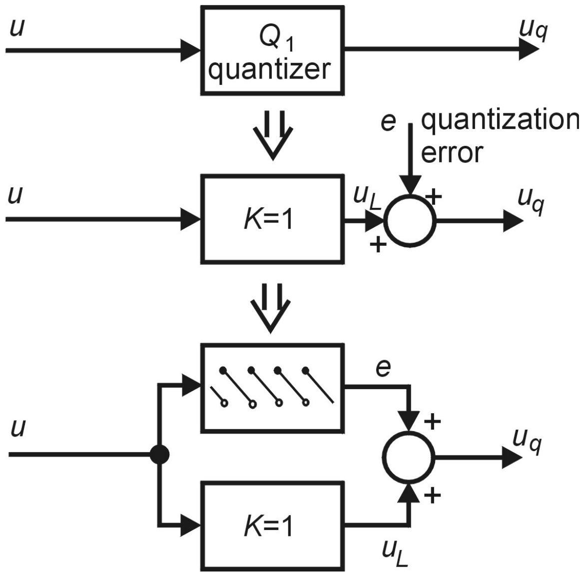

Fig. 2: Principal model of quantization effect.

that the noise is deterministic and it can be predicted. For example, the error is influenced more when the amplitude of quantizer input is smaller.

2.2 Amplitude Shape

The amplitude shape of transformation from continuous time into discrete time could be described mathematically (2.2) [5, 29]

$$\begin{aligned} u^*(t) &= u(t) \cdot m(t) \\ m(t) &= \sum_{k=-\infty}^{\infty} \delta(t - kT_\mathrm{S}) \end{aligned} \tag{2.2}$$

where $m(t)$ is modulation function of Dirac impulses $\delta(\cdot)$.

Sampler is usually followed by shape filter, very often represented by Zero-Order-Hold (ZOH) filter (2.3)

$$G_\mathrm{ZOH}(s) = \frac{1}{s} - \frac{1}{s}\exp(-sT_\mathrm{S}) = \frac{1 - \exp(-sT_\mathrm{S})}{s} \tag{2.3}$$

where the shape is given by integration of Dirac impulse (with infinitely short time length and with unit integral) in every sampling period kT_S. Then we get unit integral multiplied by amplitude of sampled continuous time signal $u(kT_\mathrm{S})\delta(kT_\mathrm{S})$. This value will be held by integrator until the next Dirac impulse. Of course, the past held values will be integrated out firstly. The sampler could be written in Fourier series (2.4)

$$m(t) = \frac{1}{T_\mathrm{S}}\Big(1 + 2\sum_{k=1}^{\infty}\cos(k\omega_\mathrm{S} t)\Big). \tag{2.4}$$

The model is illustrated in fig. 3.

The amplitude and phase changes due to ZOH filter are **important** fact which can be easily forgotten. For tested process with transfer function

$$G(s) = \frac{1}{(10s+1)(s+1)^2} \tag{2.5}$$

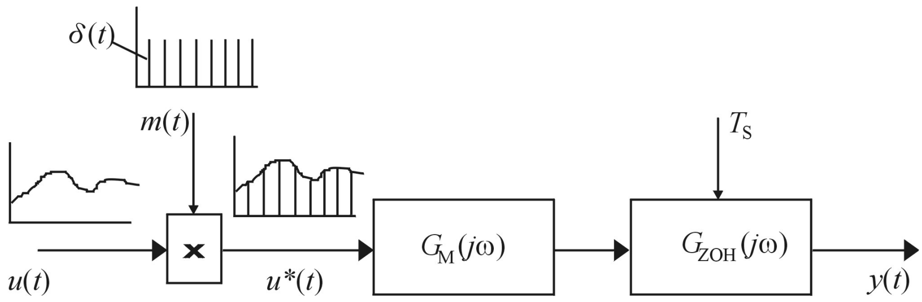

Fig. 3: Model of the sampler and the shape filter.

the Bode diagram (see fig. 4) is solved before and after conversion to discrete domain. The final transfer function after conversion from continuous time domain is

$$G_0(j\omega) = \frac{1}{T_S}\frac{1-\exp(-j\omega T_S)}{j\omega}\frac{1}{(10j\omega+1)(j\omega+1)^2}. \tag{2.6}$$

Let us see what happen if the sampling period has been set ten times longer, i. e. $T_S = 1$ s. The final changes of amplitude and phase are the same in Nyquist frequency because of the different Nyquist frequency. Although the phase is absolutely bigger, i. e. $-360°$ (for $T_S = 0.1$ s) and $-325°$ (for $T_S = 1$ s). The amplitude is again absolutely bigger, i. e. -112 dB (for $T_S = 0.1$ s) and -53 dB (for $T_S = 1$ s).

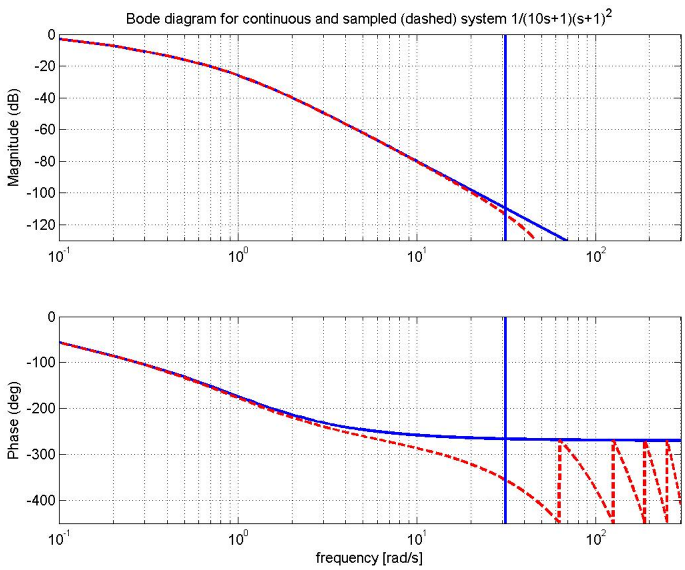

Fig. 4: Bode diagram of continuous and sampled system in dB for sampling period $T_S = 0.1$ s.

In conclusion, the results are related to the problem of the choice of the quantization precision for the set of short sampling period. According to theoretical solution of Signal to Noise Ratio (SNR) [21] for A/D converters, it is interesting to compare the SNR for the chosen quantizer resolution with the drop of the amplitude [56]. In our example, the change of sampling period from $T_S = 1$ s to $T_S = 0.1$ s is expressed in amplitude drop -59 dB in Nyquist frequency. Therefore, the precision of A/D converters should follow the amplitude drop to get the same undisturbed results. In [21], we can find that -62 dB is the theoretical SNR value for resolution $Q_{RES} = 10$ bits. For example if 8 bits A/D converter for the sampling period

$T_S = 1$ s has been chosen as the minimum appropriate resolution, than after the reduction to $T_S = 0.1$ s, the A/D resolution should be increased too.

2.3 Chapter Summary

In this chapter several reasons why the quantization effect must be considered when dealing with digital process control are presented.

1. It is shown when and why the quantization error could not be treated as the independent white or Gaussian noise.
2. It is shown why the definition of persistent excitation signal must be augmented due to quantization effect[1].
3. Next, strong relationship between the setting of sampling period and the choice of quantizers resolution for given process is shown too.

To sum up, the existing digital control theory does not deal enough with the real process control problems: quantization effect is applied when the sampling period is short or the quantizer resolution are not considered at all [22, 23, 44]. That is why the presented thesis is focused on overcoming of quantization effect.

3 The Aims of Ph.D. Thesis

The previous chapter has shown two main problems which are connected. The quantization effect due to A/D and D/A converters is always presented in the digital control process. Its influence increases when the sampling period is short. The setting of sampling period is the second problem. The reason for short sampling period is explained in chapter 7. These two problems together with basic assumptions of adaptive controller lead to determine the main aims of thesis:

- To build an adaptive optimal controller which is suitable for implementation into industrial controller, i. e. controller enables fast and numerically stable real-time solution.
- To improve current optimal controller, i. e. to explain the purpose of installed universal weighted matrix; to extend the universal weighted matrix and to include the integral action into universal weighted matrix.
- To explore briefly identification methods which are generally known with purpose to overcome the quantization effect.
- To show why and where the existing published solution of controller for the setting of short sampling period are unacceptable in real process control.
- To show identification based on neural networks in comparison with other identification methods for the same setup of the controller.
- To explain where and why the identification method based on neural networks is better than other methods for following assumptions: the quantization effect is presented and the sampling period is short.
- To build user comfortable tool which enables us to verify control algorithms before final implementation of algorithm into industrial controller.

[1] This part is completely excluded from an abridged version of Ph.D. Thesis.

4 Identification

In chapter 2 we have worked on issue called quantization effect. To reduce the quantization effect in identification process it is necessary to know all possibilities of identification in closed loop. Therefore, the overview of identification is introduced.

Basically, three adaptive control schemes [4, 33] exist. The *open-loop adaptive control* is the first scheme where the system outputs are not measured and the adaptation comes from the change of an environment around the system only (e. g. gain-scheduling system). Next, *direct adaptive control* assumes the given reference model (closed-loop) and the performance is given according to difference between the reference behaviour and the real behaviour (e. g. Model Reference Adaptive Control–MRAC). In our case, the third scheme will be applied. *Indirect adaptive control* allows firstly real-time estimation (identification) of the process and then the controller is solved in real-time according to identified model of the real process.

State of the Art

Recursive Least Square method (RLS) is well-known identification method. Therefore, it is a part of many published books, for example in Åström and Wittenmark's *Adaptive Control* [4]. Besides this book, Åström and Wittenmark's *Computer–Controlled Systems* [5] holds forth on theory and design of process identification. Ljung's *System Identification* [34] is the basic book about classic identification theory for user. Next, Bobál et al. [13] and Horáček [29] present practical aspects of identification in closed loop. Numerical stability of the algorithm is presented in Böhm [14].

The aim of thesis is also to present identification based on neural networks. Haykin's *Neural Networks* [26] shows a comprehensive foundation of neural networks and their application. Modern view of neural networks as self-optimizing non-linear model is given by Cichocki and Unbehauen [17]. Mathematical properties of neural networks are presented in Fine's book [20]. Narendra presents in his work [37, 38] neural networks for control theory and practise and Chen and Narendra [16] present adaptive control using neural networks.

4.1 Unconventional Overview of Identification Methods

Linear and even nonlinear black-box identification can be divided into three elements [56]:

- model structure of identified process,
- regression vector of observed data,
- and algorithm for minimization.

4.1.1 Model Structure

The model structure should be chosen according to the observed system. From linear point of view, the structures of model are called: FIR (Finite Impuls Response), ARMA (Auto-Regressive Moving Average model), ARX (Auto-Regressive model with eXogenous input–from econometrics), ARMAX (Auto-Regressive Moving Average model with eXogenous input), OE (Output Error model) etc. All of them are built from generally known formula [34, 41]

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(k). \tag{4.1}$$

The state-space (SS) representation is also taken as a different structure model which is powerful for its general MIMO definition.

In nonlinear case, the structures are called NFIR, NARX, NOE, NARMA, NBJ or nonlinear state-space representation where "N" generally means nonlinear model.

The new group of above mentioned structures represents the improved structure (the numerical precision). It is always given by new operator which comprises the linear combination of previous operators. The best example is given by [30] where the commonly used q time-shift operator and $\mathcal{Z}$-transform operator z (which represents complex value and is used in frequency domain $z = \exp(jw)$) are linearly combined into new δ and γ operators in δ-model domain in way:

$$\boxed{\delta = \frac{q-1}{T_{\mathrm{S}}}, \quad \gamma = \frac{z-1}{T_{\mathrm{S}}}.} \tag{4.2}$$

4.1.2 Regression Vector

The regression vector $\boldsymbol{\varphi}$ is inseparable part of the model structure but we can look at it as a new part which brings us a possibility of choice. For example the difference between ARX and OE model is just in the difference treatment of data representation. The ARX model uses past measured output data $\mathbf{y}^k$ while the OE model uses estimated output data $\hat{\mathbf{y}}^k$. Generally speaking, we can treat the data as we want to build another model that is not named yet. For example the past output and input can be filtered as in the δ-model domain. Next example is given in CLOE (Closed Loop Output Error) identification method where both estimated inputs $\hat{\mathbf{u}}^k$ and outputs $\hat{\mathbf{y}}^k$ are used and the criterion minimizes squared error between measured $y(k+1)$ and estimated $\hat{y}(k+1)$ output in closed loop.

4.1.3 Algorithm

The algorithm is used to minimize the criterion. It is the last option we have. That means any of presented linear or non-linear structures can be minimized by many algorithms (the exception is possible). The criterion is usually given as the quadratic (L_2 norm) but can be either L_1 norm or L_∞ as well [42]

$$V(\boldsymbol{\theta}) = E\,\|y(k+1) - \hat{g}(\boldsymbol{\varphi}(k), \boldsymbol{\theta}(k))\|^2 \tag{4.3}$$

where the expectation operator E is overwritten for practical purposes as the sample mean (stationary process)

$$V(\boldsymbol{\theta}) = lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \|y(k+1) - \hat{g}(\boldsymbol{\varphi}(k), \boldsymbol{\theta}(k))\|^2 \tag{4.4}$$

and the "best" parameter vector is given by

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} V(\boldsymbol{\theta}). \tag{4.5}$$

The generalization of every minimization algorithm is given in next iterative equation which is basically suggested in [17]

$$\boxed{\mathbf{w}(k+1) = \mathbf{w}(k) + \eta(k)\mathbf{d}(k+1)} \tag{4.6}$$

where new updated parameters $\mathbf{w}(k+1)$ in step $(k+1)$ are influenced by past parameters $\mathbf{w}(k)$ in previous step of iteration and the direction vector of minimization $\mathbf{d}(k+1)$ in length given by its learning rate $\eta(k)$.

Recursive Least Square Method

The RLS method is well known for identification but we do not need to look at this as the unique method for minimization. It is useful to see the comparison between accelerated LMS and RLS [17]. We can rewrite the modification of general minimization algorithm in eq. (4.6) to RLS version with exponential weighting λ_e, i. e.,

$$\boxed{\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \eta(k)e(k+1)\mathbf{P}(k)\boldsymbol{\varphi}(k+1)} \tag{4.7}$$

where we can see that the learning rate parameter $\eta(k)$ and convergence matrix $\mathbf{P}(k)$ are now time-variant

$$\begin{aligned} \mathbf{P}(k+1) &= \frac{1}{\lambda_e}\left[\mathbf{P}(k) - \frac{\mathbf{P}(k)\boldsymbol{\varphi}(k+1)\boldsymbol{\varphi}^T(k+1)\mathbf{P}(k)}{\lambda_e + \boldsymbol{\varphi}^T(k+1)\mathbf{P}(k)\boldsymbol{\varphi}(k+1)}\right] \\ \eta(k) &= \left(\lambda_e + \boldsymbol{\varphi}^T(k+1)\mathbf{P}(k)\boldsymbol{\varphi}(k+1)\right)^{-1}. \end{aligned} \tag{4.8}$$

LD-FIL Matrix Decomposition

LD-FIL as a robust algorithm could be used mainly for following attributes: numerically stable algorithm and easy implementation for real-time solution. LD-FIL (lower-diagonal-upper) decomposition algorithm [14] could be used in form as it is illustrated in eq. (4.9)

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ g_{12} & 1 & 0 \\ g_{13} & g_{23} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & g_{12} & g_{13} \\ 0 & 1 & g_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

where $\mathbf{G}$ denotes lower triangular matrix, $\mathbf{G}^T$ denotes upper triangular matrix and $\mathbf{D}$ denotes diagonal matrix. Parameters on the main diagonal mainly influence identification. Matrix $\mathbf{P}$ is given by

$$\begin{aligned} \mathbf{P}(k+1) &= \left[\boldsymbol{\Phi}^T(k)\boldsymbol{\Phi}(k) + \boldsymbol{\varphi}(k+1)\boldsymbol{\varphi}^T(k+1)\right]^{-1} \\ &= \mathbf{G}(k+1)\mathbf{D}(k+1)\mathbf{G}(k+1)^T. \end{aligned} \tag{4.10}$$

Well-known LD-FIL matrix decomposition is derived by lemma for matrix inversion (see eq. (4.11)) $\mathbf{G}(k+1)$ denotes lower-triangular matrix

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \tag{4.11}$$

and then [50, 52]

$$\boxed{\begin{aligned} \mathbf{G}(k+1)\mathbf{D}(k+1)\mathbf{G}(k+1)^T &= \mathbf{GDG}^T - \mathbf{GDG}^T\boldsymbol{\varphi}(1 + \boldsymbol{\varphi}^T \cdots \\ &\quad \mathbf{GDG}^T\boldsymbol{\varphi})^{-1}\boldsymbol{\varphi}^T\mathbf{GDG}^T \\ &= \mathbf{G}\big[\mathbf{D} - \mathbf{DG}^T\boldsymbol{\varphi}(1 + \boldsymbol{\varphi}^T\mathbf{GD} \cdots \\ &\quad \mathbf{G}^T\boldsymbol{\varphi})^{-1}\boldsymbol{\varphi}^T\mathbf{GD}\big]\mathbf{G}^T \\ &= \mathbf{G}\left[\mathbf{D} - \mathbf{Df}_i\mathbf{f}^T\mathbf{D}\frac{1}{1 + \mathbf{f}^T\mathbf{Df}}\right]\mathbf{G}^T \end{aligned}} \tag{4.12}$$

where an auxiliary vector $\mathbf{f}$ is given $\mathbf{f}(k) = \mathbf{G}^T(k)\boldsymbol{\varphi}(k+1)$.

Back-propagation Algorithm

The back-propagation learning algorithm is built on *gradient vector* [50], the vector of the first partial derivatives

$$\mathbf{g}(\mathbf{w}) = \nabla V(\mathbf{w}) = \frac{\partial V(\mathbf{w})}{\partial \mathbf{w}} = \sum_{i=1}^{n} \frac{\partial V}{\partial e_i}\frac{\partial e_i}{\partial w_i} = \sum_{i=1}^{n} e_i \frac{\partial e_i}{\partial w_i} \tag{4.13}$$

where energy $V = \frac{1}{2}\sum e^2$ is the scalar function.

To clarify the search direction [17], let us focus on the first-order Taylor's series approximation (i. e. $f(x) - f(x_0) \cong f'(x_0)(x - x_0)$) of energy function $V(\mathbf{w})$ near the point $\mathbf{w}(k)$ where $k = 0, 1, 2, \ldots$

$$V(\mathbf{w}(k+1)) - V(\mathbf{w}(k)) \cong \big(\mathbf{w}(k+1) - \mathbf{w}(k)\big)^{\mathrm{T}} \nabla V(\mathbf{w}(k)) \tag{4.14}$$

and according to eq. (4.6) and (4.13) we can write

$$\mathbf{g}(\mathbf{w}(k))\big(\mathbf{w}(k+1) - \mathbf{w}(k)\big)^{\mathrm{T}} = \mathbf{g}(\mathbf{w}(k))\big(\eta(k)\mathbf{d}(k+1)\big)^{\mathrm{T}} < 0 \tag{4.15}$$

and this steady descent condition is satisfied in simplest way according to *steepest descent algorithm*

$$\eta(k) > 0, \quad \mathbf{d}(k+1) = -\mathbf{g}(\mathbf{w}(k)). \tag{4.16}$$

Therefore the back-propagation learning algorithm can be summarized as (compare to eq. (4.6))

$$\boxed{\mathbf{w}(k+1) = \mathbf{w}(k) - \eta(k)\mathbf{g}(\mathbf{w}) + \alpha\left(\mathbf{w}(k) - \mathbf{w}(k-1)\right).} \tag{4.17}$$

4.2 Chapter Summary

Chapter 4 can be considered as the most important part of the thesis. The innovative **overview** of all identification methods is discussed (see section 4.1) with the purpose to present all available methods that can overcome the quantization effect. According to innovative view of identification methods, the **contribution** of chapter is given in following description of three modifiable possibilities in black-box identification:

- model structure of identified process,
- regression vector of observed data
- and algorithm for minimization.

Model Structure's Influence on Quantization Effect

The approximation property of nonlinear model based on sigmoidal function (NARX) is known [9, 18, 45]. Mentioned approximation property can be used in the real digital process control where quantizers are always inbuilt. In such real case, the quantizers are not ideal. Narrow Code, Missing Code, Wide Code, Integral nonlinearity ow Hystereze nonlinearity [21] are included to the previously described "ideal" quantization error (see fig. 5).

The smooth approximation property of neural networks is advantageously used because of the **permanent present** of different types of nonlinearities in originally linear processes. Next, Baron [9] pointed out that total number of parameters used in the networks $(h + 2)m$ is considerably smaller than in the classical case (polynomial or spline). Parameter m means the number of the basis functions (nodes) and h is the dimension of the input.

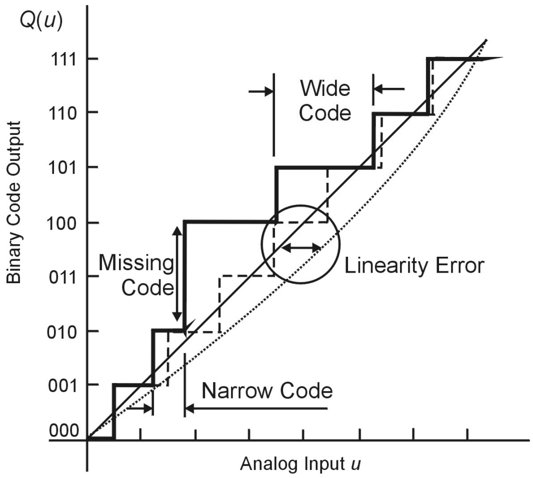

Fig. 5: Differential nonlinearities of quantizers.

Regression Vector's Influence on Quantization Effect

The second possible choice in each identification method is the regression vector [56]. The results in chapter 7 justify the idea that the gradient algorithm applied to NARX structure of neural networks works better than RLS algorithm applied to ARX structure in q time-shift or δ-model domain. The reason can be theoretically explained: δ-model are built to overcome finite word-length precision of used (saved) variables in controller but quantizers decrease the precision more. It is a fact that values of parameters with finite mantisa and exponent are easily saved when converging to zero (δ-model domain) than to one (q time-shift domain). The input-output round off error (due to A/D and D/A converters) cannot be overcame sufficiently in q time-shift domain and in δ-model domain either.

Algorithm's Influence on Quantization Effect

This section explains the difference between performance of two types of iterative minimization algorithms. Generally, we can conclude that two rates of algorithms exist [56]:

- **quadratic rate** based on Newton method where the Gauss-Newton algorithm and RLS algorithm belong;
- **gradient based** called also steepest descent method where the back-propagation algorithm and conjugate gradient algorithm belong.

No other basic methods are applicable and spread. The choice of direction of quadratic based and gradient based algorithm influences the rate how quickly the minimum is reached.

Let us consider the **rate of convergence**. The huge steps into the point where the minimum is expected surely spare time for solution. For example RLS algorithm automatically solves both the learning rate and the direction vector of minimization. The relationship with Newton algorithm is given in Hessian matrix which is solved for RLS according to well-known lemma for the matrix inversion (see eq. (4.11)). The problems arise when the minimization performance (loss function) is disturbed by noise and the minimum is unclear or even not unique. Due to disturbed data (but not by Gaussian or white noise) we cannot expect (from the probability theory) that after sufficient time the performance minimum will be reached. Exactly in this case related to the short sampling period and the low quantization resolution, too big steps in minimization which are automatically solved (see RLS algorithm (4.7)) lead to the instability of parameters in wider range than gradual controllable minimization steps (see simple gradient method (4.17)).

5 Linear Optimal Controller

The term *optimal* from Latin "optimus" always represents the critical point which is equal to the maximum or minimum. These two states are called the extremes and mathematical methods searching for the extremes are very old. The first notice comes from Aristotle in the fifth century before Christ. He solved isoperimetric problem as the extreme problem. In the year 1696, Johann Bernoulli proposed the brachistochrone problem, which asks what shape a wire must be for a bead to slide from one end to the other (without friction) in the shortest possible time only accelerated by gravity. The brachistochrone problem was one of the earliest problems posed in the calculus of variations. Finally, Pontryagin's maximum principle transfers optimal theory to control theory in 1962.

Thesis is focused on adaptive optimal controller. The objective of optimal control theory is to find *optimal* control rule according to chosen performance (e. g. quadratic or time minimum in followed example) and according to given conditions (e. g. fixed points solution). The *best* solution (control rule) is the permanent advantage of optimal control theory. Any other solution must be worse or mostly the same for the same conditions and chosen performance.

The mathematics closed the optimal theory in the late 60s when the Moon was objective of cosmic flights (Sputnik and Apollo projects) but many problems remain unsolved. Presented thesis shows linear optimal controller, i. e. LQ controller in adaptive version. LQ controller is augmented and improved in quadratic performance via its weight matrix. LQ controller is designed to achieve the aim at its possibility of direct implementation into industrial controller (e. g. Programmable Logic Controller–PLC).

State of the Art

The linear optimal theory is the fundamental part of many books. Anderson and Moore's *Optimal Control* [1] could be called the basic book. The optimal control as complete theoretic analysis and synthesis is generally presented there. Mathematical optimal theory and short historical summary is presented in Krupková [31]. Optimal control as a part of modern control theory is presented in Štecha and Havlena's textbooks [25, 43]. Bellman's dynamic programming is used to clearly derive LQ and LQG controller in Andersen and Stoustrup's lecture notes [2]. Camacho [15] presents optimal control theory and its industrial application.

The aim of thesis is to present an adaptive version of linear optimal controller where Åström and Wittenmark's *Adaptive Control* [4] is the basic book. The standard principles and structures of adaptive control are presented there. Kubík et al. [32] presents optimal and adaptive systems. Next, practical aspects of self-tuning controllers are presented in Bobál et al. [13] and practical process control is given by Pivoňka [39].

5.1 Adaptive LQ Controller

Adaptive LQ controller is solved according to identified ARX (NARX) model in each step and according to minimization of the quadratic performance [13, 48]. Identification ensures adaptation in the real time. Quadratic performance is defined by

$$J = \mathbf{x}^{\mathrm{T}}(N)\mathbf{Q}\mathbf{x}(N) + \sum_{k=0}^{N-1} q_y\big(w(k) - y(k)\big)^2 + q_u\big(u(k) - u_0(k)\big)^2 \tag{5.1}$$

where $w(k)$ denotes desired value, $y(k)$ denotes output of the process, $u(k)$ denotes action value, $u_0(k)$ denotes action value for offset elimination and it is equal to desired value. Parameter q_y (q_u) denotes weight for process output (input), $k = 0$ denotes the first step while the minimization is used and $\mathbf{x}^{\mathrm{T}}(N)\mathbf{Q}\mathbf{x}(N)$ denotes the minimum at the last step N.

5.2 Universal Weight Matrix

The quadratic performance can be rewritten into more suitable form [49, 50, 53]

$$J = \sum_{k=0}^{N} \mathbf{z}^{\mathrm{T}}(k)\mathbf{Q}\mathbf{z}(k) \tag{5.2}$$

where $\mathbf{z}^{\mathrm{T}}(k) = \mathbf{S}(k)[u(k), \boldsymbol{\varphi}(k-1), w(k), u_0(k)]$. Next, $\mathbf{z}^{\mathrm{T}}(k) = \mathbf{S}(k)\mathbf{z}(k-1)$. Weight matrix $\mathbf{Q}$ is more universal. The weight matrix (5.5) is implemented to the quadratic performance in equations (5.9), (5.13) and (5.15). This means the matrix can accomplish the quadratic performance or even an incremental weighting and an integral action. We will work with pseudo-state matrix $\mathbf{S} = [\mathbf{S}_u, \mathbf{S}_{\boldsymbol{\varphi}}, \mathbf{S}_w, \mathbf{S}_{u_0}]$ defined by equations (5.3) and (5.4)

$$\begin{aligned} \mathbf{S}_u &= \begin{bmatrix} 1 & 0 & 0 & b_0 & 0 & \dots & 0 \end{bmatrix}^{\mathrm{T}} \\ \mathbf{S}_w &= \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}^{\mathrm{T}} \\ \mathbf{S}_{u_0} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{5.3}$$

and

$$\mathbf{S}_{\boldsymbol{\varphi}} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \dots & 0 & 0 \\ b_1 & \dots & b_r & a_1 & \dots & a_n \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{5.4}$$

Universal weight matrix can be written in many forms to designer's expectation. The idea to use nonstandard state-vector $\mathbf{z}$ allows to build universal quadratic performance. An example of universal weight matrix shows equation (5.5)

$$\boxed{\mathbf{Q} = \begin{bmatrix} q_u & -q_u & 0 & \dots & 0 & \dots & 0 & -q_u \\ -q_u & q_u & 0 & \dots & 0 & \dots & 0 & 0 \\ 0 & 0 & q_u & \dots & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & q_y & 0 & -q_y & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & q_y & -q_y & 0 \\ 0 & 0 & 0 & \dots & -q_y & -q_y & q_y & 0 \\ -q_u & 0 & 0 & \dots & 0 & \dots & 0 & q_u \end{bmatrix}.} \tag{5.5}$$

In this example, the incremental weighting of the input and output is included as well. This solution leads to smoother reaction of both action value (weighted by q_u) and output error (weighted by term q_y). Finally, the mutual ratio between q_y and q_u decides according to designer demands between fast controller reaction and smooth action value.

5.3 Integral Action Implementation

It is well-known that integral action is not included in original definition of linear optimal controller. That is why the integral action is basically solved as parallel action to basic action of controller. Integral is defined as

$$\begin{aligned} u_{\mathrm{i}}(t) &= \int_0^t e(\tau)d\tau \\ u_{\mathrm{i}}(q) &= \frac{1}{1-q^{-1}}e(q). \end{aligned} \tag{5.6}$$

The solution is given in next equation

$$u_{\mathrm{i}}(k) = e(k) + u_{\mathrm{i}}(k-1) \tag{5.7}$$

which can be included into the quadratic performance

$$\begin{bmatrix} u(k) \\ u(k-1) \\ u(k-2) \\ y(k) \\ y(k-1) \\ y(k-2) \\ u_i(k) \\ w(k) \\ u_0(k) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} q_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -q_u \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_y & 0 & 0 & 0 & -q_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q_{\mathrm{i}} & 0 & 0 \\ 0 & 0 & 0 & -q_y & 0 & 0 & 0 & q_y & 0 \\ -q_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_u \end{bmatrix} \begin{bmatrix} u(k) \\ u(k-1) \\ u(k-2) \\ y(k) \\ y(k-1) \\ y(k-2) \\ u_i(k) \\ w(k) \\ u_0(k) \end{bmatrix} \tag{5.8}$$

where we weighted integral action by term $\boxed{q_{\mathrm{i}} u_{\mathrm{i}}^2}$.

5.4 LD-FIL Application

The method for the minimization in the each step ahead is known and it is described in next equations [49, 51, 50]. The performance is given by

$$\begin{aligned} J(N-1) &= \mathbf{z}^{\mathrm{T}}(N-1)\mathbf{S}^{\mathrm{T}}\mathbf{QSz}(N-1) \\ &= \mathbf{z}^{\mathrm{T}}(N-1)\mathbf{Hz}(N-1). \end{aligned} \tag{5.9}$$

We can simplify the vector $\mathbf{z}$ for this moment in form $\mathbf{z}^{\mathrm{T}} = \mathbf{S}[u, \boldsymbol{\varphi}]$. Hence,

$$\begin{aligned} J(N-1) &= \left[u, \boldsymbol{\varphi}\right] \begin{bmatrix} H_{uu} & \mathbf{H}_{u\varphi} \\ \mathbf{H}_{\varphi u} & \mathbf{H}_{\varphi\varphi} \end{bmatrix} \begin{bmatrix} u^{\mathrm{T}} \\ \boldsymbol{\varphi}^{\mathrm{T}} \end{bmatrix} \\ &= \left[uH_{uu} + \boldsymbol{\varphi}\mathbf{H}_{\varphi u}, u\mathbf{H}_{u\varphi} + \boldsymbol{\varphi}\mathbf{H}_{\varphi\varphi}\right] \begin{bmatrix} u^{\mathrm{T}} \\ \boldsymbol{\varphi}^{\mathrm{T}} \end{bmatrix} \\ &= uH_{uu}u^{\mathrm{T}} + \boldsymbol{\varphi}\mathbf{H}_{\varphi u}u^{\mathrm{T}} + u\mathbf{H}_{u\varphi}\boldsymbol{\varphi}^{\mathrm{T}} + \boldsymbol{\varphi}\mathbf{H}_{\varphi\varphi}\boldsymbol{\varphi}^{\mathrm{T}} \\ &= uH_{uu}u^{\mathrm{T}} + 2u\mathbf{H}_{u\varphi}\boldsymbol{\varphi}^{\mathrm{T}} + \boldsymbol{\varphi}\mathbf{H}_{\varphi\varphi}\boldsymbol{\varphi}^{\mathrm{T}}. \end{aligned} \tag{5.10}$$

Next, derivative of $J(N-1)$ with respect to u is

$$\frac{\partial J(N-1)}{\partial u} = 2H_{uu}u^{\mathrm{T}} + 2\mathbf{H}_{u\varphi}\boldsymbol{\varphi}^{\mathrm{T}} = 0. \tag{5.11}$$

Tab. 1: LQ Controller Solved in Iteration Method

Step	Equation	Notes
1.	$\mathbf{H}^* = \mathbf{H}_{\varphi\varphi} - \mathbf{H}_{u\varphi}^{\mathrm{T}} H_{uu}^{-1} \mathbf{H}_{u\varphi}$	recursively solves lost function
2.	$\mathbf{GDG}^{\mathrm{T}} = \mathbf{H}$	LD-FIL decomposition
3.	$u(k) = -G_{uu}^{-1} \mathbf{G}_{u\varphi} \boldsymbol{\varphi}(k-1)$	solves action value

The action value is solved by equation

$$u^{\mathrm{T}} = -H_{uu}^{-1} \mathbf{H}_{u\varphi} \boldsymbol{\varphi}^{\mathrm{T}} = u \tag{5.12}$$

which we can use in previous equation (see eq. (5.10)).

Hence,

$$min[\Psi(N-1)] = \bar{\mathbf{z}}^{\mathrm{T}}(N-1)\big(\mathbf{H}_{\varphi\varphi} - \mathbf{H}_{u\varphi}^{\mathrm{T}} H_{uu}^{-1} \mathbf{H}_{u\varphi}\big)\bar{\mathbf{z}}(N-1) \tag{5.13}$$

where $\bar{\mathbf{z}}^{\mathrm{T}}(N-1)$ is $\mathbf{z}^{\mathrm{T}}(N-1)$ without $u(N-1)$ and

$$\mathbf{H} = \begin{bmatrix} H_{uu} & \mathbf{H}_{u\varphi} \\ \mathbf{H}_{\varphi u} & \mathbf{H}_{\varphi\varphi} \end{bmatrix}. \tag{5.14}$$

We can simply see that $\mathbf{H}$ is the symmetric matrix and consecutively the next minimization step $(N-1)$ is defined by

$$\begin{aligned} min[\Psi(N-1)] = {} & \bar{\mathbf{z}}^{\mathrm{T}}(N-1)\mathbf{H}^*\bar{\mathbf{z}}(N-1) \\ & + \mathbf{z}^{\mathrm{T}}(N-1)\mathbf{Q}\mathbf{z}(N-1) \end{aligned} \tag{5.15}$$

where matrix $\mathbf{H}^* = \mathbf{H}_{\varphi\varphi} - \mathbf{H}_{u\varphi}^{\mathrm{T}} H_{uu}^{-1} \mathbf{H}_{u\varphi}$ is defined at step (N). Using LD-FIL decomposition (see (4.9)) for matrix $\mathbf{G}$ instead of $\mathbf{H}$, where $\mathbf{H} = \mathbf{GDG}^{\mathrm{T}}$. We can rewrite the quadratic performance to the triangular factor quadratic norm

$$\|\mathbf{G}[u(k), \boldsymbol{\varphi}(k-1), w(k), u_0(k)]^{\mathrm{T}}\|^2. \tag{5.16}$$

Now, it is simple to find control law $u(k)$ with influences on the first row only of the minimization at step k

$$G_{uu}u(k) + \mathbf{G}_{u\varphi}\boldsymbol{\varphi}(k-1) = 0 \tag{5.17}$$

where the minimum is given $\|\mathbf{G}_{\varphi\varphi}\boldsymbol{\varphi}(k-1)\|^2$ and G_{uu}, $\mathbf{G}_{u\varphi}$ and $\mathbf{G}_{\varphi\varphi}$ are sub-matrices of $\mathbf{G}$. Finally, the control law is given by

$$\boxed{u(k) = -G_{uu}^{-1}\mathbf{G}_{u\varphi}\boldsymbol{\varphi}(k-1).} \tag{5.18}$$

LQ is solved at the each one step ahead. Tab. 1 shows recursively solved LQ iteration method.

5.5 Chapter Summary

Chapter Linear Optimal Controller brings us the improved LQ controller which satisfies given aims (see chapter 3). The **contribution** of this chapter is as follows:

1. Improved LQ controller is presented which is able to work in the real-time (section 5.4). Deterministic time of its solution in each step is short. That is why it can be implemented in industrial controller.

2. The universal matrix is augmented by incremental weighted inputs and outputs (section 5.2). Next, new solution of integral action is presented (section 5.3).

To sum up, the existing linear optimal controller is rebuilt with the purpose to use it in industrial controller. The same setup of the controller is necessary assumption for correct comparison of investigated identification methods.

6 MATLAB Toolbox

The implementation of the control algorithms from the simulation environment into industrial controller is necessary to solve effectively in relation to minimization of implementation errors. The capability of algorithm transmission into the industrial controller is important and transmission time should be minimal. Is it obvious that the control algorithms are designed and developed in simulation environment first and they are implemented into the industrial controllers in the real production processes afterwards.

The object of this chapter is an analysis of possibilities of the control algorithm direct implementation from MATLAB, i. e. today's most applicable simulation environment in automation, into PLC, i. e. programmable logic controller.

The advantages of the direct implementation of control algorithms are given first of all by the minimum of implementation errors in comparison with the indirect implementation. The development of heterogenous control algorithms becomes faster. The choice of optimal control algorithm is relatively simple due to its proven functionality in the real processes.

State of the Art

The new MATLAB/Simulink toolbox of direct implementation is one of practical benefits of presented thesis. The handbooks of direct implementation can be found in works of Pivoňka [39], Middleton and Goodwin [36] and Horáček [29].

The MathWorks' *Real-Time Workshop* and *Real-Time Windows Target* [35] present the standard inbuilt real-time communication between the simulation environment and the real process. The basic principles in the real-time systems are presented by Årzén course [3].

Programmable Logic Controller in use and its inbuilt communication protocols are published in B&R help [8].

6.1 Basic Description of Solution

The real-time communication between MATLAB/Simulink and the real process controlled by PLC is aimed at time decreasing required for the transmission of algorithm from simulation environment to the control of the real process. At the beginning, the control algorithm is developed in simulation environment, e. g. MATLAB. Afterwards the control algorithm is directly tested on the real process [28].

The direct implementation of control algorithms is based on three steps [54, 57] which are mutually connected and form explicit way how to implement any control algorithm from the simulation environment into industrial controllers. General view on direct implementation is presented in three steps as fig. 6 shows.

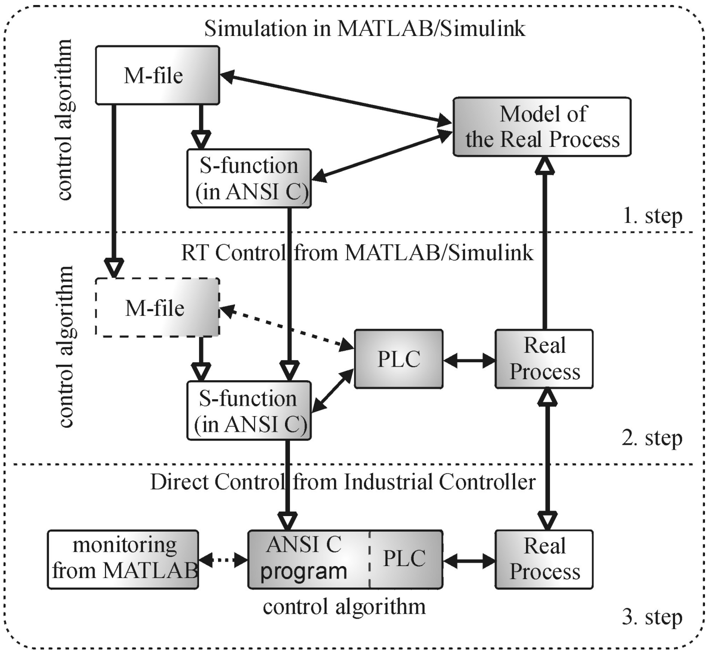

Fig. 6: The scheme of complex solution of three steps of direct implementation from MATLAB/Simulink into PLC.

6.2 New Real-Time Toolbox

The real-time communication between simulation environment and real process is an advantageous tool. It is important to have in mind that one nondeterministic cycle in communication should not stop or crash the whole operating system only for the fact that soft real-time system is used [3]. The suggested real-time algorithm is shown in fig. 7.

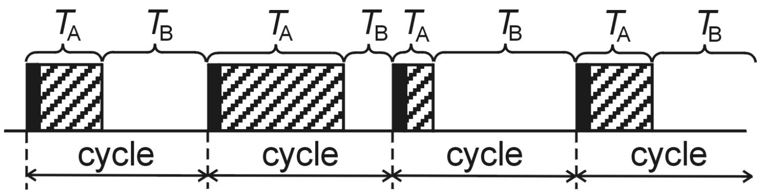

Fig. 7: Control sequence of implemented real-time algorithm.

The synchronization is given from PC-CPU clock. The algorithm sampling period T_S is equal to the cycle time. Each cycle is sum of two times T_A and T_B. Time T_A represents algorithm reaction time and time T_B represents the rest of cycle time. Algorithm 1 shows how the algorithm could be implemented to the system and describes algorithm reaction time T_A in details.

6.3 Chapter Summary

The presented chapter deals with possibilities of complex algorithm implementation from simulation environment MATLAB/Simulink into industrial controller–PLC. The **contribution** can be seen in suggested solution of direct implementation which effectively spare time needed for control algorithm implementation. It also minimizes eventuality of implementation errors. The **new real-time communication toolbox** enables us to verify very quickly many diffe-

Algorithm 1 Real-time communication ($T_A = T_1 + T_2 + T_3 + T_4 + T_5$)

1: to read process output from PLC (T_1)
2: to transfer process output to MATLAB/Simulink (T_2)
3: to solve control algorithm action value (process input) in MATLAB/Simulink according to process output and desired value (T_3)
4: to transfer process input from MATLAB/Simulink (T_4)
5: to write process input into PLC (T_5)

rent control algorithms directly on the real process without need of sequential implementation of each algorithm.

The suggested solution is advantageous due to its independence of communication protocol because it consists of many accessible communication protocols including Ethernet, RS 232, CAN or Profibus. These possibilities are given by applied PVI interface made by B&R Automation Company.

This chapter gives information about alternative real-time communication toolbox in environment MATLAB/Simulink instead of using inbuilt Real-Time Workshop. The new toolbox is stable, easy to implement and understandable for users, robust and applicable for verification and testing of complex heterogenous algorithms for technological process control.

7 Commented Results

This chapter shows achieved simulation results (from MATLAB/Simulink ver. 6.5) and real results (from real physical models controlled via PLC B&R CP360–Pentium 266 MHz). The simulation is extended with purpose to be more closer to real process control instead of using "pure" simulation, i. e. the action value is bounded to ±10 V, quantizers given by A/D and D/A converters are added etc. In regulation, the output response is not the only single scale for the measurement of the control quality. That is why the action value in time is shown too. Graphs with parameters of the model in time and loss function in time can be also shown but graphs with process input and output in time testify enough the control quality.

Setting of Short Sampling Period: Firstly, the reason for decreasing of the sampling period is shown in the simulation experiment[2] (published in papers [50, 53]).

Choice of Quantizers Resolution: The idea of decreasing sampling period has already been justified. The problem arises when the simulation is extended with a quantization effect[2] (published in papers [52, 53]).

Using the New Real-Time Toolbox: Ethernet TCP/IP and serial link have been used in laboratory experiment[2] (published in papers [54, 55, 57, 58]).

Real Process Control Results: The comparison of classical RLS identification algorithm of the third order ARX structure with LD-FIL decomposition in q time-shift domain and nonlinear ARX identification (NARX) based on NN for the real process is shown[2] (published in paper [52]). The real digital process control results have proven the simulation results (see paper [50]).

δ-Model ARX Identification versus NN Based Identification: The second possible choice of identification method (in comparison with [50, 52, 53]) is the Recursive Least Square

[2]This part is completely excluded from an abridged version of Ph.D. Thesis.

method (RLS) applied to δ-model of ARX structure [56]. The simulation results have already justified the idea that the gradient algorithm applied to Nonlinear ARX structure (NARX) of neural net have worked better. This conclusion means that δ-model are built to overcome finite word-length precision of used variables in controller only [23, 44, 22]. The values with finite mantisa and exponent are easily stored when converging to zero (δ-model domain) than to one (q time-shift domain). The input-output round off error cannot be overcame sufficiently.

The real results have been obtained for process given by

$$F_{\mathrm{E}}(s) \approx \frac{0.9}{(10.1s+1)(0.9s+1)^2}. \tag{7.1}$$

Fig. 8 and 9 show the real process response and disturbance rejection together with controller action value. The process output (every upper sub-figures) and input (every lower sub-figures) are shown for desired step set to +2 V at time 60 s. Disturbances have disturbed proces all the time. 10 bits quantizers have been used. Exponential weighting has been se to $\lambda_{\mathrm{e}} = 0.95$. LQ controller has been used with incremental weighting matrix Q where parameters have been set to $q_u = 0.005$, $q_y = 1$ and $q_{\mathrm{i}} = 0.1$.

The sub-section presents the comparison of the real process control for NN identification and classical identification where NN identification is better for assumed conditions:

- short sampling period T_{S} to produce faster disturbance rejection and process response;
- highly decreased numerical precision due to used A/D and D/A converters as a necessary part in practical regulation.

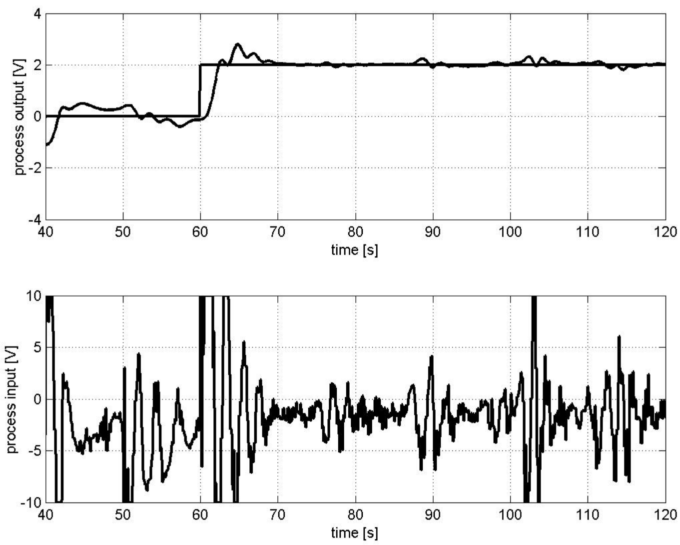

Fig. 8: The real result of closed-loop identification based on δ-model with adaptive LQ controller. The sampling period has been set $T_{\mathrm{S}} = 0.1$ s. A/D and D/A converters have been set to 10 bits.

The reasons why the identification based on neural networks has been significantly successful in the real process control in comparison with simulation control can be explained as follows:

1. in real process control, the several nonlinearities (hystereze nonlinearity, integral nonlinearity, etc.) which have not been included into simulation are better approximated by NARX structure than ARX structure;
2. the classical identification sampling period cannot be too short otherwise identified transfer function loses the correct estimation of the real process.

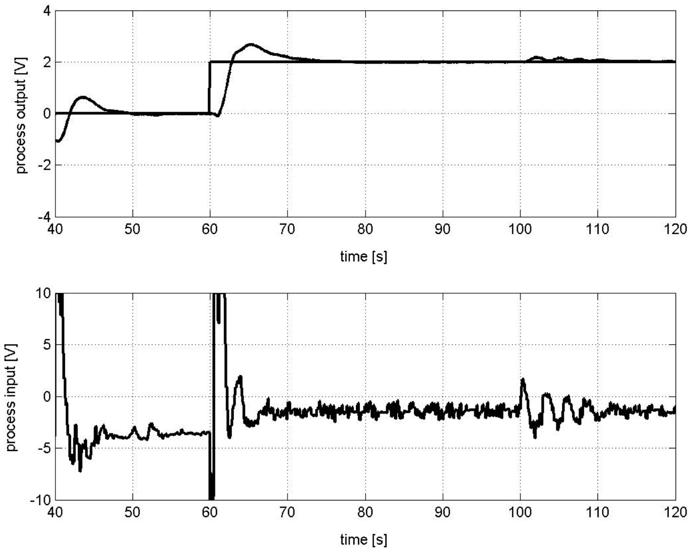

Fig. 9: The real result of closed-loop identification based on NN with adaptive LQ controller. The sampling period has been set $T_S = 0.1$ s. A/D and D/A converters have been set to 10 bits.

8 Conclusion

The presented thesis is motivated by problems which arise from differences between "pure" simulation results and real results after implementation. The same control algorithm is implemented both into simulation environment (MATLAB/Simulnik) and into industrial controller (PLC B&R). The differences are shown in chapter 2 where the choice of quantizers resolution and the setting of sampling period are two options which need to be discussed. Naturally, the reduction of negative influence of quantization effect and objective reasons for short sampling period have lead to the aims of thesis.

Adaptive control scheme is assumed because of reasons mentioned in chapter 5. It comprises the advantage of linear optimal controller with several identification methods. Liner optimal controller gives an optimal solution according to quadratic performance (therefore it is called LQ controller where L means Linear gain and Q means Quadratic performance). In identification, several classical methods are compared with neural network based identification.

The **contribution** of presented thesis is:

- Adaptive LQ controller is suitable for implementation into industrial cont-roller–PLC (i. e. fast and numerically stable real-time solution is ensured due to LD-FIL decomposition). The issue has been published in [48, 49, 51].
- LQ controller is improved by universal weighted matrix **Q** where integral action and incremental weighting are included. This issue has been published in [49, 50].

- New and unconventional overview of identification methods is presented with purpose to overcome the quantization effect. The issue has been published in [56].
- The existing solution of controller for the setting of short sampling period published in recent scientific papers is not longer acceptable. The issue has been published in [52].
- Several comparisons of different identification methods are shown in case where the quantization effect plays an important role. The issue has been published in [50, 52].
- Benefits of using identification based on neural networks instead of other methods are presented for assumptions: the quantization effect is presented and the sampling period is short. The issue has been published in [50, 52, 53].
- New real-time toolbox in MATLAB/Similink enables verifying of control algorithms before its final implementation into industrial controller much faster than before. The issue has been published in [54, 55, 57, 58]. B&R company has been interested in our solution published in paper [54], therefore our paper has been cited in company journal *Automotion* vol. 12, 2003, pp. 11. Next, paper [57] has been cited in Czech journal *Automatizace* vol. 12, 2003, pp. 795.

Thesis aims have been clearly fulfilled.

References & List of Published Papers

[1] Anderson B. D. O. and Moore J. B.: *Optimal Control, Linear Quadratic Methods.* Prentice Hall, Inc. 1990, ISBN 0-13-638560-5.

[2] Andersen P. and Stoustrup J.: *Optimal Control Lecture.* Aalborg University, 2001, *http://www.control.auc.dk/∼pa/kurser/Optimal/index.html*.

[3] Årzén K. E.: *Real-Time Systems: Course material.* Lund Institute of Technology, 2002, *http://www.control.lth.se/∼kurstr/material02.html*.

[4] Åström K. J. and Wittenmark B.: *Adaptive Control.* Addison-Wesley Publ. Co., 3. ed., 1995, ISBN 0-201-55866-1.

[5] Åström K. J. and Wittenmark B.: *Computer-Controlled Systems, Theory and Design.* Prentice Hall, Inc. 3.ed., 1997, ISBN 0-13-314899-8.

[6] Balda P., Schlegel M. a Štětina M.: New Control System REX Supports the Simulation in MATLAB/Simulink Environment (Nový řídicí sytém REX podporuje simulaci v prostředí Matlab/Simulink). *Automatizace*, 2003, vol. 2, pp. 100–103, in Czech, ISSN 0005-125X.

[7] Bamieh B.: Intersample and Finite wordlength Effects in Sampled-Data Problems. In *Proc. of the 35th IEEE Conference on Decision and Control*, Kobe (Japan), 1996, ISBN 0-7803-3590-2.

[8] B&R Documantation: Automation Studio Help. Automation Net, PVI. 1997–2002, *http://www.br-automation.cz*.

[9] Barron A. R.: Universal Approximation Bounds for Superpositions of Sigmoidal Function. *IEEE Transactions on Information Theory*, May 1993, vol. 39, no. 3, pp. 930–945, ISSN 0018-9448.

[10] Bílek K.: One toll, many targets—efektivita v automatizaci v podání B&R. *Automatizace*, 2000, vol. 9, pp. 619–621, in Czech, ISSN 0005-125X.

[11] Bílek K.: One toll, many targets—efektivita v automatizaci v podání B&R. *Automatizace*, 2001, vol. 3, pp. 199–200, in Czech, ISSN 0005-125X.

[12] Blaha P. a Václavek P.: Performance Analysis of Serial Port Interface in MATLAB. In *Proceedings of MATLAB Conference*, Praha (Czech Republic), 2002, ISBN 80-7080-500-5, *http://www.humusoft.cz/matlab02/index.htm*.

[13] Bobál V., Böhm J., Prokop R. and Fessl J.: *Practical Aspects of Self-Tuning Controllers, Algorithms and Implementation (Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace)*. VUTIUM, 1999, in Czech, ISBN 80-214-1299-2.

[14] Böhm J.: *Survey in Using Square-roots factors (Přehled použití odmocninových filtrů)*. Czech Academy of Sciences, UTIA, 1980, no. 1054, in Czech.

[15] Camacho E. F., Berenguel M. and Rubio F. R.: *Adavanced Control of Solar Plants, Advanced in Industrial Control*. SpringerVerlag, 1997, ISBN 3-540-76144-6.

[16] Chen L. and Narendra K. S.: Nonlinear Adaptive Control Using Neural Networks and Multiple Models. *Automatica*, 2001, vol. 37, pp. 1245–1255, ISSN 0005-1098.

[17] Cichocki A. and Unbehauen R.: *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons, Inc. 1993, ISBN 0-471-93010-5.

[18] Cybenko G.: Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 1989, vol. 2, pp. 303–314, ISSN 0932-4194.

[19] Dušek F. a Honc D.: Using the Serial Link Under MATLAB 6 (Využití sériové linky pod MATLABem verze 6). *Konference MATLAB*, Praha (Czech Republic), 2002, in Czech, ISBN 80-7080-500-5, *http://www.humusoft.cz/matlab02/index.htm*.

[20] Fine T. L.: *Feedforward Neural Network Methodology*. SpringerVerlag, 1999, ISBN 0-387-98745-2.

[21] Fowler K.: Part 7: Analog to Digital Conversion in Real-Time Systems. *IEEE Instrumentation & Measurement Magazine*, September 2003, vol. 6, pp. 58–64, ISSN 1094-6969.

[22] Gessing R.: Whether the Delta Operator Models are Really Better for Small Sampling Periods. In *15th IFAC Triennial World Congress*, Barcelona (Spain), 2002, Paper no. 1426, ISBN 0-08-044221-8.

[23] Guo L. and Tomizuka M.: Parameter Identification with Derivative Shift Operator Paramatrization. *Automatica*, 1999, vol. 35, pp. 1073–1080, ISSN 0005-1098.

[24] Gray R. M. and Neuhoff D. L.: Quantization. *IEEE Transactions on Information Theory*, October 1998, vol. 44, no. 6, pp. 2325–2383, ISSN 0018-9448.

[25] Havlena V. and Štecha J.: *Modern Control Theory (Moderní teorie řízení)*. ČVUT, 2000, in Czech, ISBN 80-01-02095-9, *http://dce.felk.cvut.cz/stecha/mtr.pdf* .

[26] Haykin S.: *Neural Networks, A Comprehensive Foundation*. Mac-Millan College Publishing Company, 1994, ISBN 0-02-352761-7.

[27] Händel P., Skoglund M. and Petterson M.: A calibration scheme for imperfect quantizers. *IEEE Transactions on Instrumentation & Measurement*, October 2000, vol. 49, no. 5, pp. 1063–1068, ISSN 0018-9456.

[28] Hnilička B. and Pivoňka P.: Implementation of Heterogeneous Algorithms from MATLAB-Simulink into PLC. In *Proceedings of 9th Zittau Fuzzy Colloquium*, Zittau (Germany), 2001, pp. 151–156, ISBN 3-9808089-0-4.

[29] Horáček P.: *Systems and Models (Systémy a modely)*. ČVUT, 2001, in Czech, ISBN 80-01-01923-3.

[30] Johanson R.: *System Modelling and Identification*. Prentice Hall, Inc. 1993, ISBN 0-13-482308-7.

[31] Krupková V.: *Optimal Process Theory (Teorie optimálních prosesů).* VUT, 2000, in Czech.

[32] Kubík S., Kotek Z., Razím M., Hrušák J. and Branžovský J.: *Theory of Automatic Control II, Optimal, Adaptive and Self-learning Systems (Teorie automatického řízení II, Optimální, adaptivní a učící se systémy).* SNTL, Praha 1982, in Czech.

[33] Landau I. D., Lozano R. and M'Saad M.: *Adaptive Control.* Springer, 1997, ISBN 3-540-76187-X.

[34] Ljung L.: *System Identification: Theory for the User.* Prentice Hall, Inc. 1987, ISBN 0-13-881640-9.

[35] The MathWorks Inc. Documentation. *Writing S-Functions, Real-Time Workshop User's Guide, Real-Time Windows Target User's Guide.* 1994–2003, *http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml*.

[36] Middleton R. H. and Goodwin G. C.: *Digital Control and Estimation, A Unified Approach.* Prentice Hall, Inc. 1990, ISBN 0-13-211798-3.

[37] Narendra K. S.: Neural Networks for Control: Theory and Practise. *The IEEE*, 1996, vol. 84, no. 10, pp. 1385–1406, ISSN 0018-9219.

[38] Narendra K. S. and Parthasarathy K.: Identification and Control of Dynamic Systems Using Neural Networks. *IEEE Transaction on Neural Networks*, 1990, vol. 1, no. 1, pp. 4–27, ISSN 1045-9227.

[39] Pivoňka P.: *Digital Control (Číslicová řídicí technika).* VUT, 2003, in Czech, E-text AMT 101.

[40] Reisner L.: Real-time System with Win32 API (Operační systémy reálného času s Win32 API). *Automa*, 2002, vol. 3, pp. 64–66, in Czech, ISSN 1210-9592, *http://www.automa.cz*.

[41] Södeström T. and Stoica P.: *System Identification.* Prentice Hall, Inc. 1989, ISBN 0-13-881236-5.

[42] Sjöberg J. et al.: Nonlinear Black-box Modeling in System Identification: a Unified Overview. *Automatica*, 1995, vol. 31, no. 12, pp. 1691–1724, ISSN 0005-1098.

[43] Štecha J.: *Optimal Decision and Control (Optimální rozhodovaní a řízení).* ČVUT, 1999, in Czech, *http://dce.felk.cvut.cz/stecha/orr.pdf*.

[44] Świder Z.: Realization Using the γ-Operator. *Automatica*, 1998, vol. 34, no. 11, pp. 1455–1457, ISSN 0005-1098.

[45] Tikk D., Kóczy L. T. and Gedeon T. D.: A Surwey on Universal Approximation adn Its Limits in Soft Compution Techniques. *International Journal of Approximate Reasoning*, May 2003, vol. 33, no. 2, pp. 185-202, ISSN 0888-613X.

[46] Vychodil H., Pivoňka P. and Krupanský P.: The Algorithm for Choice of Samples in Training Set for Neural Networks. In *Proceedings of 10th Zittau Fuzzy Colloquium*, Zittau (Germany), 2002, pp. 224–229, ISBN 3-9808089-2-0.

[47] Williamson D.: *Digital Control and Implementation: Finite Wordlength Considerations.* Prentice Hall, Inc. 1991, ISBN 0-13-211640-5.

List of Published Papers

Conference Papers

[48] Švancara K.: Adaptive Optimal Controller in MATLAB-Simulink. In *Proceedings of 7th Conference Student FEI*, Brno (Czech Republic), 2001, pp. 56–58, in Czech, ISBN 80-214-1859-1.

[49] Švancara K.: Adaptive Optimal Controller Using Square-root Factors Implemented in PLC. In *Proceedings of 8th Conference Student EEICT*, Brno (Czech Republic), 2002, pp. 99–103, ISBN 80-214-2115-0, *http://www.feec.vutbr.cz/UVEE/STC/*.

[50] Švancara K. and Pivoňka P.: Closed Loop On-line Identification Based on Neural Networks in Adaptive Optimal Controller. In *Proceedings of 10th Zittau Fuzzy Colloquium*, Zittau (Germany), 2002, pp. 218–223, ISBN 3-9808089-2-0.

[51] Švancara K. and Pivoňka P.: Adaptive Optimal Controller Direct Implemented from MATLAB into PLC B&R. In *Proceedings of the 13th International DAAAM Symposium*, Vienna (Austria), 2002, pp. 547–548, ISBN 3-901509-29-1.

[52] Švancara K. and Pivoňka P.: The Issue of Quantization Effect in Direct Implementation of Adaptive LQ Controller with NN Identification into PLC. In *Proceedings of 3rd EUSFLAT Conference*, Zittau (Germany), 2003, pp. 361–366, ISBN 3-9808089-4-7.

[53] Švancara K. and Pivoňka P.: Quantization Effect Influences Identification in Adaptive LQ Controller. In *Proceedings of 11th IEEE Mediterranean Conference on Control and Automation*, Rhodos (Greece), 2003, pp. 77–82, ISBN 960-87706-0-2.

[54] Švancara K. and Pivoňka P.: The Real-Time Communication between MATLAB and The Real Process Controlled by PLC. In *Proceedings of 7th International Research/Expert Conference TMT*, Lloret de Mar (Spain), 2003, pp. 1077–1080, ISBN 9958-617-18-8.

[55] Švancara K. and Pivoňka P.: Possibilities of the Direct Implementation of Control Algorithms from Environment MATLAB/Simulink. In *Proceedings of 2nd WSEAS Int.Conf. on Electronics, Control and Signal Processing*, Singapore, 2003, Paper no. 471-111, ISBN 960-8052-91-2.

[56] Švancara K., Pivoňka P. and Voda A.: The Relationship between the Setting of Sampling Period and the Choice of Quantizers Resolution for Different Identification Methods. In *Proceedings of 12th IEEE Mediterranean Conference on Control and Automation*, Kusadasi (Turkey), Submitted.

Journal Papers

[57] Švancara K. and Pivoňka P.: Přímá implementace řídicích algoritmů z prostředí Matlab/Simulink do PLC. *Automatizace*, 2003, vol. 12, pp. 786–789, in Czech, ISSN 0005-125X.

[58] Švancara K. and Pivoňka P.: Possibilities of the Direct Implementation of Control Algorithms from Environment MATLAB/Simulink. *WSEAS Transactions on Circuits and Systems*, Janurary 2004, vol. 3(1), pp. 7–10, ISSN 1109-2734.

A Curriculum Vitae

Name: Kamil Švancara (Mr)
Date Of birth: 14 March 1978
Nationality: Czech
Marital Status: Single
Address: Pustějov 196, 742 43, Czech Republic
E-mail: kamilsvancara@email.cz
WWW: *http://wes.feec.vutbr.cz/∼svancara//*

Education

1992–96 Secondary Electro-technical School Frenštát p. R., Frenštát pod Radhoštěm, Czech Republic

1996–01 Brno University of Technology (Ing. which is equivalent to MSc. in Electrical Engineering), Department of Control and Instrumentation, Faculty of Electrical Engineering and Communication, Brno, Czech Republic

2001 Aalborg University (5 months), Defended Master Thesis at Department of Control Aalborg, Denmark, Thesis was assigned by Grundfos Company with title: Saturation Detection in Heating Systems, Supervisor: assoc. prof. Palle Andersen (pa@control.auc.dk)

2001-04 Brno University of Technology (Ph.D. in Cybernetics, Control and Measurements), Department of Control and Measurement, Faculty of Electrical Engineering and Communication, Brno, Czech Republic, Supervisor: prof. Petr Pivoňka (pivonka@feec.vutbr.cz)

2003-04 Institut National Polytechnique de Grenoble, (4 months), Laboratoire d'Automatique de Grenoble, Grenoble, France, Supervisor: prof. Alina Voda (Alina.Voda@lag.ensieg.inpg.fr)

Experience

1999–01 IAESTE Brno, Member of the International Organization for the Exchange of Student for Technical Experiences, Brno, Czech Republic (2 years)

2000 Kumamoto Laboratory (2 months), Modelling, Simulation and Measurement of G6B Electromagnetic Relay, Technical training at OMRON Corporation, Kumamoto Laboratory, 2081-17, Tabaru, Mashiki-machi, Kimi- mashiki-gun Kumamoto, 861-2202 Japan, Supervisor: MSc. Hideyuki Urata (urata@aipc.kml.omron.co.jp)

Skills

- English (very good)
- French (basic)
- Driving Licence
- MATLAB/Simulink, C/C++, PLC programming (B&R company products), Pascal, Assembler, Visual Basic, Dynamic System Modelling
- Advanced PC Skills (web design, LaTeX, etc.)

B Abstract & Acknowledgment

Abstract: Presented thesis is focused on adaptive linear optimal controller which is built and improved with the purpose to use controller in the real process control. The issue of overcoming the quantization effect for decreased sampling period is investigated. Several identification methods working in real-time are compared with identification based on neural networks. Application of neural networks shows the best results on that issue.

Abstrakt: *Disertační práce je zaměřena na návrh adaptivního lineárního optimálního regulátoru za účelem jeho využití pro řízení reálných procesů. V této souvislosti je vyšetřována problematika potlačení kvantizačního efektu při předpokladu krátké periody vzorkování. Použité identifikační metody pracující v reálném čase jsou porovnávány s identifikační metodou založenou na neuronových sítích. V dané problematice vykazuje aplikace neuronových sítí nejlepší výsledky.*

Acknowledgment: I would like to thank professor Pivoňka for his help and useful comments during the whole period of study at Department of Control and Instrumentation in Brno. Next, I would like to thank my parents and Alexandra's parents for their support and encouragement mainly in the last year of study. I have spent four months at the Laboratoire d'Automatique de Grenoble in France in the last year of my study. I would like to thank professor Voda who supervised my work there.

Presented thesis is dedicated to Alexandra.

This thesis has been partially financially supported by the MŠMT research plan CEZ MSM: 260000013 Automation of Technological and Manufacturing Processes and by GAČR grant TALENT 102/03/H116 coordinate education of Ph.D. students in control engineering and robotics.