DIVISION OF AUTOMATIC CONTROL
AND
ENGINEERING INFORMATICS

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY
OF MECHANICAL
ENGINEERING

Modeling of Complex Dynamic Systems by Nonconventional Artificial Neural Architectures and Adaptive Approach to Evaluation of Chaotic Time Series

Ph.D. Thesis

IN THE FIELD OF

CONTROL AND SYSTEMS ENGINEERING

Author: Ing. Ivo Bukovský

Supervisor: Prof. Ing. Jiří Bíla, DrSc.

2007

Acknowledgements

I would like to express my sincere thanks to my supervisor, Prof. Ing. Jiří Bíla, Dr.Sc.[1], for his professional and valuable supervision, open-minded attitude towards my research, and all support resulting in this Ph.D. thesis.

Very special thanks belong to Professor Madan M. Gupta[2] who kindly introduced me to the research of nonconventional neural architectures. I would like to thank him for his excellent supervision during my visit to his research lab in 2003 and for his kind overseas co-supervision since 2003.

My sincere thanks also belong to Professor Zeng-Guang Hou[3] for his valuable suggestions regarding my research, publication activity, and support within scientific community.

The development of higher-order nonlinear neural units (HONNU) and Time-Delay Dynamic Neural Units (TmD-DNU) are parallel branches of the research of nonconvential neural architectures. By courtesy of the NATO Science Fellowships Program and with partial support of an internal grant of the Czech Technical University in 2003, the research of these nonconventional neural units was conducted with the cooperation of the Department of Instrumentation and Control Engineering directed by Prof. Jiří Bíla, with the Intelligent System Research Laboratory in Canada directed by Prof. M. M. Gupta, and also with Prof. Zeng-Guang Hou.

Ivo Bukovský
Prague, March 2007

[1] Professor and director of the Department of Instrumentation and Control Engineering at the Faculty of Mechanical Engineering, Czech Technical University in Prague.

[2] Professor and director of Intelligent System Research Laboratory at the University of Saskatchewan in Canada, (IEEE life fellow, SPIA).

[3] Professor at the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation at the Chinese Academy of Science, Bejing, China.

Contents

Used Symbols and Terms

ANS	…	Autonomous nervous system
BP	…	Backpropagation learning algorithm
CD	…	Correlation dimension.
CNU	…	Cubic neural unit
DNU	…	Dynamic neural unit
ED	…	Embedding Dimension
FFNN	…	Feed-Forward Neural Networks
HONNU	…	Higher-order nonlinear neural unit, (where “order” relates to the order of the polynomial or the level of nonlinearity aggregating function)
DHONNU	…	Dynamic HONNU, i.e., HONNU with time integration of nonlinear aggregation that is also used as the unit’s state feedback
HRV	…	Heart Rate Variability
LLE	…	Largest Lyapunov exponents
MFNN	…	Multi-Layer Feed Forward Neural Networks
MLP	…	Multi-Layer Perceptron (common architecture of NN)
NDE	…	Nonlinear differential equation
Neural Unit	…	Artificial neuron
NN	…	(Artificial) Neural networks
QNU	…	Quadratic neural unit
TmD-DNU	…	Dynamic neural unit with time delays as adaptable neural parameters
TmD-DHONNU	…	Time-delay dynamic higher-order nonlinear neural unit
TmDNN	…	Time-delay dynamic neural networks
TmD-DNN	…	TmDNN
TDNN	…	Tapped-delay neural networks, i.e., conventional NN with delays in interlayer connections

Goals of the Thesis

The research work proposed in this thesis has been pursuing the need for novel evaluation of complex systems that generate complex output signals featuring chaos, multi-attractor behavior, external and internal perturbations,...

The research has shown that the adaptive evaluation of variability of chaotic time series is a very promising direction.

1) As the first objective of this thesis, the theory of the proposed neural units is established. The need for this theory comes from the requirements for the general nonlinear approximation (identification) of complicated nonlinear dynamic systems with minimum number of network parameters, the simplicity of neural network architecture, a practical technique for stable learning algorithm applied to general class of nonlinear aggregating functions, and the capability to implement *apriori* knowledge about an investigated system into a neural unit or a network. The prospects are demonstrated in the application to fast state-feedback neural control of nonlinear plant with variable parameters, in the application to adaptable identification of nonlinear dynamics of parallel tripod manipulator, and in the identification of time-delays within systems or approximation of complex systems by time-delayed systems. The first main objective of this thesis can be more particularly categorized as follows:

    1.1) To develop a tool capable of describing complex systems with focus on the minimum number of neural parameters and with sufficient approximating capability,

    1.2) To benefit from cognitive capabilities of artificial neural network tools, and

    1.3) To maintain the simplicity of mathematical notation of a problem consisting in a low number of parameters and a simple dynamic structure.

2) The second objective of this work is to introduce these new artificial neural architectures as a tool applicable to fast monitoring of changes in levels of variability in signals generated by complex dynamic systems. The advantage over commonly available methods of nonlinear analysis is the adaptive monitoring of actual changes in system dynamics, especially for systems displaying multi-attractor behavior, such as the human cardiovascular system. This second main objective can be categorized as follows:

    2.1) To develop a tool reflecting important characteristics of the dynamics as well as appropriately responding to sudden and continuous changes in dynamics of complex systems in real time, and

    2.2) To establish foundations for a novel method for the variability evaluation of complex signals.

The thesis introduces nonconventional utilization of the gradient-descent backpropagation learning algorithm resulting in:

1. the theoretical development,
2. the systematic approach to and the proposal for standardization, and
3. the practical utilization

of the nonconventional artificial neural architectures for approximation and for further assessment of dynamic systems. The nonconventional neural units are called the higher-order nonlinear neural units (HONNU), time-delay dynamic neural units (TmD-DNU), and time-delay higher order nonlinear neural units (TmD-DHONNU). A special neural unit called HRV-HONNU is proposed for adaptive monitoring of changes in variability of complex time series with a focus on heartbeat tachograms (R-R diagrams).

1 Introduction

Since the beginning of the development of artificial neurons and artificial neural networks (NN), the emphasis has been put on linear synaptic operation (input aggregating function) of a neuron while the neural somatic (output) operation has been considered nonlinear except for the output layer. Nonlinear synaptic operation has not attracted much interest in the literature and the artificial neurons, especially those with a linear synaptic operation, are still poor imitations of their biological counterparts. On the one hand, a conventional structure of an artificial neuron (linear synaptic operation) can provide us with a NN displaying good capability for practical solutions to typical tasks such as pattern classification, system identification, adaptive control, and signal prediction. On the other hand, and from the point of view of exact mathematical solutions (represented by one or more governing implicit dynamic equations), such a conventional NN still represents a black box that does not allow us to obtain information about useful explicit mathematical description of a system and prevents a user from seeking natural and simple solutions.

To assure that conventional neural networks (consisting of neurons with linear synaptic and nonlinear somatic operations) converge toward the sufficient solution of typical tasks mentioned above, the NN shall be composed of a corresponding optimum number of artificial neurons and layers (e.g., Multi-Layer Perceptron NN). The implementation of multiple conventional neural units minimizes the chance of finding appropriate mathematical solutions in the form of corresponding, meaningful, and simple mathematical equations hidden in the structure of trained or adapted NNs.

Therefore, the design of artificial neural architectures that would consist of the minimal number of neural parameters, sustain great computational power to converge to a sufficiently accurate solution, and allow us to retrieve a useful mathematical structure of the system directly from the NN structure has been one of the motivations and guidelines during this research.

The static and dynamic Higher-Order Nonlinear Neural Units (HONNU), whose state of the art and results achieved during the research at the Intelligent System Research Laboratory in summer 2003 follows in the next chapter, represents a movement toward a design of a more natural morphology of an artificial neural unit. This design is facilitated with the natural ability to find existing implicit mathematical solutions in the form of corresponding static or dynamic equation if the appropriate mathematical structure is included as a subset in its nonlinear aggregating operation. In other words, these higher-order nonlinear neural units can be applied for static or dynamic tasks for nonlinear static as well as dynamic systems. HONNU and other proposed neural units can be implemented in a network or can function as stand-alone units undertaking the role of a generally applicable adaptive algorithm for the identification of nonlinear static or dynamic mathematical equations that approximate real unknown systems. The novel neural architectures promise to provide us with further approaches and ideas for investigation of nonlinear systems and development of the theory necessary to understand and solve engineering problems including systems displaying complex behavior.

In summary, two major motivations for development of the nonconventional neural units can be seen in:

1. increasing the computational capability of artificial neural units, and
2. approaching the possible functional and structural resemblance to the real biological neurons with respect to universally applicable achievements of technical sciences (mathematics, control engineering, cognitive sciences, soft computing,…).

It has become apparent that the research of evaluation of complicated nonlinear dynamic systems and the development of HONNU, TmD-DNU, and TmD-DHONNU shall be closely related issues. Naturally, the simplification of an artificial neural network architecture would result in a simpler acquisition of knowledge stored in a minimum number of neural parameters resulting in the appropriate and the simplest mathematical description of a problem. Such a simplification can help solve or simplify many current problems, especially many current engineering challenges regarding, complex dynamic systems, control, monitoring, classification...

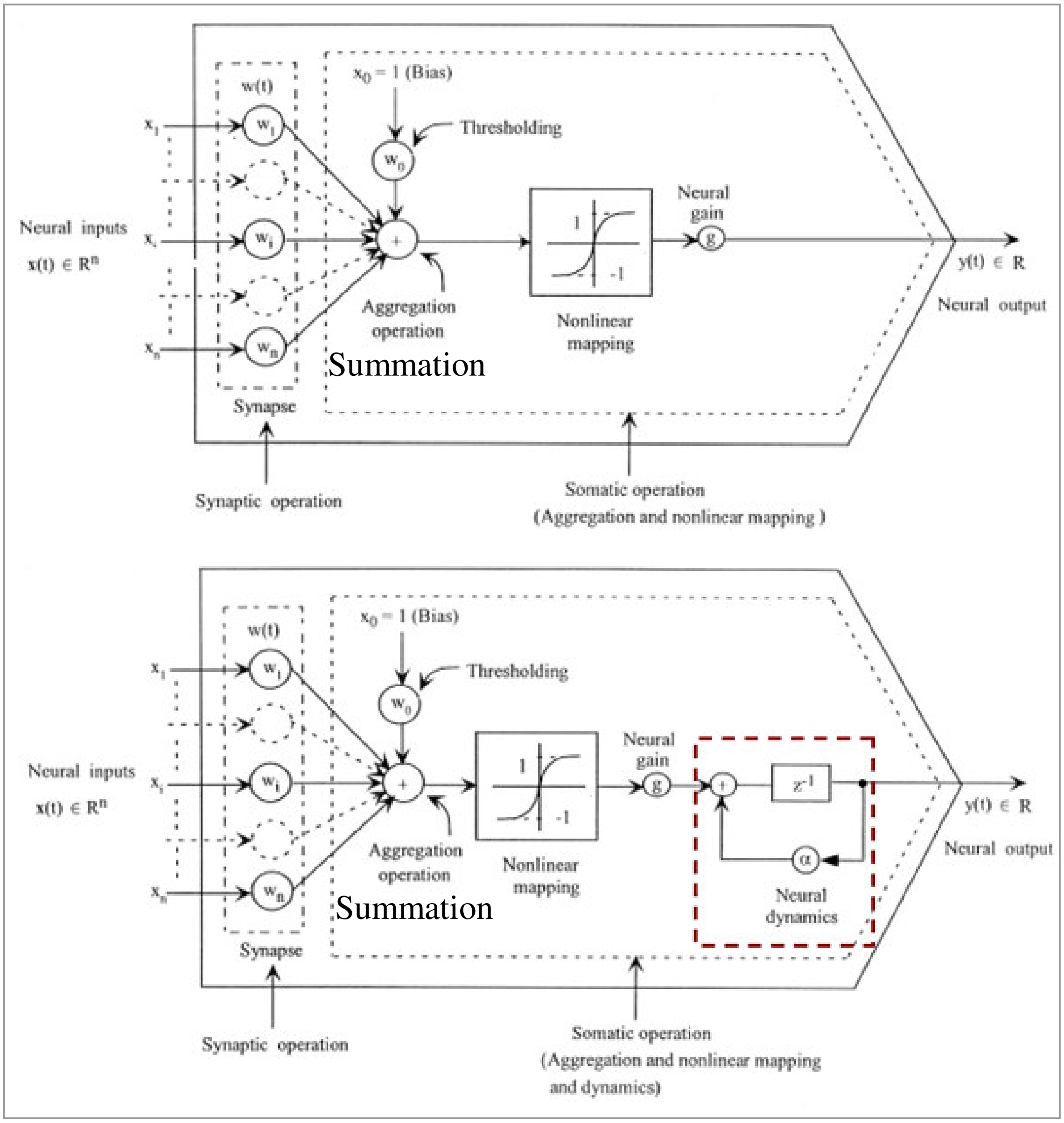


Figure 1: A conventional architecture of an artificial neural unit [adapted from: Madan M. Gupta: *Neural Computing Systems (Introduction to Theory and Applications of Neural and Fuzzy-Neural Systems)* [U. of S., 2000].

2 State of the Current Research

The first section in this chapter highlights the achievements and important aspects of common nonlinear methods for evaluation of chaotic systems with a focus on heart rate variability (HRV). The section also looks at conventional artificial neural networks and addresses the issue of deterministic-chaos component in heart rate variability and the multi-attractor nature of HRV.

The second section (2.2) discusses aspects of the relevant and best known conventional artificial neural architectures as they have been used for handling complex static or dynamic systems. Further, the state of development of nonconventional neural architectures up to the spring of 2003 is sketched [50] to [53].

2.1 IMPORTANT ASPECTS OF HEART RATE VARIABILITY EVALUATION BY NONLINEAR METHODS AND NEURAL NETWORKS

The first subsection in this section introduces the achievements of nonlinear methods in evaluating chaotic systems and HRV. The second subsection discusses the utilization of neural networks for evaluation of chaotic systems and heart rate variability and thus introduces the challenge of the novel evaluation of chaotic systems and multi-attractor dynamics of HRV that is answered by the proposed methodology in this thesis.

2.1.1 Evaluation of HRV and Challenges Resulting from Nonlinear Methods

Characteristic invariants of nonlinear systems have attracted great attention in literature since the 1980's, as in [3] to [5], and have been investigated widely and applied to the evaluation of HRV with a focus on medical diagnosis [6] to [14]. Today's commonly known nonlinear methods (correlation dimension (CD), largest Lypaunov exponents (LLE), and others) have been summarized especially in [26], which focuses on their applicability to the evaluation of heart rate variability. The clinical advantages of the application of nonlinear methods to HRV, such as the increased accuracy of classification of pathological signals, have been summarized in [14] and are also cited and further elaborated in [26]. On the contrary, the common nonlinear methods suffer from seemingly quite complicated dynamic nature of the data, e.g., R-R diagrams (the interbeat time series). The general properties expected from signals allowing the correlation dimension (CD) to be reliably evaluated [3] include:

1. the signal is not too complicated (not too high of an embedding dimension),
2. the signal is sufficiently self-returning (cycle-like behavior),
3. the signal is long enough, and
4. the signal has appropriately low noise to signal ratio.

Even though the above mentioned conditions are originally expressed for CD, other common nonlinear methods, such as LLE, also suffer when the above conditions are exceeded.

Considering the fact that physiological signals (including signals generated by the human cardiovascular system of healthy subjects and sometimes even the strictly deterministic ones [23]) often break especially the top three aforementioned conditions to some considerable extend. The results on heart rate variability evaluation by nonlinear methods suffer from a considerable degree of uncertainty and may lead to non-uniform results and conclusions ([6] to [14]), even though their advantages over the time-based and frequency-based methods are evident. The above mentioned conditions should be viewed as related merely to a particular evaluated signal generated by the system (rather than to the system as a whole) which possibly alters between chaotic modes [14] [26] as opposed to always running in chaotic mode (section 3.2, [19] to [23]).

The possible existence of multiple strange attractors in HRV has already been indicated in [14]. The idea of the multi-attractor behavior of HRV was supported by research and simulation experiments in [26]. This research ([14] [26]) points out the ineffectiveness of common nonlinear methods for evaluation of HRV because they have to be applied to sufficiently long time-series of R-R interbeat recordings of relatively simple dynamics, but the dynamics (attractor) may change significantly within the evaluated signal. Moreover, the evaluations of correlation dimension have failed even for apparently single-attractor highly chaotic signals generated by models [18] to [20] and by those in section 3.2, where parameters of the model were kept constant during simulation.

Further, modeling of the behavior of the cardiovascular system that becomes complex due to fast beat-by-beat controlling influences of the autonomous nervous system (ANS) has been investigated within the frame of cooperation of U12110.3 CVUT FS and 1st Faculty of Medicine of Charles University in Prague [18] [20]. Established by work [18] to [23], the ability of ANS to develop complicated (chaotic) heartbeat dynamics by settings a very few physiological parameters in the "control loop of ANS" has been revealed. It was found throughout physiological analysis and simulation experiments that deterministic chaos in HRV can develop due to the time-delayed fast beat-by-beat control influences of ANS and can be significant even if these delays are kept constant [18] to [23]. The development of deterministic chaos in time-delay systems conforms to common observations made in technical systems or in differential equations that are much simpler than the human cardiovascular system (e.g., Mackey-Glass delay differential equation). Moreover, the origins of chaos in HRV due to fast ANS control correspond to the low HRV observed with patients after heart transplants where the neural lines of ANS have been cut [8] [9] .

Interesting results were obtained by the recurrence-plot method originally introduced in [15] and further developed and applied to the evaluation of HRV (prediction of ventricular tachyarrhythmia) in [16]. The method visualizes n-dimensional-system behavior into 2-D plot and reveals hidden periodicity (recurrences) of the evaluated signal. It displays a two-dimensional visualization of approaching orbits of a solution in reconstructed state space with embedding dimension (theoretically ED $\geq 2n+1$ Takens [4]). According to [16], this method can capture attractor transitions of a system (also corresponding to 'laminar states' in [16]) where common nonlinear methods may not provide reliable results as mentioned above.

Common nonlinear methods (CD, LLE) may become ambiguous for systems with varying dynamics due either to varying system parameters or to deterministic transients on multiple chaotic attractors. Conversely, the recurrence plot handles multi-attractor behavior to some extent, but the method has to work in a 'sliding-window' regime and always has to wait for relatively considerable amount of new samples that will make significant changes in recurrence plot in order to indicate change in the dynamics (change in the attractor) of a system. The, recurrence plot cannot always display the changes in dynamics clearly. An example of the sudden change of bifurcation diagram of the well-known logistic equation is shown below. The left-hand side recurrence plot displays the change in dynamics when the bifurcation parameter *a* changes from 3.8 to 4, while the right hand side recurrence plot does not clearly indicate the change of *a* from 3.95 to 3.96.

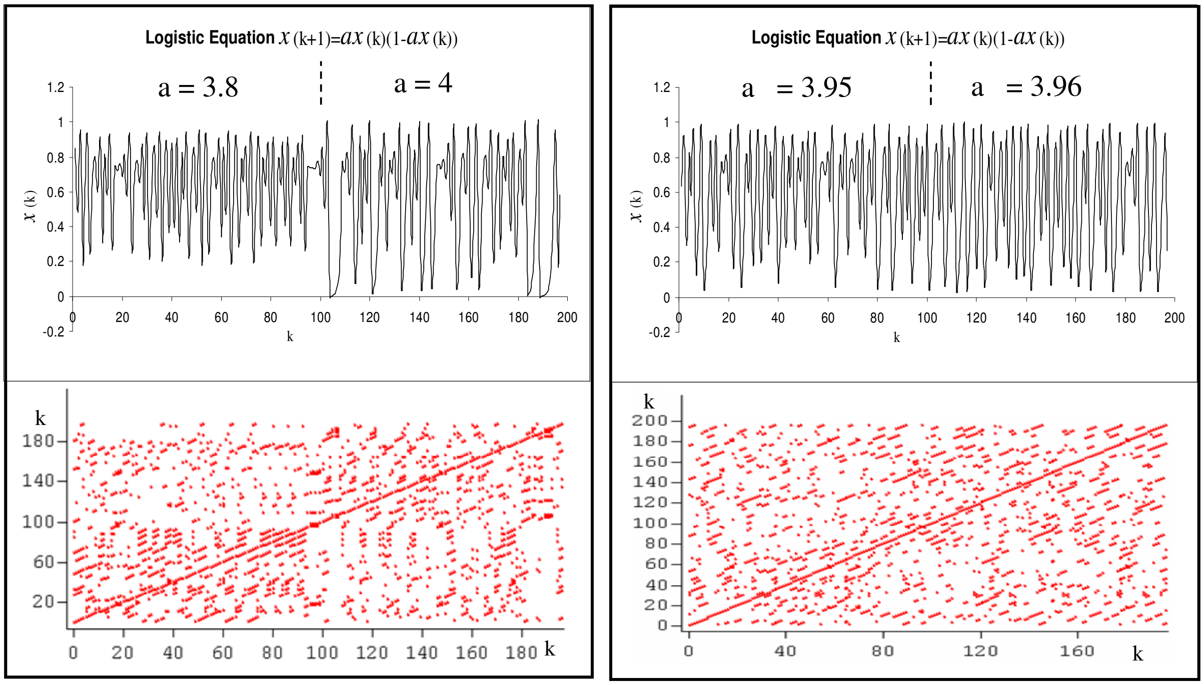


Figure 2: The recurrence plot of time series generated by logistic equation; the recurrence plots do not clearly indicate the change of the bifurcation parameter *a* from 3.95 to 3.96 (see Figure 69, p.98).

Both of the above aftermaths (of common nonlinear methods and recurrence plot) establish a challenge that is answered in this thesis by both the proposed nonconventional neural units (chapter 4) and by the proposed methodology using these neural units (chapters 5 and 6) for monitoring the actual changes in heart rate variability.

2.1.2 HRV and Artificial Neural Networks

As previously mentioned, the utilization of NN for prediction of chaotic signals with a focus on the analysis of cardio-signals has been elaborated on in [27] (Mankova 1997) and further elaborated in [26] (Vitkaj 2001). The analytical and practical observations resulting from the development of feed-forward neural-network (FFNN) models, and their application to chaotic signals and R-R diagrams prediction (interbeat tachograms) in [26], introduce important facts that conform to original ideas leading to this proposed thesis. *Loosely translated* and with comments, a summary of relevant conclusions from [26] follows:

1. *"Small" FFNN (~4/6/1) seemed to extract the characteristic orbit of the system or the characteristic transition between orbits* ([26], p.59). The FFNN with a smaller number of

neurons can learn the chaotic behavior starting from particular initial conditions and can capture the transition to another (coexisting) attractor.

2. *"Large" FFNN (16/24/1) generated signals with a frequency spectrum more similar to that of the original signal.* A FFNN with a higher number of neurons tends to learn the dynamics of the system as a whole.
3. *Low-dimensional chaotic systems with appropriately "returnable orbits" can be very accurately predicted by simple neural models* ([26], p.62). A simple FFNN can very accurately predict systems behaving apparently on a single attractor.
4. *A predicting NN model does not have to characterize the modeled process as a whole; the extracted characteristic can be used for the modeling and classification of chaotic systems* ([26], p.71). Even though the trained NN describes only the actual dynamics of a system for the orbit (attractor) on which the system at that time behaves, it can be used for modeling and classification of chaotic signals.
5. *NN can extract an attractor's geometrical characteristics from noisy data and from a low number of input data* ([26], p.79); this NN property may improve in the case of incremental adaptation of NN if artefacts or noise were present in the minority of data samples. For example, if 10% of samples were due to unwanted artefacts or were affected by noise, then 90% of weight increments accurately approache the system dynamics while only 10% diverges neural weights away. Given 1. and 3. above, it is a technical task to suppress the weight-convergence-distorting data sample (presumably an artefact or a noisy sample) by a simple modification of the general adaptation learning rule; that is, the distorted samples can be detected in a signal in this way. Extracting an attractor's geometrical characteristics from noisy data may become more problematic in case of common batch training techniques used in [26] (e.g., Widrow-Hoff or Levenberg-Marquardt [29]), where all data samples (including the artefacts and noise-distorted samples) may be considered equally significant to the system dynamics.
6. *NN can be applied successfully in cases where ECG recordings are not long enough for common nonlinear methods due to their insufficient convergence or instability* ([26], p.104).
7. *Pathological changes within a critical group of patients can be detected by generating the residua resulting from the comparison of a NN model of the patient and the patients actual physiological recordings.* ([26], p.104).
8. *The design of a NN model of a patient will not be easy because of the complexity and multi-attractor behavior.* ([26], p.104).
9. *NN models should be utilized also for decomposition of multi-attractor dynamics. Due to multi-attractor dynamics, the common nonlinear methods are incorrect over the whole length of the recorded signal, they do not converge, or they result in a significant variance of results, preventing precise and detailed medical diagnoses.* ([26], p.104).

2.2 ARTIFICIAL NEURAL ARCHITECTURES AND COMPLEX SYSTEMS

2.2.1 Conventional Artificial Neurons with Linear Synaptic Operation

Common architectures of Neural NN (MLP, RBF, Hopfield networks) consisting of conventional artificial neurons (with linear synaptic operation - or the aggregating function) (Figure 1) have provided researchers with NNs displaying the ability to perform typical tasks such as pattern classification, system identification, adaptive control, and signal prediction.

Unfortunately, from the point of view of exact mathematical solutions (represented by one or more governing implicit static or dynamic equations), conventional NN still represent a black or gray box that does not allow a user to obtain information about useful explicit mathematical description of the problem. The black box effect prevents researchers from revealing natural and simple solutions derived from further mathematical analysis of simple equations describing a complex problem.

When complex systems, such us nonlinear ones, are to be identified (or rather approximated), the implementation of multiple conventional artificial neurons with nonlinear output function minimizes the chance of finding appropriate mathematical solutions in the form of corresponding, meaningful and simple governing equations hidden in the structure of trained or adapted NNs (Figure 3). To retrieve a useful governing equations of a system that are hidden in a trained conventional neural network is usually difficult when the equations are required for further use, such as for further mathematical analysis of a simple nonlinear differential equation that would still accurately describe a system.

For example, in a typical case of 3-layer single-output FFNN architecture as in Figure 3, the internal mapping function that would be the basis for a governing equation of an approximated system has the structure shown in Eq.(2-1):

$$y_{out} = f(\mathbf{w}_1, \mathbf{W_2}, \mathbf{w}_3, \mathbf{u}) = \phi_3\left(\sum_{i=1}^{m_2} w_{3_i} u_{2_i}\right) = \phi_3\left(\sum_{i=1}^{m_2} w_{3_i}\, \phi_{2_i}\left(\sum_{j=1}^{m_1} w_{2_{ij}}\, \phi_{1_j}(w_{1_j}\, u_j)\right)\right), \qquad (2\text{-}1)$$

where u_j is j^{th} neural input to a network, ϕ_{k_i} is nonlinear output function (somatic operation) of the i^{th} neuron in k^{th} layer, w_{kij} is neural synaptic weight of *ith* neuron in k^{th} layer from j^{th} neuron from previous $k\text{-}1^{th}$ layer, x_{k_i} is output from i^{th} neuron in k^{th} layer, m_k is the number of neurons in k^{th} network layer.

Naturally, the number of neurons and hidden layers correlates to the complexity of observed behavior of an approximated system. The more complicated a system behavior, the more it is to be investigated (identified, controlled, classified…) to a certain degree of considered autonomy, the higher level of nonlinearity may be expected in the dynamics of the system. Therefore, the applied conventional FFNN should be equipped with appropriate nonlinear capabilities in its input-output mapping function (the more robust the NN, the more "degrees of freedom" of the NN in the state space). This is often achieved only by adding neurons or even layers into a network in the case of conventional NN architectures (i.e. with neurons with linear synaptic [input] and nonlinear somatic [output] operations) (Figure 1).

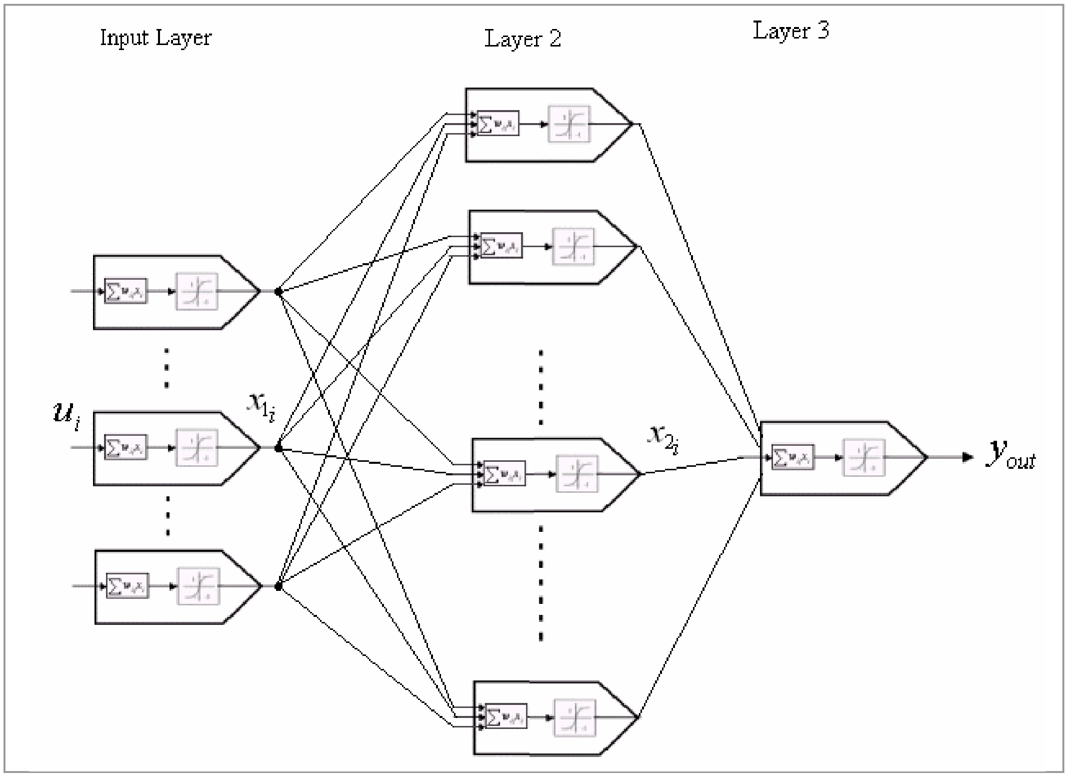


Figure 3: Example of a conventional feed-forward three-layer multiple-input and single-output neural network.

A generalized form of input-output mapping function of such FFNN is then shown in Eq.(2-2)

$$\begin{aligned} y_{out} &= f(\mathbf{w_1}, \mathbf{W_2}, ..., \mathbf{W_{n\text{-}1}}, \mathbf{w_n}, \mathbf{u}) = \phi_n(\sum_j^{m_{n-1}} w_{n_j} u_{n_j}) = \\ &= \phi_n(\sum_j^{m_{n-1}} w_{n_j}\, \phi_{n-1_j}(\sum_i^{m_{n-2}} w_{n-1_{ji}} \phi_{n-2_j} \ldots \phi_{2_j}(\sum_j^{m_2} w_{2_{ij}} \phi_{1_j}(\sum_i^{m_1} w_{1_i} u_{1_i}))\ldots)), \end{aligned} \tag{2-2}$$

where n is the number of network layers including the input layer, $\mathbf{w_i}$ or $\mathbf{W_i}$ are weight vectors or weight matrices representing synaptic connections into neurons of i^{th} layer, and other symbols are similar to Eq.(2-1).

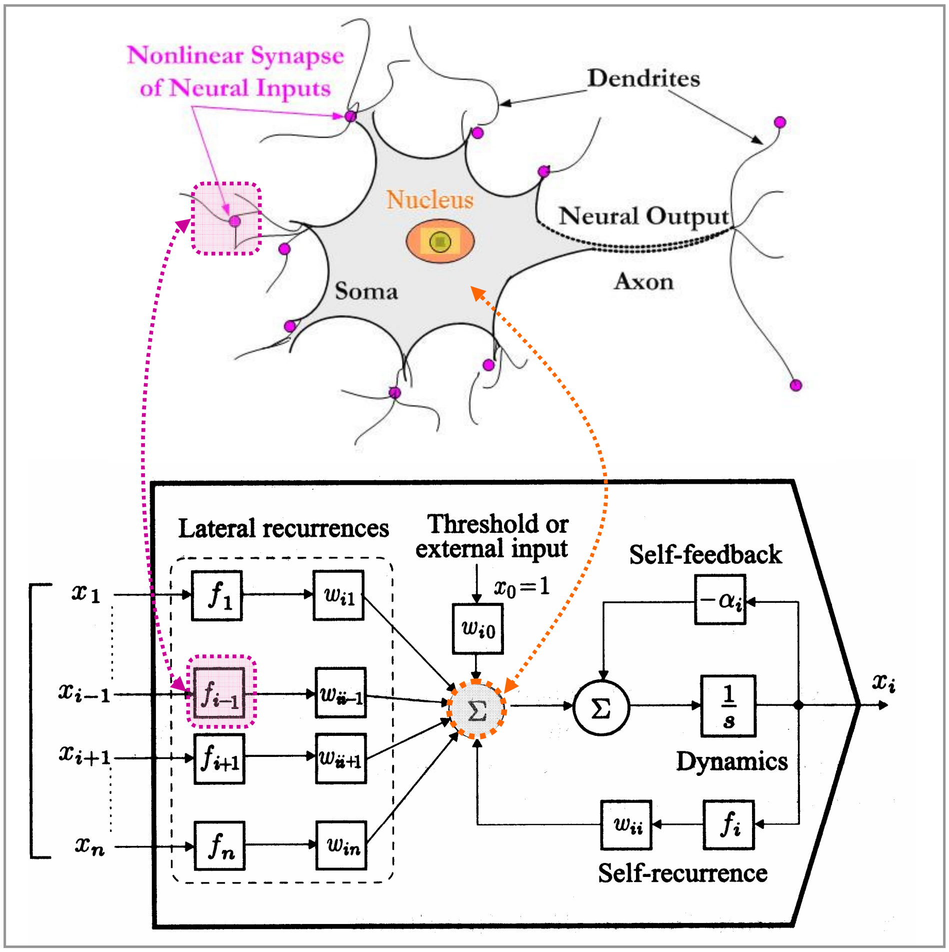


Figure 4: A simplified sketch of a biological neuron assuming nonlinearity in neural synapses (from Gupta, 2003) compared to a conventional artificial dynamic neural unit with linear aggregation (summation) of neural inputs and neural state feedback (Hopfield, Pineda 1980).

2.2.2 The Onsets of Nonconventional Artificial Neural Units

The need for an increase in the computational power of individual artificial neurons was first introduced by A.G. Ivakhnenko (in the late 1960s) in the polynomial neural networks (PNN), where Kolmogorov-Gabor polynomials were utilized in the aggregating function of neurons [30]. Since then, the PNNs have been further developed and have become a branch of neural network research. [33].

In this thesis, the research concept of artificial neural networks (ANN) outlined by Hopfield and later by Pineda [30] [31] is the primary focus. It has been further pursued and developed by the research group of Prof. M.M. Gupta (ISRL, U. of S., Canada) [49] which is linked to research group of Prof. J. Bila (FS CVUT, CR), e.g. [24] to [26] and [28]. Some readers, however, can consider this proposed work to be inspired by the both of the above introduced approaches to research of ANN.

There is a very important point that should be mentioned and that is considered to be the natural synthesis of the two (as well as others) approaches to the artificial neural networks mentioned above. The Wikipedia web page on a biological neuron, in a section titled "Challenges to the neuron doctrine", states:

> „The neuron doctrine is a central tenet of modern neuroscience, but recent studies suggest that this doctrine needs to be revised...
>
> ....., dendrites, like axons, also have voltage-gated ion channels and can generate electrical potentials that carry information to and from the soma. This challenges the view that dendrites are simply passive recipients of information and axons the sole transmitters. It also suggests that the neuron is not simply active as a single element, but that complex computations can occur within a single neuron..."
>
> (*http://en.wikipedia.org/wiki/Neuron#Challenges_to_the_neuron_doctrine* , 22/11/2006)

As mentioned above, another signs of equipping artificial neural units with greater computational power and with the utilization of backpropagation gradient learning algorithm ([29][35][36][48]) was recently published in literature (Wiley & Sons) by Gupta, Liang, Homma in 2003 [49], where one can find the first signs and rudimentary concepts of Higher Order Neural Units. This work, we will be based on the terminology used by M.M. Gupta in his book on neural networks [49]. In the spring of 2003 (when the author joined the research group of Prof. Gupta), the concept of the nonconventional artificial architectures was briefly introduced in its basic principle, and static (Figure 5) and simple dynamic modifications were being developed (Gupta, Song, Redlapalli, Bukovsky, [50] to [52]).

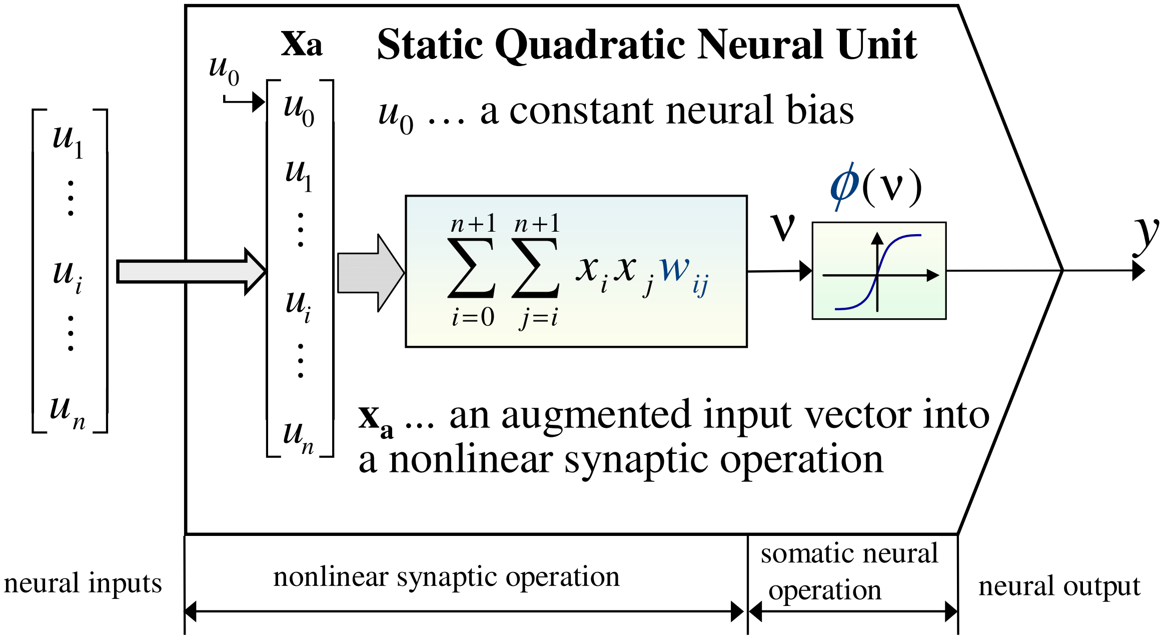

Static Quadratic Neural Unit
Xa
u_0
u_0 … a constant neural bias
$\begin{bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{bmatrix}$
$\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{bmatrix}$
$\sum_{i=0}^{n+1} \sum_{j=i}^{n+1} x_i x_j w_{ij}$
ν
$\phi(\nu)$
y
$\mathbf{x_a}$ ... an augmented input vector into a nonlinear synaptic operation
neural inputs
nonlinear synaptic operation
somatic neural operation
neural output

Figure 5 : Static quadratic neural unit ($\mathbf{x_a}=[x_0,x_1,\dots,x_n]^T=[u_0,u_1,\dots,u_n]^T$)

3 Approaches and Applied Methods

The research and methodologies presented in this work will be sketched in sections 3.1, 3.2, and 3.3. The notion of deterministic chaos in dynamic systems is introduced in section 3.1. In section 3.2, some of our achievements (up to 2003) in research of heart rate variability in the human cardiovascular system will be introduced and synthesized with the basic theory of deterministic chaos in dynamic systems. In section 3.3, the widely applicable mathematical concepts that are successful in technological applications (i.e. system identification and control) are shown to be the inspiration for a new understanding of biological neurons throughout mathematical terms. This merging of mathematics and biology results in the development of proposed higher-order nonlinear neural units, of time-delay dynamic neural units, and of time-delay dynamic higher-order nonlinear neural units.

The basis of the new method of evaluation of complex systems applicable to cardiovascular dynamics will be introduced in sections 3.1, 3.2 and 3.3 in this chapter. Section 3.2 is significantly based on the author's former research work ([19] to [23]) that precedes the development of the new artificial neural architectures proposed in this work.

3.1 INTRODUCTION TO THE APPROACH TO NONLINEAR DYNAMIC SYSTEMS

This section introduces nonlinear dynamic systems which, by their structure, correspond to the design of the proposed neural architectures, HONNU. The principles of developing chaotic behavior in these systems are discussed in subsection 3.1.2 as HONNU display the ability to approximate the dynamics of complex systems. The approaches to evaluating the stability of these systems are introduced in subsection 3.1.3; however, the need for analytical evaluation of the proposed dynamic HONNU has not become the matter of interest because the stability problems during adaptation of dynamic HONNU are avoided by combining static and dynamic implementations of the proposed neural units (section 5).

3.1.1 Nonlinear Dynamic Systems

To introduce the concept of deterministic chaos, the notion of determinism of a system should be approached first. As an aside, the notion of determinism also implies no random inputs, no random variables, and no random external or internal perturbations into a system. We can begin by recalling the uniqueness of solution of set of differential equations (Eq.(3-1)) in region $\mathbf{U} \in \mathbb{R}^n$ in the state space which is assured by its differentiability or, at least, by the Lipschitz property in $\mathbf{U}$ (see the Lipschitz condition below).

$$\begin{aligned}\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{W_f}), \quad &where \;\; \mathbf{x} = \begin{bmatrix} x_1(t) & x_2(t) & \dots & x_n(t) \end{bmatrix}^T, \\ &and \;\; \mathbf{W_f} = \begin{bmatrix} \mathbf{w_{f_1}} & \mathbf{w_{f_2}} & \dots & \mathbf{w_{f_n}} \end{bmatrix}^T,\end{aligned} \qquad (3\text{-}1)$$

The $\mathbf{x}$ is the state vector, **f** is the set of nonlinear functions, $\mathbf{W_f}$ can be a multidimensional matrix or a vector of system parameters (see subsection 4.1.2 for details). The Lipschitz property (the smoothnes) of function **f** also represents the condition for the solution uniqueness of system Eq.(3-1). By definition, the set of functions **f** is Lipschitz on an open set **U** in $\mathbb{R}^n$ if there exists a Lipschitz constant L such that

$$\left|\mathbf{f}(\mathbf{x_1})-\mathbf{f}(\mathbf{x_2})\right| \leq L\left|\mathbf{x_1}-\mathbf{x_2}\right|,\ \forall\ \mathbf{x_1},\mathbf{x_2} \in \mathbf{U} \in R^{\mathrm{n}}, \tag{3-2}$$

which is in fact the condition of all first partial derivatives of **f** being bounded on **U**

$$\left|\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_i}\right|_{\forall \mathbf{x} \in \mathbf{U}} \leq const.,\ \ i = 1,\, 2,\, \ldots\, ,\, n \tag{3-3}$$

The uniqueness of a solution of system Eq.(3-1) on $\mathbf{U} \in R^n$ determines the flow **F** of this system in **U** for all initial conditions in **U**. In brief, the flow **F** of the system can be understood simply as a system solution $\mathbf{F} = \mathbf{F}(t, \mathbf{x}_0)$ or $\mathbf{x} = \mathbf{x}(t, \mathbf{x}_0)$, which is also called an orbit or trajectory of a solution that is a function of time t starting from initial conditions $\mathbf{x}_0$ in a state space.

So far, the notion of determinism of a dynamic system given in Eq.(3-1) has been simply introduced by the uniqueness of solution which corresponds to a single orbit of the system (solution $\mathbf{x}(t, \mathbf{x_0})$) starting from initial conditions $\mathbf{x}_0 \in \mathbf{U}$ located on that orbit.

Once the determinism has been introduced, the features and the origins of deterministic chaos can be introduced by defining a few more common terms used to describe the behavior of nonlinear dynamic systems.

The eigenvalues of a linear system determine the type of stability as one of the following:

1. stable, meaning that the solution (orbit) $\mathbf{x}(t, \mathbf{x}_0)$ finally arrives to an equilibrium point,
2. unstable, meaning that the solution (orbit) $\mathbf{x}(t, \mathbf{x}_0)$ departs, optionally with oscillations, to ideally infinite distance from the equilibrium point,
3. at the edge of stability, meaning that the nonequilibrium solutions (orbit) $\mathbf{x}(t, \mathbf{x}_0)$, where $\mathbf{x}_0$ is not an equilibrium point, is periodically oscillating when at least one eigenvalue of a linear system is complex conjugate number with its real part equal to zero.

System behavior becomes different and complicated with nonlinear systems where an orbit can be stable in the sense that it does not leave some region in state space, but it is not periodically oscillating. Systems that behave this way may be referred to simultaneously as Lyapunov stable (for a particular region of initial conditions) and chaotic (unless they are quasiperiodic).

In order to describe and explain features of this phenomenon referred to as deterministic chaos, more terms such as limit cycle, limit set, chaotic orbit, chaotic set, chaotic attractor, transient chaos, basic types of bifurcations, cascades, manifolds, crisis, and coexisting attractors (multi-attractor behavior) will be briefly introduced below.

There are two kinds of limit sets to be defined:

1. ω-limit sets, further denoted as $\omega(\ )$, which are forward limits sets of an orbit of a solution $\mathbf{x}(t, \mathbf{x}_0)$. A point z is in ω-limit set of system in Eq.(3-1) if the following limit equation holds:

$$\forall z \in \omega(\mathbf{F}(\mathbf{x})) = \omega(\mathbf{x}(t, \mathbf{x}_0)),\ \lim_{t \to \infty} \mathbf{x}(t, \mathbf{x}_0) = z\ , \tag{3-4}$$

2. α-limit sets, further denoted as $\alpha(\ )$, which are backward limit sets of an orbit of a solution $\mathbf{x}(t, \mathbf{x}_0)$. A point *a* is in α-limit set if following limit equation holds:

$$\forall a \in \alpha(\mathrm{F}(\mathbf{x})) = \alpha(\mathbf{x}(t, \mathbf{x}_0)),\ \lim_{t \to -\infty} \mathbf{x}(t, \mathbf{x}_0) = a\ . \tag{3-5}$$

A limit set, whose definitions are meaningful in association with discrete time domains (systems described by time-t maps Eq.(3-10), p.22) is sometimes also referred to as a “limit cycle”, that better fit the definition of a continuous time domain, especially one with linear systems. However, these two terms are not equal in meaning because chaos has a chaotic limit set that is not a cycle. An orbit has been briefly defined as a forward (as well as backward) trajectory of a solution $\mathbf{x}(t, \mathbf{x}_0)$ of an autonomous system Eq.(3-1) in state space, so the chaotic orbit of a flow can be briefly defined as an orbit $\mathbf{x}(t, \mathbf{x}_0)$ that satisfies following conditions:

1. $\mathbf{x}(t, \mathbf{x}_0)$ is bounded,
2. At least one of the Largest Lyapunov Exponents of $\mathbf{x}(t, \mathbf{x}_0)$ is positive (for LLE see chapter on evaluation of chaotic signals),
3. Forward orbit $\mathbf{x}(t, \mathbf{x}_0)$ is not periodic and does not consist only of equilibria or the equilibria with connection arcs (the ω -limit sets does not return to itself), and
4. A point $\mathbf{x}_{ca}$ (*ca* … for connecting arcs) of an orbit $\mathbf{x}(t, \mathbf{x}_0 = \mathbf{x}_{ca})$ either is on a connecting arc or $\mathbf{x}(t, \mathbf{x}_0 = \mathbf{x}_{ca})$ is a connecting arc if both forward limit set $\omega(\mathbf{x}_{ca})$ and backward limit set $\alpha(\mathbf{x}_{ca})$ of this orbit consist only solely of equilibria, or $\mathbf{x}(t, \mathbf{x}_0 = \mathbf{x}_{ca})$ is a connecting arc if it starts and ends in equilibria.

It is important to distinguish between chaotic and quasi-periodic orbits of a solution. While a chaotic orbit was defined a few lines ago and excludes the periodicity of a system in its n-dimensional state space, some particular systems in $\mathbb{R}^n$ for $n \geq 3$ may produce an orbit whose frequency spectrum consists of incommensurate natural frequencies of a system, and whose orbit never returns to itself (including some nonautonomous systems in $\mathbb{R}^2$ that can be transformed into $\mathbb{R}^3$, allowing a solution to wind around an irrational 3-D torus). If such a solution winds indefinitely on an irrational torus never returning to itself (the orbit is dense on such a torus) and if the orbit has positive LLE, this orbit is called quasiperiodic, not chaotic. Especially for systems given by two NDEs, i.e. a continuous dynamic system in $\mathbb{R}^2$, chaotic behavior of a system cannot be achieved. Only quasi-periodical behavior may be achieved under special conditions in accordance with the Poincaré-Bendixon theorem ([1], p.337). In other words, a solution of a system in $\mathbb{R}^2$ (given by two NDEs as in Eq.(3-1)) travels on a curve in the plane. As the Jordan curve theorem states, every closed curve in $\mathbb{R}^2$ creates two separate regions and such a curve must be crossed in order to move between those two regions. Then, the uniqueness of the solution of a given system ensures that the solution in this case may be either diverging (unstable), converging to an equilibrium point (stable) or the limit cycle, but not chaotic or quasiperiodic in that sense. However, for previously mentioned nonautonomous but planar quasiperiodic systems that can draw an irrational torus with dense orbit in $\mathbb{R}^3$ due to existing incommensurate natural frequencies of a system, there is no analogy of the Poincaré-Bendixon theorem and a torus for no closed curve can be divided into two parts. Therefore, only quasiperiodic behavior can develop in continuous time systems in $\mathbb{R}^2$.

Once a chaotic orbit has been defined, we can advance to a chaotic set and then to a chaotic attractor.

A chaotic limit set is a forward limit set $\varpi(\mathbf{x}_0)$ of a chaotic orbit $\overline{\mathbf{x}}(t,\mathbf{x}_0)$. In principal, the points of a chaotic orbit with chaotic forward limit set may also be included in the chaotic limit set. If a nonlinear dynamic system with a chaotic limit set $\varpi(\mathbf{x}_0)$ is started from various initial conditions considered apart from its chaotic limit set $\mathbf{x}_0 \neq \varpi(\mathbf{x}_0)$, and if the orbits on these initial conditions finally arrive at the $\varpi(\mathbf{x}_0)$, then the chaotic limit set $\varpi(\mathbf{x}_0)$ is the chaotic attractor of the system, and the region of the initial conditions which arrives at $\varpi(\mathbf{x}_0)$ is a basin of attraction of this attractor.

It should be emphasized that a chaotic limit set shall not be called a chaotic limit cycle because a chaotic orbit never returns to itself and the terms cannot be used interchangeably.

Another important type of chaotic behavior in nonlinear dynamics systems is the transient chaos when an orbit of a solution is passing through a chaotic region up to some specific time after which the solutions suddenly quits its chaotic regime for the same parameters and inputs to a system. For example, transient chaos can be found in the Lorenz's equations [1]. Until now, chaotic behavior has been described with a focus on autonomous continuous-time dynamic systems.

However, the second (and very important) mechanism of development of chaotic behavior in nonlinear dynamic systems can be found in nonautonomous systems such as

$$\begin{gathered}\mathbf{x}'(t) = \mathrm{f}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{W_f}) = [f_1(\mathbf{x}(t),\mathbf{w_{f_1}},u_1(t)) \quad f_2(\mathbf{x}(t),\mathbf{w_{f_2}},u_2(t)) \quad \ldots \quad f_n(\mathbf{x}(t),\mathbf{w_{f_n}},u_n(t))]^T,\\ where\ \mathbf{x} = \left[x_1(t) \quad x_2(t) \quad \ldots \quad x_n(t)\right]^T,\\ \mathbf{W_f} = \left[\mathbf{w_{f_1}} \quad \mathbf{w_{f_2}} \quad \cdots \quad \mathbf{w_{f_n}}\right]^T,\\ \mathbf{u} = \left[u_1(t) \quad u_2(t) \quad \ldots \quad u_n(t)\right]^T,\end{gathered} \tag{3-6}$$

or

$$f(x^{(n)},x^{(n-1)},\ldots,x',x,u,\mathbf{w}_f) = 0, \tag{3-7}$$

where $u, u_1, u_2, \ldots, u_n$ are periodic input functions where at least one is periodic while the remaining are zero or constant.

Thus Eq.(3-6) or Eq.(3-7) form a common family of forced oscillators in which quasiperiodic or chaotic behavior may develop if a system is of second or higher order of dynamics, respectively. A well-known example of a forced oscillator with quasiperiodic behavior is the Duffing equation [1]. In short, there are two ways of development of chaos in forced oscillators, both corresponding to the previously mentioned concepts. According to a physical interpretation, a dissipative system loses its energy along its orbit, thus it is attempting to arrive at some of its energy minima. However, due to periodically forced input or inputs into a system, the solution does not settle down to an equilibrium point or limit cycle and may orbit among various equilibria of various types in a state space. According to a particular nonlinearity of a system itself as in Eq.(3-6) or Eq.(3-7), its behavior corresponds to a complicated vector (slope) field (Figure 6-b, Figure 7) in the state space where the solution flows, and complicated and even chaotic behavior of a system may develop.

Another way of looking at nonautonomous systems as in Eq.(3-6) and Eq.(3-7) is as a transformation into a system with constant unit input by declaring time as an additional state variable,

$$t \leftarrow x_{n+1},\ \dot{x}_{n+1} = 1 \Rightarrow \mathbf{u}(t) \leftarrow \mathbf{u}(x_{n+1})\,, \tag{3-8}$$

which is useful mainly for two- dimensional systems in demonstrating principles of quasi-periodic behavior on an irrational torus [1].

Such a nonautonomous system in n-dimensional state space with forcing input given by function vector $\mathbf{u}(t)$ then yields a n+1-dimensional nonautonomous system with a single constant input function $\mathbf{u}(t)=1$, which is

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \\ \dot{x}_{n+1}(t) \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}(t), \mathbf{w}_{\mathbf{f_1}}, u_1(x_{n+1})) \\ f_2(\mathbf{x}(t), \mathbf{w}_{\mathbf{f_2}}, u_2(x_{n+1})) \\ \vdots \\ f_n(\mathbf{x}(t), \mathbf{w}_{\mathbf{f_n}}, u_n(x_{n+1})) \\ u_{n+1} = 1 \end{bmatrix}, \quad \text{or simply } \begin{bmatrix} \mathbf{x}'(t) \\ x_{n+1} \end{bmatrix} = \begin{bmatrix} \mathrm{f}(\mathbf{x}(t), \mathbf{u}(x_{n+1}), \mathbf{W_f}) \\ 1 \end{bmatrix}. \tag{3-9}$$

Up to this point, some of most important terms were defined and various types of behavior of nonlinear (real) dynamic systems have been discussed with a focus on introducing the complicated behavior of real-world systems known as “deterministic chaos”.

3.1.2 Basic Principles of Development of Chaotic Behavior in Nonlinear Dynamic Systems

In this chapter, common mechanisms of development of chaotic behavior in deterministic systems, such as Eq.(3-1) will be discussed. These mechanisms, which are basically bifurcations, bifurcation cascades, and the related crisis, will be discussed as they can occur in a simple or only a one-parameter class of autonomous systems whose parameter varies in real time.

As mentioned previously, the chaotic nature of a deterministic system corresponds to the complex morphology of the vector field (slope field), the dislocation of system equilibria, and the variety of their types in the state space wherein a system performs. All these properties of a system are due to various types of nonlinearities presented in a system[4]. When the behavior of such a system in the state space is ruled by either governing differential equations (continuous flow in the state space) or discrete functions mapping the subsequent states of a system along orbits of solutions (so called time-T maps), the actual configuration of parameters of these equations together with initial conditions of a system and system inputs determine system behavior. Changes in these parameters may trigger chaos. In other words, these changes can result in the sudden appearance or disappearance of a chaotic attractor, can cause hysteresis in transitions between various types of behavior including chaos, and can cause other complex phenomena which cannot appear with linear systems (see footnote 4).

[4] Time-delay systems are not included in this introductory part because the final application to heart rate variability in this thesis has not used adaptable time delays for monitoring the dynamics yet; it is subject of further research.

This work concerns continuous-time and discrete-time neural architectures, and an introduction to basic chaos triggering mechanisms via discrete governing equations follows in this subsection.

The time-T map in Eq.(3-10) may represents a discrete analogy to the system in Eq.(3-1), where the continuous-time variable of time t becomes the fixed time difference $t \leftarrow k\,.\,T$.

$$\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}, \mathbf{W}_\mathbf{g}) \tag{3-10}$$

The family of maps $\mathbf{g}(\;)$ that represent diffeomorphism or at least homomorphism are considered in this subsection to introduce chaos triggering mechanisms. Certainly, these maps do not have to necessarily represent the discrete-time alternative to continuous systems with fixed period *T*. If the period *T* is not to be fixed, then the map $\mathbf{g}(\;)$ can represent the governing equation of system behavior on, e.g., Poincaré section. Another case is the map of discrete time series such as R-R diagrams where data are sampled every R peak occurs in ECG signal (section 5).

As previously noted, mechanisms developing chaotic behavior in dynamic systems can be well approached through discrete time-T map system description Eq.(3-10). The mechanisms, also called “generic bifurcations”, are the period doubling bifurcation and the saddle-node bifurcation, which are the most common chaos triggering principles. Other cases, such as the pitchfork bifurcation, are considered to be less common (Table 1, [1]).

In general, a bifurcation occurs when some particular parameter of a nonlinear dynamic system is varied and appears as a sudden change in qualitative system behavior related to a change in appearance or a shift in the number, type, or in both number *and* type of equilibrium points in a state space.

For example, a saddle-node bifurcation in time-T maps is the appearance of a new pair of fixed points in a region in the state space when a parameter of a system has been varied and no fixed points had existed previously in that location. For time-T maps in the one-dimensional state space, the bifurcation is called the tangent bifurcation. The term “equilibrium point”, here denoted as $\bar{\mathbf{x}}_\mathbf{e}$, is used normally for continuous-time systems in Eq.(3-1) as follows

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{W}_\mathbf{f}) = 0 \Rightarrow \bar{\mathbf{x}} = \bar{\mathbf{x}}_{\mathbf{equillibrium}}\,, \tag{3-11}$$

while the term “fixed point” refers to the equilibrium point of equation Eq.(3-10) that may be the time-T map of the same system as in Eq.(3-1) and may be its equilibrium point unless periodic solution with period *T* occurs.

$$\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}, \mathbf{W}_\mathbf{g}) = \mathbf{x}_{k-1} \Rightarrow \bar{\mathbf{x}} = \bar{\mathbf{x}}_{\mathbf{fixed}}\cdot\,, \tag{3-12}$$

The Eq.(3-1) and Eq.(3-10) may be two different representations describing behavior of the same dynamic system in its state space, where Eq.(3-1) stands for continuous-time representation and Eq.(3-10) “samples” the continuous-solution trajectory in intervals of length *T*. While an equilibrium point in the case of a continuous-time system (Eq.(3-1)) is strictly given and clearly defines the steady state of a system in its state space, a single existing fixed point of a time-T map is to be viewed as a point on possible periodic orbit with period *T*.

$$\mathbf{x}(t, \bar{\mathbf{x}}_\mathrm{f}) = \mathbf{x}(t+T, \bar{\mathbf{x}}_\mathrm{f}) \;\Leftrightarrow\; \mathbf{x}_{k+1} = \mathbf{x}_k = \bar{\mathbf{x}}_f = \mathbf{g}(\mathbf{x}_f) \tag{3-13}$$

Obviously, the map would display multiple points on Poincaré section that would result from intersections between the Poincaré section and the orbit of a solution as the orbit, starting from initial conditions $\mathbf{x_0}$ **,** would travel in a state space toward the attractor, which is a periodic ω-limit set with period T in such case. Later in the thesis, a possible way to find the time-T map that would be the discrete-time alternative to continuous-time dynamic system will be proposed using discrete static as well as dynamic neural architectures (HONNU), which belong among systems as in Eq.(3-10).

$$\begin{array}{l} \mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{W_f}), \;\; \textit{where} \;\; \mathbf{x} = \left[x_1(t) \quad x_2(t) \quad \ldots \quad x_n(t)\right]^T, \\ \qquad\qquad\qquad\qquad \textit{and} \;\; \mathbf{W_f} = \left[\mathbf{w_{f_1}} \quad \mathbf{w_{f_2}} \quad \ldots \quad \mathbf{w_{f_n}}\right]^T, \end{array} \Leftrightarrow \;\; \mathbf{x}_{(t+T)} = \mathbf{g}(\mathbf{x}_{(t)}, \mathbf{W_g}) \qquad (3\text{-}14)$$

The time difference T in the above considered time-T maps is not limited in length as in the case of common discrete approximations of continuous systems nor is the sampling period T necessarily constant (Poincaré sections, R-R diagrams), thus it is distinguished from these common discrete approximations of continuous systems.

Further, the general parallel between a system described by Eq.(3-1) and its discrete-time alternative time-T map described by Eq.(3-10) will be discussed with a focus on the principles of deterministic chaos.

A periodic continuous-time orbit x(t,$\mathbf{x_0}$,$\mathbf{W_f}$) of system Eq.(3-1) with period T or its by-integer multiples $T_p = i.T$, where $i = 1,2,3,\ldots$, has to be viewed as an underlying structure of a fixed point in time-T map representation Eq.(3-1) of the same dynamic system. Thus, the saddle-node bifurcation can be viewed as the appearance of a new pair of fixed points within systems given by time-T maps. In parallel with that, the creation of two fixed points can represent the establishment of two new continuous-time periodic orbits of the system where there had been a single equilibrium (sink) point before. When a new pair of fixed point appears with a saddle-node bifurcation of time-T maps, these two points move on emanating branches as a bifurcation parameter of a system (some particular parameter in matrix $\mathbf{W_g}$, and thus parallelly in $\mathbf{W_f}$) is further varied. In particular, this property of moving fixed points on branches emanating from a former single fixed point is referred to as the concept of fixed point continuability (generally applicable to some neighborhood of a fixed point). It is typical for a saddle-node bifurcation after passing the bifurcation value where the type of one of the emerging fixed points is a saddle (or repeller in one-dimensional systems) and the second one is an attractor.

In the case of period doubling bifurcation, each existing fixed point of a system splits into two more fixed points as the particular one or possibly more parameters of a system (those in $\mathbf{W_g}$ and parallelly in $\mathbf{W_f}$) are varied over next bifurcation value.

Unlike a saddle-node bifurcation, the period doubling bifurcation appears where some fixed point had already existed. In most generic cases, when the existing fixed points bifurcate, they switch their stability, i.e. the attractor sinks become repellers (sources) and the unstable fixed points become attractors. In the bifurcation diagram, two new branches of fixed points emanate from an actual periodic point of period *k,* called period-*k* fixed point, and thus period-*2k* fixed point creates and moves in the diagram as some bifurcation parameter or parameters of a system are varied. Even though the principles describing the two generic bifurcations, i.e., the period doubling and saddle-node bifurcation, can be found in classes of one, two, and possibly more dimensional time-*T* maps, they should still be considered basic examples introducing notions of the development of chaotic behavior in deterministic dynamic systems. Because the stability types of fixed points are simpler in the case of one-dimensional time-*T* maps (either an attractor or source (repeller)), the development of chaotic behavior through bifurcations can be grasped clearly.

The third and quite common case of bifurcation is the pitchfork bifurcation, which occurs in systems under special conditions, for which it is referred to as non generic bifurcation. This bifurcation can be explained again using the concept of one-dimensional maps. A specific condition placed upon a system to undergo a pitchfork bifurcation can be the symmetry of a map Eq.(3-1) describing the system.

$$g(\text{-}\mathbf{x}_k, \mathbf{w_g}) = -g(\mathbf{x}_k, \mathbf{w_g}), \tag{3-15}$$

where w_g is a vector of system parameters.

When a particular parameter of a system is varied over a bifurcation value, thus evoking the pitchfork bifurcation, then an existing branch of a continuous path of an unstable fixed point continues further as a path of stable fixed point, and two new branches emanate from the bifurcation point.

The changes in the fixed point properties, which indicate the generic and pitchfork bifurcations, can be summarized in Table 1, where wgi_{bifur} is a bifurcation parameter at the bifurcation value and x_{fixed} is the fixed point.

type of bifurcation	$\left.\frac{\partial g(x,\mathbf{Wg})}{\partial x}\right\|_{\left[\begin{smallmatrix} x=x_{fixed} \\ w_{gi}=w_{gi_{bifur}} \end{smallmatrix}\right]}$	$\left.\frac{\partial g(x,\mathbf{Wg})}{\partial w_{gi}}\right\|_{\left[\begin{smallmatrix} x=x_{fixed} \\ w_{gi}=w_{gi_{bifur}} \end{smallmatrix}\right]}$	$\left.\frac{\partial g^2(x,\mathbf{Wg})}{\partial x^2}\right\|_{[...]}$
saddle node	1	$\neq 0$	$\neq 0$
pitchfork, transcritical	1	0	$\neq 0$
period doubling	-1	$\neq 0$	$\neq 0$

Table 1: Fixed-point conditions for common bifurcations of 1-D maps, copied and modified from Ref. [1] p. 467

Suppose, for some particular region in a state space in $\mathbb{R}^n$, that a time-T map $\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}, \mathbf{W_g})$ represents a discrete analogy (not an approximation) to a continuous-time description of a system (flow)

$$\mathbf{x}(t) = \mathbf{x}(t, \mathbf{x_0}, \mathbf{W_{fs}}) \Rightarrow \mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}, \mathbf{W_g}), \tag{3-16}$$

where $t \leftarrow T.\,k$ and $\mathbf{W_{fs}}$ consists of parameters of the solution that are presumably functions of continuous-time system parameters $\mathbf{W_f}$ as in Eq.(3-1), that is $\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{W_f})$.

Then, let us simply draw parallels between previously introduced bifurcations in discrete-time-T maps and bifurcations in (continuous-time) NDEs in order to introduce basic chaos triggering mechanisms in NDEs, as in Eq.(3-1).

Suppose that there exists a deterministic nonlinear system capable of displaying distinctly complicated types of behavior, including a chaotic one. Moreover, suppose bifurcations are the chaos triggering mechanism there. Suppose that there exist two equivalent sets of governing equations that accurately approximate the system behavior in a particular region in $\mathbb{R}^n$ for $n \geq 3$.

The first set of governing equation describes a system in discrete-time domain, i.e. time-T maps as in Eq.(3-1)

$$\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}, \mathbf{W}_\mathbf{g}) .$$

The second one describes the system in continuous-time domain, i.e. by NDEs as in Eq.(3-10)

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{W}_\mathbf{f}) .$$

What are the parallels to the corresponding continuous flow $\mathbf{x}(t, \mathbf{x_0})$ of the solution of NDEs' if the previously described bifurcations appear in the time-T map of such a system?

A single fixed point of the time-T map can also be associated with a single periodic orbit of period T of a continuous flow Eq.(3-10). One can simply think of the most basic instance of a saddle-node bifurcation in NDEs as a splitting of the existing equilibrium point and the appearance of a new periodic orbit, of period T. The new orbit further develops in a state space simultaneously with the further variation of a bifurcation parameter (or parameters), while its time period T is sustained (as results from simulated or experimentally-obtained bifurcation diagrams).

As the saddle-node bifurcation parameter is further varied, the new saddle-node periodic orbit appears and then splits out to one stable (attractor) and one unstable (source, repeller) orbit [1]. These two orbits may depart from each other as the parameter is varied (note the distance between branches of the fixed point and the periodic continuous orbits in the bifurcation diagram).

A brief note should be made about an obvious, yet important point. The common paradigms known from the domain of linear systems barely apply to nonlinear dynamic systems. For example, when a parameter of a linear system is varied, it influences the stability and oscillations of the system via the roots of a system. Conversely, if a (bifurcation) parameter is varied in nonlinear dynamic system, the period of the system may remain constant (the same number of branches of fixed points in a bifurcation diagram) until another bifurcation occurs. In other words, the varying bifurcation parameter can modify the location of the created periodic orbits, i.e., the location of the attractors and sources. This dislocation modifies the distance in $\mathbb{R}^n$ that the solution travels on the periodic orbit before returning to itself; possibly it might modify its shape[5], but it does not prolong nor shorten the time that the continuous solution $\mathbf{x}(t, \overline{\mathbf{x}}_f)$ spends on a stable periodic orbit (attractor) before returning to itself. This can also be seen clearly with the period-doubling bifurcation. Given the previous proposition, an association can be made between period doubling bifurcations in time-T maps and the corresponding phenomenon in the same system described alternatively by NDEs. The fixed points continue on their branches in the bifurcation diagram, as the bifurcation parameter is varied; that is, the periodic orbit may vary in its location, length, and possibly shape in $\mathbb{R}^n$. They do not, however, change in the time length of the interval (integer multiples of T) that they spend on the actual attractor until the next

[5] Bifurcation diagrams do not say anything about the length and morphology of the periodic orbits, but they specify the location of fixed points that are single points in the path of continuous periodic orbits with a period of integer multiple of T.

(possibly period-doubling) bifurcation occurs. While the length of the periodic orbit in $\mathbb{R}^n$ state space varies, the relative position and distance of the fixed points in the bifurcation diagram and the period still remains constant (integer multiple of T), because of the continuous path of a fixed point in a bifurcation diagram.

This concept can be explained by the system gaining kinetic energy on its stable orbit while the bifurcation parameter is varied and the attractor is prolonged in $\mathbb{R}^n$ (and possibly modified otherwise), and the time period of the attractor remains unchanged at least according to bifurcation diagrams.

Once the basic types of bifurcations were introduced as possible mechanisms triggering the complicated behavior of nonlinear dynamic systems described in the discrete as well as in the continuous-time domain, the whole process of consequent bifurcations in a system, as one or more bifurcation parameters of a system are varied, is called a cascade. A cascade can result in a mode of system behavior referred to as chaotic once a measureless number of fixed points has been generated through the cascade. A typical case is the well known cascade of period doubling bifurcations in a logistic (population) time-T map that is generally observed with the tent-shaped family of maps [1].

The previously introduced types of cascades, and their bifurcations resulting in chaos, are basic ones and can be found in selected families of maps and in nonlinear differential equations. The nonlinear system behavior at the bifurcation transitions, resulting in a variety of changes in the types of fixed points, smooth continuous and sudden topological changes in limit sets and attractors in a state space may be more complicated for more general nonlinear dynamic systems. Their description is difficult, especially in the case of higher dimensional systems, and is outside the scope of this work. While a basic cascade in a time-T map may be interpreted as a process of consequent bifurcations generating fixed points of various type (sinks, repellers, saddles) and their continuous paths (branches in a bifurcation diagram) interconnected through the bifurcation points, as some system bifurcation parameter is varied, it would generally represent variations in complexity of the slope field of the NDE describing the same system in a state space related to the increase of number, type changes, and locations of single equilibrium points, as well as creating, prolonging, and merging various limit cycles (α-limit sets) and thus prolonging the period of flow of solution as previously interpreted from discrete into continuous time domain for each particular type of bifurcation.

While the concept of cascades can be good for introducing chaos in discrete systems (recurrent time T-maps) and their possibly parallel continuous-time systems (NDEs), the concepts of stable and unstable manifolds and the related existence of homoclinic points and homoclinic trajectory must be recalled, as they represent a significant condition for the development of deterministic chaos in dynamic systems [1] [2]. Thus, a crisis in a chaotic attractor should be mentioned as it represents another important principle resulting in a sudden change of level of chaos in a system behavior.[6] Contrary to previously mentioned chaos triggering mechanisms, the concept of stable and unstable manifolds related to crisis in a chaotic attractor will be recalled as one of the most significant ones relating to chaotic developments in dynamic systems.

[6] Simply, the ”level of chaos” such as measured by LLE or correlation dimension, e.g.

Let us begin with the saddle-node type of equilibria. In brief, a stable manifold of an equilibrium point is the set of all points whose ω-limit set is the equilibrium itself. Similarly, the unstable manifold is the set of points whose α-limit set is the particular equilibrium itself. In linear dynamic systems, the manifolds reduce to straight lines in the directions of the eigenvectors of fixed or equilibrium points (-a), while they may become complicated curves in nonlinear cases (Figure 6-b). In Figure 6-a, the stable and unstable manifolds of a linear system, Eq.(3-17), of a saddle equilibrium point are shown. In Figure 6-b, the manifolds of nonlinear system Eq.(3-18) of one saddle [0,0] and one elliptical [-1 0] equilibrium point are shown. The blue arrows indicate the orientation of slope vectors emanating from points shown either as red spots, denoting positive derivative of Lyapunov function (Eq.(3-23), p.31), green spots, denoting its negative value, and black dots, indicating equilibrium points. The stable manifold between the equilibria is the connecting arc.

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= 2\,x_1(t) + x_2(t) \end{aligned} \tag{3-17}$$

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= x_1(t) + x_1^{\,2}(t) + 0.1\,x_2(t)\,x_1(t) + 0.1\,x_2(t) + x_2^{\,2}(t) \end{aligned} \tag{3-18}$$

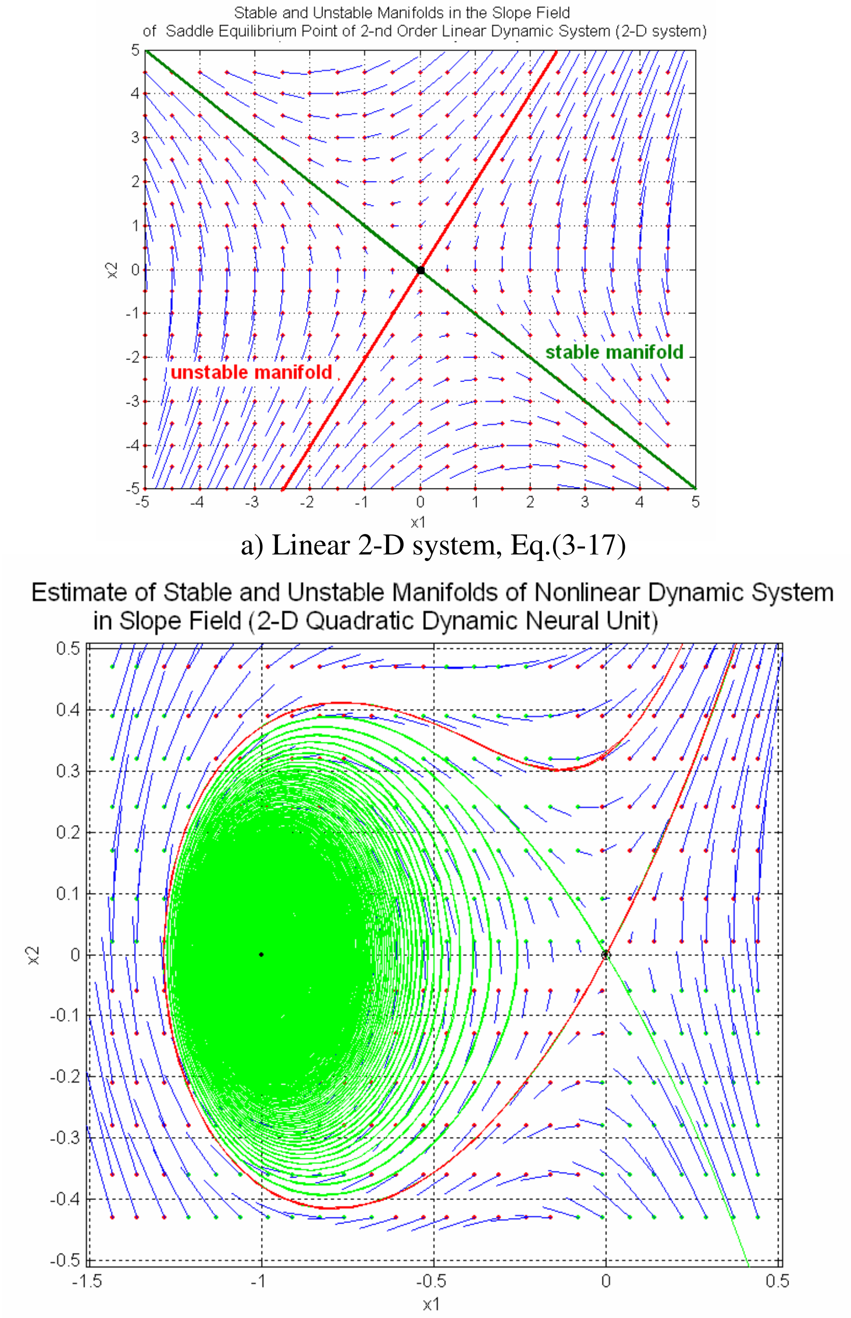
As seen from the 2-D examples in Figure 6 and Figure 7, topological structures of manifolds of nonlinear dynamic systems become complicated, especially once dealt with higher-dimensional systems. Lambda lemma [1] says that if some curve *L* crosses an unstable manifold of a hyperbolic fixed point transversally, then the particular orbits, containing the points of that curve *L,* follow the unstable manifold; thus, such solutions cannot be (locally) stable in this area of the state space.

It is well known, that given the uniqueness of a solution and the Poincaré-Bendixon theorem, any 2-D continuous time deterministic systems may not become chaotic.

As we can see in Figure 6-b as well as in Figure 7, the stable and unstable manifolds of the saddle equilibrium, x_0 = [0,0], tangentially approache each other[7]; however, they do not cross each other in the case of Figure 7.

Consider dynamic systems of dimension $n > 2$ with existing homoclinic points, which are constituted by the crossings of stable and unstable manifolds. Considering the previously introduced inclination (lambda) lemma [1], some concepts of the origins of chaotic dynamics can be then easily grasped. Once there is a homoclinic point created by the transversal crossing of stable and unstable manifolds, indicating that a homoclinic structure may exist (e.g., a homoclinic loop that is a closed loop consisting of stable and unstable manifolds), then a solution that flows on an orbit along the stable manifold will not settle into an equilibrium because it has drifted away along the unstable manifold according to the inclination lemma [1], demonstrated in Figure 7. Such a homoclinic structure that is attracting and is not a limit cycle is called a chaotic attractor (or sometimes a strange attractor [Ruelle and Takens]).

[7] In fact, the system in Figure 6-b and Figure 7 is asymptotically unstable in any neighborhood of the two equilibrium points except in stable manifolds of x0 = [0,0] (green curves) ; the saddle equilibrium x0 = [0,0] affects the oscillating character of the elliptical equilibrium x0 = [-1,0].



a) Linear 2-D system, Eq.(3-17)

b) Nonlinear 2-D dynamic system, Eq.(3-18)

Figure 6: Examples of stable (green curves) and unstable (red curves) manifolds in the slope field of 2-D dynamic systems (linear and nonlinear 2-D dynamic HONNU).

The phenomenon of the abrupt variations of a chaotic attractor as system parameters are varied can be related to the inclination lemma. This occurrence is called a crisis in a chaotic attractor and occurs under special conditions or when a chaotic attractor collides with unstable manifolds (as of a hyperbolic equilibrium point or a limit cycle). Thus, the chaotic attractor can suddenly increase in size when a transversal crossing with unstable manifolds occurs.

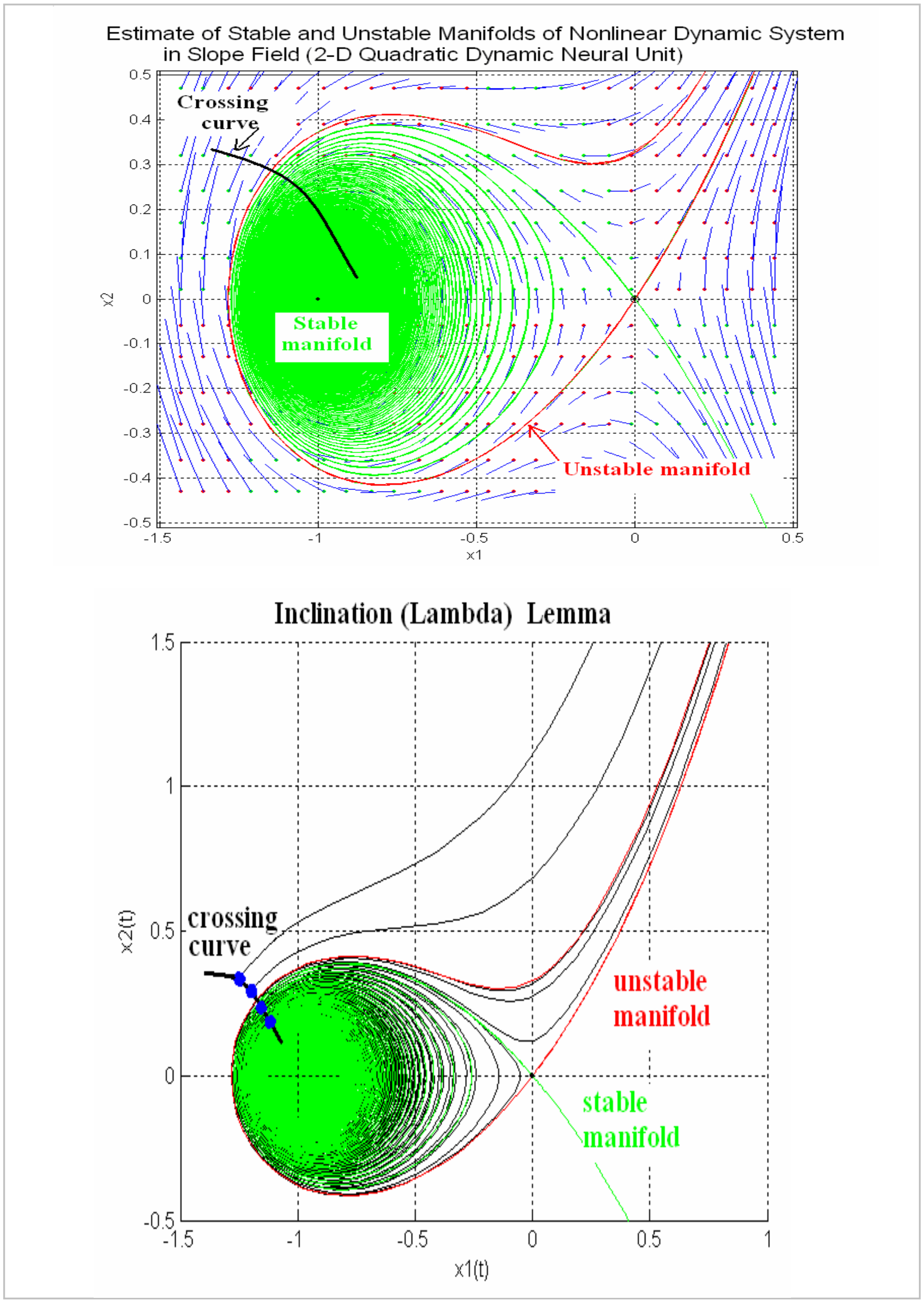


Figure 7: Inclination (Lambda] Lemma - A curve crossing the unstable manifold of a saddle equilibrium point for system in Eq.(3-18). All points at the black crossing curve except those belonging to the stable manifold (green curve) represent initial conditions of asymptotically unstable solutions (orbits) that finally follow the unstable manifold.

Some complicated nonlinear dynamic systems may display various attracting homoclinic structures in distinct regions of state space, resulting in several prevailing attractors (chaotic attractors, limit cycles) in those regions. The in-Lyapunov-sense stable solution $\mathbf{x}(t,\mathbf{x}_0)$ does not leave the region of its particular attractor; however, under some circumstances, such as a change in system parameters (in autonomous systems) or an external disturbance (in nonautonomous systems), the flow may drift toward a different attractor where it would not move otherwise. This transient behavior sometimes occurs without any internal or external system perturbations and is known as transient chaos. System perturbations propose a feasible reason for a flow to drift into a region of another attractor in a state space. If these changes are smooth, the transients among various distinct attractors might not be clearly distinguished and the system will seem to perform a single attractor behavior. This behavior is another type of multi-attractor behavior while the more obvious multi-attractor behavior in terms of coexisting attractors refers to systems with constant parameters and with sudden transitions among attractors [1].

3.1.3 Stability Analysis of Nonlinear Systems

Before dealing with the stability of HONNU as a non-linear system, it should be recalled that the stability of linear systems does not depend on the initial conditions, but only on the real values of the poles. In the case of the second-order non-linear dynamic system with variable damping in Eq.(3-19), where the nonlinearity can be considered a subset of dynamic quadratic neural unit (QNU) (or a plant controlled by a controller with variable damping, in terms of control theory [52]).

$$\ddot{x}(t) + k_v(1 - x^2)\dot{x}(t) + k_p x(t) = r(t) \tag{3-19}$$

The approach of motion in the vector field to asses the stability of such nonlinear dynamic systems can be used. Stability analyses of these systems (especially with technical tasks) often involve differential equations of the form, where the damping is not only a variable but also a function of position. The differential equations describing the dynamics of this type of nonlinear system can be expressed in a more general form as

$$\ddot{x}(t) + f(\dot{x}(t), x(t)) + \frac{\partial P(x(t))}{\partial(x(t))} = r(t)\,, \tag{3-20}$$

where $P(\ )$ is the potential function, and $\dot{x}(t)$ and $x(t)$ are the state variables of a system. The unity static gain in Eq.(3-19) assures zero steady state error and the damping assures the stability. It should be mentioned that the unity gain can be reached by either the CNU or the QNU controller once the plant has been identified [52].

For dissipative second order systems, we can obtain the derivative of energy function E whose derivative is given in [1] as

$$\dot{E} = \ddot{x}(t)\cdot\dot{x}(t) + \frac{\partial P(x(t))}{\partial x(t)}\cdot\dot{x}(t) = -f(\dot{x}(t), x(t))\cdot\dot{x}(t) \tag{3-21}$$

Unlike with the limited potential field approach, a stability analysis of any dynamic system of any order can be done if the Lyapunov function is found.

Theorem 1 :

If $\overline{\mathbf{x}}$ is an equilibrium point of the system of state differential equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ and there exists a Lyapunov function for $\overline{\mathbf{x}}$, then $\overline{\mathbf{x}}$ is stable. If a strict Lyapunov function exists, then the equilibrium $\overline{\mathbf{x}}$ is asymptotically stable.

The stability of a non-linear control system is assured if the state space region in which the system flows lies entirely within the basin of attraction. The morphological properties of the basin of attraction to the equilibrium are related to the properties of the strict Lyapunov function. The larger area of existence of the strict Lyapunov function, the larger basin of attraction to the equilibrium assured. Therefore, the existence of the strict Lyapunov function over as large an area as possible is to be followed to assure stability of a non-linear system.

In the case of non-linear second-order systems, the task simplifies, as the nonlinear differential equations describing the system can be expressed in state space representation as

$$\begin{aligned}\dot{x}_1(t) &= x_2(t)\\ \dot{x}_2(t) &= -x_1(t) - f\left[x_2(t), x_1(t)\right] + r(t).\end{aligned} \tag{3-22}$$

The Lyapunov function can be chosen then as

$$V(x_1, x_2) = \sum_{i=1}^{n} \frac{1}{2} \cdot x_i^{\,2} = \frac{1}{2} \cdot x_1^{\,2} + \frac{1}{2} \cdot x_2^{\,2}, \text{ where } n = 2, \tag{3-23}$$

where its derivative is given as

$$\dot{V}(x_1, x_2) = x_1 \cdot \dot{x}_1 + x_2 \cdot \dot{x}_2 = x_1 \cdot x_2 + x_2 \cdot (-x_1 - f(x_2, x_1)) = -x_2 \cdot f(x_1, x_2) = \dot{E}\,. \tag{3-24}$$

Eq.(3-21) and Eq.(3-24) show that the derivative of the Lyapunov function is equal to the derivative of the energy function for second-order nonlinear system in Eq.(3-22).

Figure 8 shows the flow of one stable and one unstable solution of an autonomous system in Eq.(3-19) for *k_p=1, k_v=2.34*, and the boundary of the basin of attraction is partly denoted by the stable trajectory converging to equilibrium$[0,0]$. Figure 9 shows the basin of attraction, denoted by '**o**', and the area of existence of the strict Lyapunov function (with vertical '**+**' stripes) in state space. The curves are nullclines denoting $\dot{x}(t) = 0$.

In subsection 7.2.2 or in [52], the application of the subset of the cubic neural unit to nonlinear state feedback control is shown demonstrating optimal performance of a controlled plant (Eq.(3-19)) from any initial condition for any desired value. The corresponding adaptive controller (Eq.(7-10), p.94) eliminates the system nonlinearity and forces the plant to behave with variable damping (Eq.(3-19)) by adaptable nonlinear state feedback.

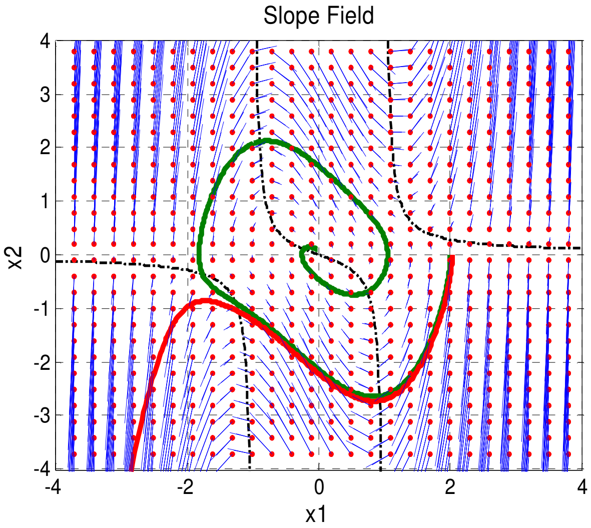


Figure 8: The slope field, flow of two solutions, nullclines, and limit cycle indicated in control Eq.(3-19) for $k_p=1,\ k_v=2.34$ in phase plane.

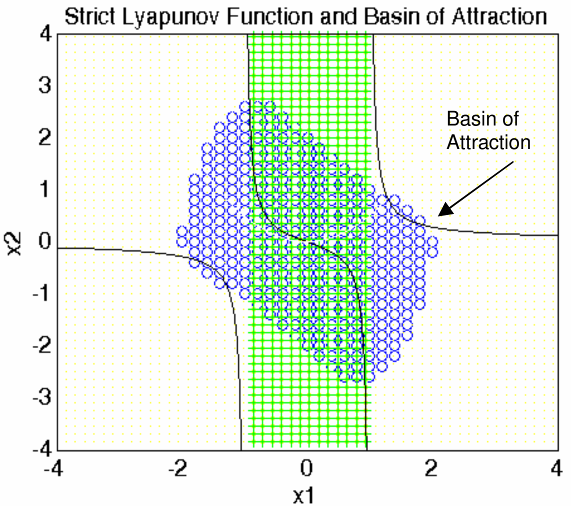


Figure 9: The stability region for system in Eq.(3-19) for $k_p=1,\ k_v=2.34$, area of decreasing energy and nullclines.

3.2 DETERMINISTIC CHAOS AS A SIGNIFICANT COMPONENT IN HEART RATE VARIABILITY DUE TO BEAT-BY-BEAT CONTROL OF THE AUTONOMOUS NERVOUS SYSTEM

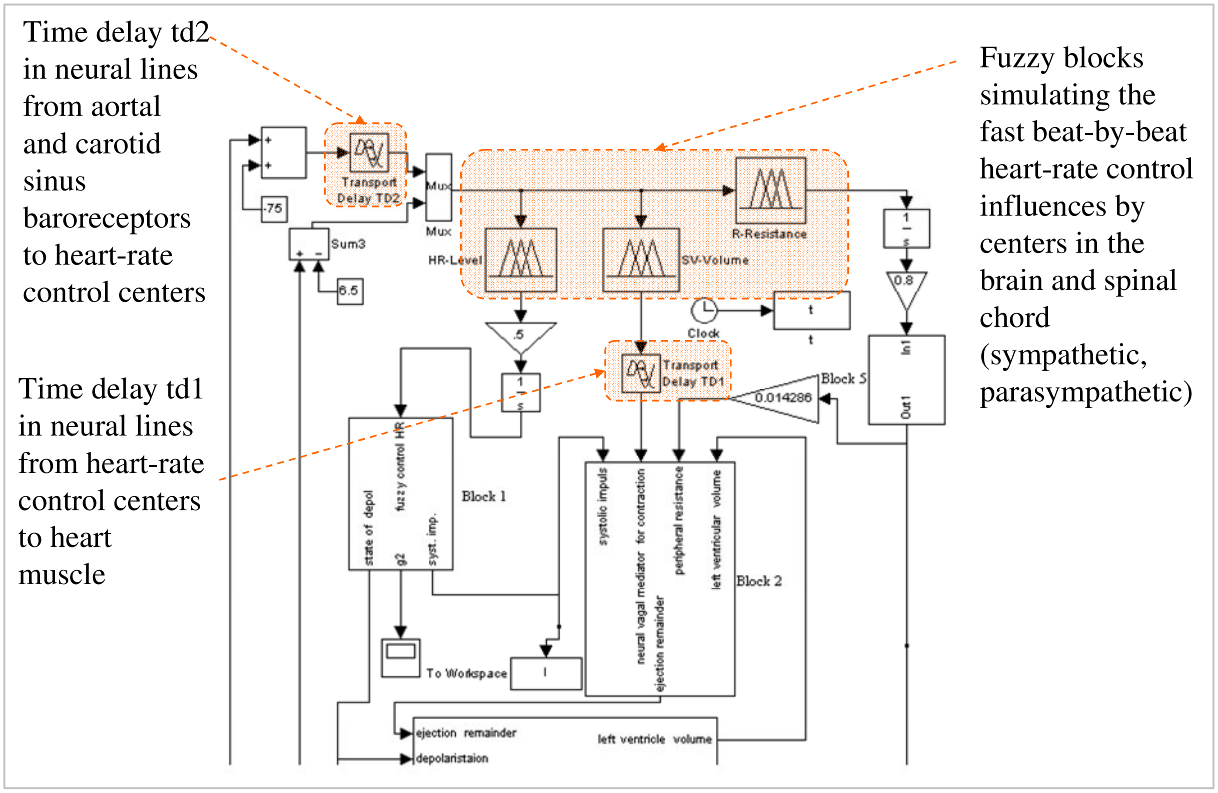
It is well known that even though the sinoatrial node itself is a very periodic pacemaker of heart rhythm, complicated heart rate variability is observed even with people in a relaxed state and a supine position. The complicated heart rate variability results from the complexity of the dynamic system which the cardiovascular system represents. The complicated heart rhythm is due to the many physiological control mechanisms of the cardiovascular system, due to inputs (perturbations) to a human cardiovascular system (mental activities, humoral influences, …), and due to fast beat-by-beat control influences of the autonomous nervous system (ANS). Evidence of the existence of a deterministic-chaos component in heart rate will be proposed below, based on simulation experiments and on the evaluation of heart rate variability [18] to [23]. In other words, evidence of the deterministic-chaos in heart rate will be shown to underline the important role of the ANS in developing the complicated behavior of heart rate, featuring high-level chaos, and including a considerable deterministic component caused by ANS feedback control [18] to [23]. Certainly, the fast control feedback influences of the ANS on heart performance are only one of the reasons for the development of complicated behavior of the heart rate. We propose that the deterministic contributions of the ANS to changes in heart rate variability are crucial, and that the novel diagnostic method can be based on tracking the dynamics of the deterministic heart rate variability component. Tracking the dynamics of the deterministic component of heart rate variability is made possible by the novel nonconventional neural architectures introduced in this work.

By elucidating both the physiological (fast beat-by-beat control influences of the ANS, Frank-Starling law, and peripheral resistance) and the technical principles (multiple-feedback control, transport delays) in a simplified model, [18] to [20], the existence of a significant chaotic component in heart rate will be concluded in this section. The focus will be on merging certain parallel aspects of both the physiological and the technical approaches, resulting in an understanding of the development of the deterministic-based chaotic component in heart rate.The reasons for the development of our novel method of evaluation of HRV (using artificial nonconventional neural units) for diagnostic purposes will be established.

Further, the results and conclusions are introduced in this section. They have been achieved during recent years of investigation and experiments using the model of the fast beat-by-beat control influences of the ANS on heart performance in a model of oxygen transport in the human cardiovascular system [18] to [20]. These results and conclusions support a hypothesis of the existence of a significant deterministic component in the chaotic behavior of heart rate variability [18] and provide us with an explanation of the development of the deterministic chaos component in a human cardiovascular system.

The human cardiovascular system is modeled [18] [20] as a simplified, deterministic, continuous-time, non-linear system whose parameter settings make it work in different modes, from periodic-like through quasi-periodical to highly irregular behavior with features of deterministic chaos featuring high state-space dimension. It corresponds to the fact that the correlation dimension (exponent) saturates at a high embedding dimension (>10 e.g.) for sufficient length of the data series.

The development of the involved model [18] is based on the assumption that the complicated HRV may significantly result from the multilple-feedback control influences of the ANS. This assumption has been based on experience in technical applications, where the multiple-feedback control essentially leads to quasi-periodical or even more irregular (chaotic) system behavior. The model of the blood circulation system [18] is based on principles of the fast physiological beat-by-beat control of the heart performance rather than on those of the slow influences, such as humoral control mechanisms. The actual heart rate (HR) of the myocardium is driven by the sinoatrial (SA) node, which works as the main pacemaker inside the heart, and whose electric impulses propagate through the myocardium and initiate heart contractions. Malfunction of the SA node or abnormal conducting in the paths of electric impulses in myocardium cause heart disorders that are usually easily detected by physicians via an ECG. In addition to the pacemakers inside the heart tissue, excitation of the heart is controlled by the vasomotor and cardio-inhibiting centers in the brain that represents the vagal feedback controllers. Among others, this dominant baroreflex control mechanism influences the stroke volume and peripheral resistance. There are other important control mechanisms in the model, such as the Frank-Starling law and the natural inotropy effect. The abnormalities in the HR caused by the external vagal multiple feedback control may be not so easily detected from an ECG signal as malfunctions of the internal pacemakers, where particular dysfunctions are visualized by corresponding abnormalities in the ECG wave complex. For example, the outage of the SA node can be easily detected immediately by the abnormal P wave in ECG signal. Conversely, abnormalities caused by obstacles in the vagal feedback may not correspond directly to the electrical potential of a particular part of myocardial tissue, and need to be revealed by other non-linear measures assessing the complex nonlinear behavior



Time delay td2 in neural lines from aortal and carotid sinus baroreceptors to heart-rate control centers

Fuzzy blocks simulating the fast beat-by-beat heart-rate control influences by centers in the brain and spinal chord (sympathetic, parasympathetic)

Time delay td1 in neural lines from heart-rate control centers to heart muscle

Figure 10: Schematics of the blood circulation system model [18] [19]; the main part contains sympathetic and parasympathetic control centers (implemented as Fuzzy blocks HR-Level, SV-Volume and R-Resistance), and transport delay blocks TD1 and TD2 represent actual physiological delays on ganglial neural lines.

The simplified model has implemented all important cardiovascular fast control mechanisms, including baroreflex, the Frank-Starling law, peripheral resistance, respiration influence, arterial blood pressure wave propagation, peripheral resistance, and venous preload. The following part of the section focuses on the features causing the model to generate distinct heart performance regimes, rather than concentrating on the fine details of the model. Readers interested in a more detailed description of the model implementation may refer to [18] and [19]. The model is implemented in Matlab/Simulink environment. The baroreflex is considered the main control mechanism in the modeling of the fast beat-by-beat control. The main pacemaker of the heart muscle is represented in the model as a special “saw-tooth” signal generator (block 1, Figure 10), which times the beats andor adjusts their amplitudes, i.e., performs the cyclic polarization and depolarization of the sinoatrial node. In our experiments, the time beating generator did not have implemented any internal myocardial dysfunction that would be caused by outages of SA node, nor had it any abnormalities in electric impulse conduction, nor had the model any random input at all. Similar to reality, the generator’s cyclic timing is subject to both kinds of neural activities: parasympathetic (vagal, prolonging the depolarization in case of increasing pressure and venous backflow) and sympathetic (shortening the depolarization in the opposite case). The systolic ejection volume is derived from the heart volume at the end of diastole (Frank - Starling law), and from the instantaneous HR (inotropy effect).

The sympathetic and parasympathetic control of the cardiovascular system are represented by three fuzzy blocks (Fig.1), which influence variables of the HR, stroke volume, and peripheral resistance. Each of these fuzzy blocks has the same basic structure of rules

$$\langle \text{IF}\langle(B_A \text{ is } \text{ AND } (B_V \text{ is } S)\rangle \Rightarrow \langle \text{THEN } \langle(U \text{ is } W)\rangle\rangle\rangle , \tag{3-25}$$

where BA and BV are signals that represent activities of the arterial (i.e., aortal and carotic) baroreceptors and arterial receptors respectively. U is the action variable which affects the values of variables HR, R, and SV. Symbols Q, S, and W denote fuzzy values (of the type Large Positive, Middle Positive, etc.) of fuzzy variables BA, BV and U. Fuzzy inference blocks are built up on three membership functions in the input and on five in the output, with symmetrical distributions. The inference outputs are delayed, corresponding to the actual delays in neural transfer facilities, then, they are converted to deviation signals of HR, stroke volume, peripheral resistance and venous volume via non-linear characteristics (look-up tables). These signals represent the control variables of the model. After a proper tuning, the parameter’s influence on the aforementioned non-linear and delayed feedbacks results in moderate irregular oscillations of aortal pressure, HR, venous preload and other variables in the vicinity of their ideal steady state points. This type of behavior has not been conditioned by the respiratory influence in the model. Even in the case of suspended respiring system, the system continued to exhibit this complex behavior.

Similar to the cardiovascular system in which the afferent and efferent neural lines cause the vagal control signals to be delayed during their transfer from carotic baroreceptors to the brain or from the brain to atrial receptors, transport delay blocks (TD1 and TD2 in Figure 10) have been implemented and used to set various courses of vagal control under which the HRV has been investigated. Only these two (external) transport delays are employed for altering heart performance complexity in the model; unlike the organic system where other “external” mechanisms, e.g., humoral control, and “internal”, e.g., disturbed the sinoatrial or atrioventricular node, are present and make cardiovascular system more irregular.

Results of the research on this topic are not too unified [6] to [14]. By investigation of deterministic chaos in heart rate, which also points to difficulties in reaching saturation of the correlation dimension (CD) that violates the worth of this method, we hope to provide a brief look at the model’s behavior and compare it to real myocardial disorders whose invariants were computed with the same basic algorithms.

The changes in heart rate variability for distinct settings of time-delays on the ganglial neural lines of simulated influences of the autonomous nervous system (Figure 10) were estimated as

$$C(r) = \frac{1}{N} \cdot \sum_{i=1..n-m}^{n-1} \sum_{j=i+1..n}^{n} \Theta(r - |x_i - x_j|), \tag{3-26}$$

where CD is the correlation dimension, $C(r)$ is the correlation function, x_i is a sample at time *i*, Θ is the Heaviside function, and r>0 is a radius in phase space [1]. Because of the difficulties in reaching the saturation of the CD with the G-P method [3] for R-R diagrams from the model and from the disorder database [17], the results are more for loose comparison than for exact consideration. In other words, the characteristic invariants evaluated below should reflect changes in the complexity of the HRV of the model and of database data rather than showing their exact values. The CD was also checked by the program Dataplore (Datan, GMBH, trial version), even though the saturation of the correlation dimension was rarely reached for the data from simulations of one-hour physiological performances. LLE were also calculated by Dataplore. For all the computations, 4096 samples were used, which corresponds to one-hour of ECG recording.

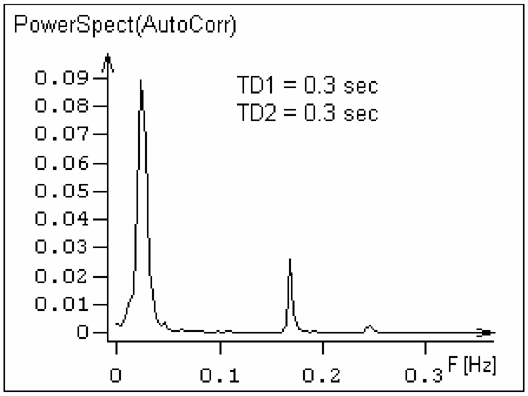


Figure 11: Power spectral density of simulated R-R diagram for TD1=TD2=0.3s. Estimated invariants: CD∈(3,4), LLE∈(0.5,1).

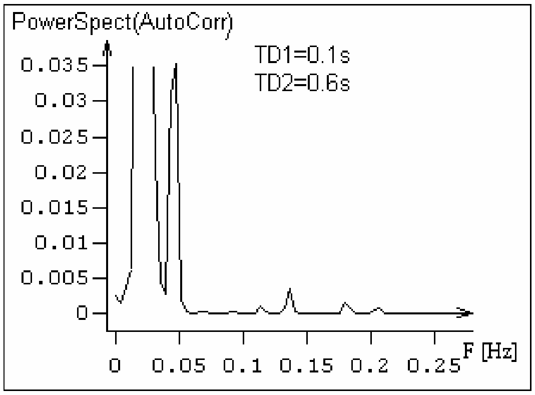


Figure 12: Power spectral density of simulated R-R diagram for TD1=0.1s, TD2=0.6s. Estimated invariants: CD∈(1,2), LLE∈(0.5,1).

Figure 13 to Figure 15 show the capability of the transport delays in afferent and efferent neural lines to modify the power spectra of the generated HR, where increasing irregularity of the HR, from periodic-like behavior to much more irregular types, is observed. The shape of the power spectra curve can be significantly modified by the delays. Through more sensitive and proper tuning of the model (of TD1, TD2 and, if necessary, by fine tuning the fuzzy blocks representing brain interventions), it can reflect more accurately the curve of a healthy person.

Seeking the origins of chaotic HRV through the principles of system non-linearity, and through the introduced transport delays in the neural feedback, together with the estimation of characteristic non-linear invariants, is one of the main goals of this section.

Figure 17 serves to verify the results shown in Figure 18 because the saturation of the CD (calculated by Dataplore) was reached rarely for the dark areas in Figure 17 to Figure 19 (see Figure Appendix - 18).

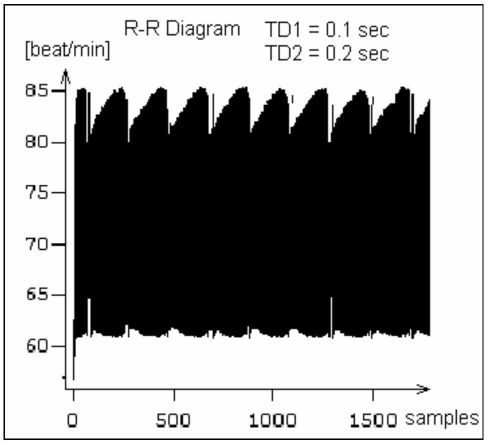


Figure 13: A sample of heart beat frequency of simulated data for settings of TD1=0.1s, TD2=0.2s. Some periodicity can be observed. Estimated invariants: CD∈(0,1),LLE∈(0,0.5)

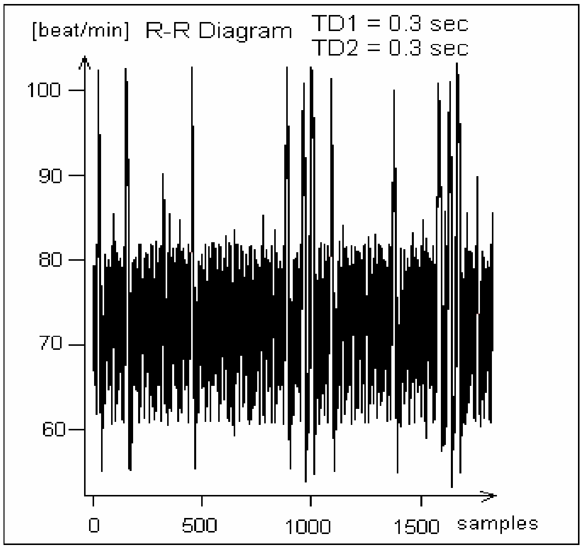


Figure 14: A sample of heart beat frequency of simulated data for settings of TD1=0.3 s, TD2=0.3. The HRV increase is observable contrary to (Fig.5). Estimated invariants: CD∈(3,4), LLE∈(0.5,1).

It has been proved that the model increases the irregularity of the heart rate with the particular various constants of the neural transport delays. The introduced model is deterministic (with no random inputs or run-time changing parameters) and shows the ability to bifurcate, i.e., it can increase the behavioral irregularity from periodic-like to quasi-periodical. Behavior can also range up to the regimes where saturation of the correlation exponent is not reached even for the embedding dimension *m*=15 for the used length of samples. We believe that deterministic chaos in heart rate has been modeled by a nonlinear deterministic model and can be used in further research on the chaotic dynamics of HR, even in real physiological recordings.

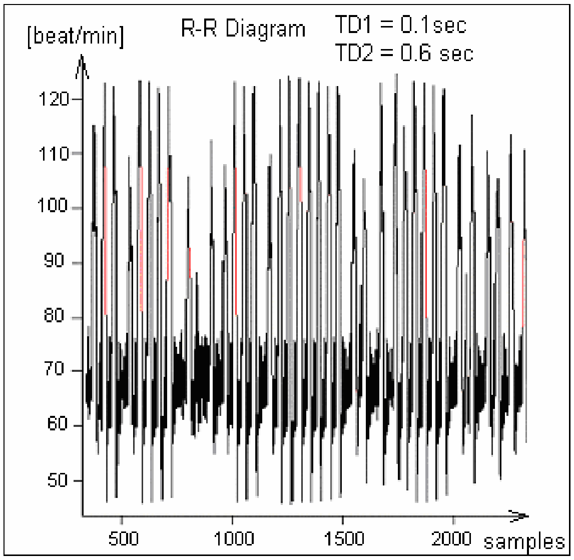


Figure 15: A sample of heart beat frequency of simulated data for settings of TD1=0.1s, TD2=0.6. Again, the HRV irregularity increase is observed in comparison to Figure 13. Estimated invariants: CD∈(1,2), LLE∈(0.5,1).

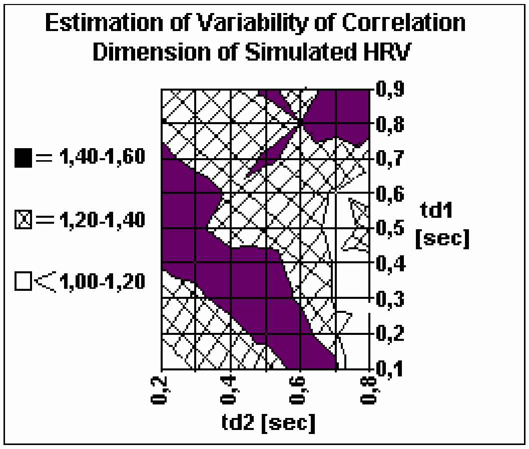


Figure 16: Simple estimation of locations with higher and lower CD of heart rate as a function of the configuration of transport delays in the neural control feedback loop. Estimated by Eq.(3-26).

If the parameters of vagal transport delays were being changed during the run, the HRV would vary, and the attractor of the heart rate would vary, and multiple attractor behavior would be observed, similar to the human cardiovascular system. In comparing the complexity of real disorder signals and model data, it can be assumed that the multi-attractor behavior of the heart might be caused by the intrinsic complexity of multilevel control feedbacks in an organism. The periodic-like behavior seen in patients after transplants [8] [9] corresponds to the decreased sensibility of the heart due to those controlling influences (vagal and humoral).

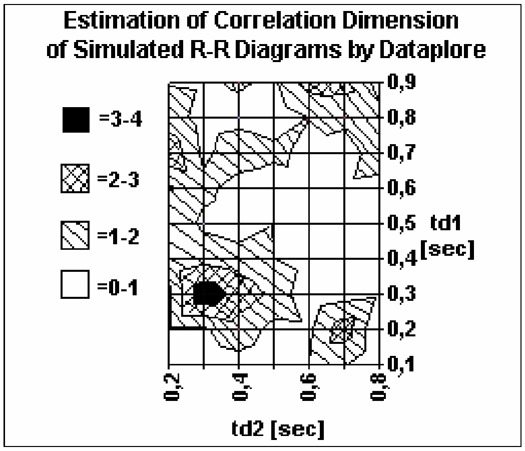


Figure 17: Estimation of the correlation dimension by the program Dataplore. Values fluctuate between the range of values characteristic of cardiac disorders (Figure 19) to the range typical of healthy subjects [9] [12].

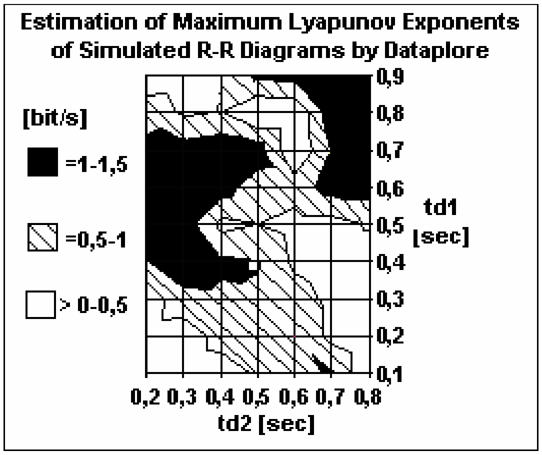
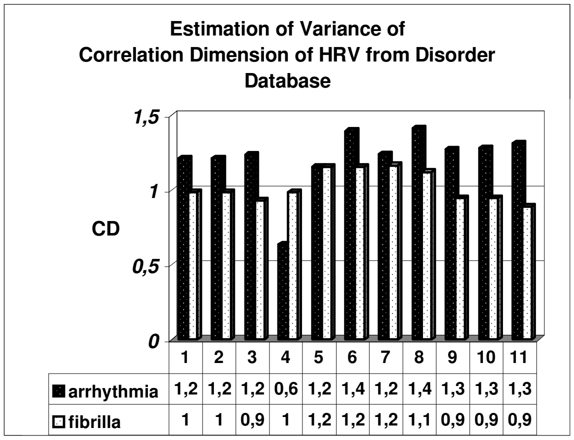


Figure 18: Estimation of largest Lyapunov exponents by Dataplore.

Therefore, if low-dimensional chaos appears, it could be due to an improper function of the neural system influencing heart rate by vagal and humoral interventions, when the heart is not so sensitive as before transplant, but it is more sensitive than after the transplant. As based on observations from the model, the presence of low-dimensional chaos in HRV (Figure 12, Figure 13 and Figure 15) might indicate some pathological phenomena due to some dysfunction of the neural control of the heart while higher dimensional (Figure 11 and Figure 14) or even multiple-attractor behavior [14] [26] could correspond to a healthy state.



	1	2	3	4	5	6	7	8	9	10	11
■arrhythmia	1,2	1,2	1,2	0,6	1,2	1,4	1,2	1,4	1,3	1,3	1,3
□fibrilla	1	1	0,9	1	1,2	1,2	1,2	1,1	0,9	0,9	0,9

Figure 19: Example of the correlation dimension of the R-R diagrams of subjects suffering from cardiac disorders estimated by the same method by Dataplore. Data come from MIT-BIH Database. From the charts above, locations of higher and lower complexity of HRV can be observed. The range of values of CD by Dataplore (Figure 17) corresponds to results from real signals presented in [9].

3.3 BIOLOGICAL NEURONS AND UNIVERSAL APPROACHES OF MATHEMATICS

Today, neurologists assume that neurons possess more computational power than previously thought (see page 16 above). In this section, the technical way of describing dynamic systems with functional capabilities expected by modern physiologists of biological neurons is revealed. The following subsections show that the well known and commonly used mathematical (technical) notation of complicated dynamic systems, i.e., the state space representation with nonlinearities and time-delays (which have been successfully applied to technology, such as in the approximation of complex systems, control) agrees ideas about the computation capabilities of biological neurons. This new comparison provides insight into biological neurons by paralleling their known structure to the universal achievements of mathematics. The comparison is made here as well as explored further in sections 4.1 and 4.1.2.

3.3.1 Biology Inspired by Mathematics and Technology - The Inverse Inspiration

It is interesting from the philosophical point of view that today we observe an inverse merge of technical concepts with the nature of biological systems. Generally, technical or mathematical researchers were inspired by biological systems (neural networks, genetic algorithms, immune-based controllers). As the technology and mathematics advances further; however, useful and successful principles have emerged. We can see, that the commonly used technique for handling dynamic systems by engineers and mathematicians could correspond to how modern researchers look at a biological neuron. For example, the simple case of state-space representation shown in Eq.(3-27) can describe the higher computational capability of a neuron, and biological parallels can be drawn, as introduced in subsection 4.1.2.

$$\begin{aligned} \frac{dx(t)}{dt} &= f(x(t),\mathbf{u}(t)) \\ y(t) &= g(x(t)) \end{aligned} \tag{3-27}$$

where x is the internal state variable, y is the output variable, $f(\,)$ and $g(\,)$ are static functions, $\mathbf{u}$ is a vector of inputs, and t is a continuous variable of time.

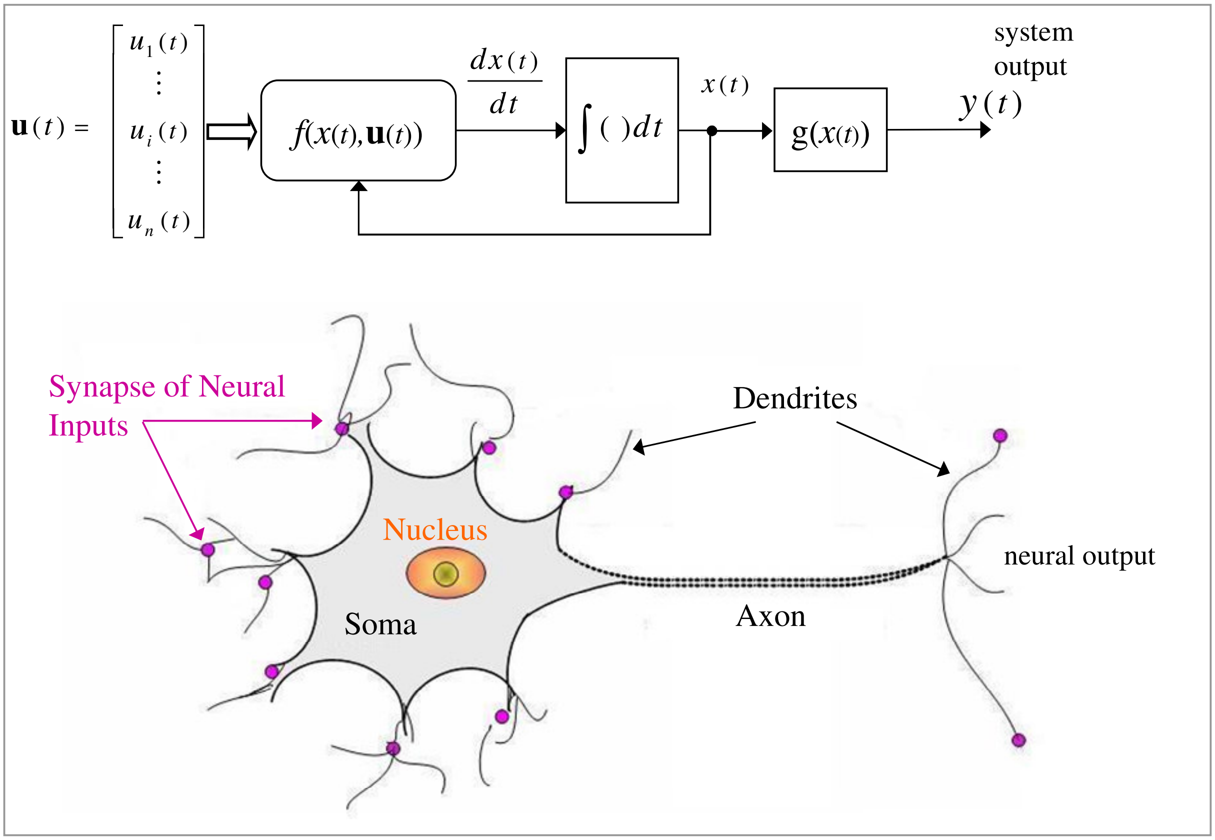


Figure 20: The state-space representation, Eq.(3-27), is a common way in which technicians understand nonlinear dynamic systems. Sections 4.1 and 4.1.2 show how state-space repesentation can correspond to how current researchers could look at a biological neuron; i.e., assuming higher computational capability.

In further subsections, the phenomena of nonlinearity and time delays, powerful in their ability to approximate state-space models, will be deduced as present in not only the synaptic but also the somatic neuronal part. This conception introduces new understanding of HONNUs and possibly of biological neurons as well. These novel proposals are made with respect to most universally applicable approaches of mathematics for solutions into deterministic dynamic systems as indicated in this subsection. It will be shown that the interpretation of the nonlinearity phenomenon in the synaptic and somatic parts of a neuron may depend on the static or dynamic nature of the artificial neural architecture, i.e., on the static or dynamic HONNU. The importance of both the synaptic and the somatic neuronal part will be introduced and emphasized, especially for the class of dynamic HONNU.

3.3.2 Nonlinear Synapses and Static HONNU

Static HONNUs are suitable as solutions to static problems or for finding initial neural weights for dynamic HONNUs, as will be shown further in this thesis. From the mathematical notation of static HONNU, one can observe its biological analogy as shown in Figure 21 of this subsection[8].

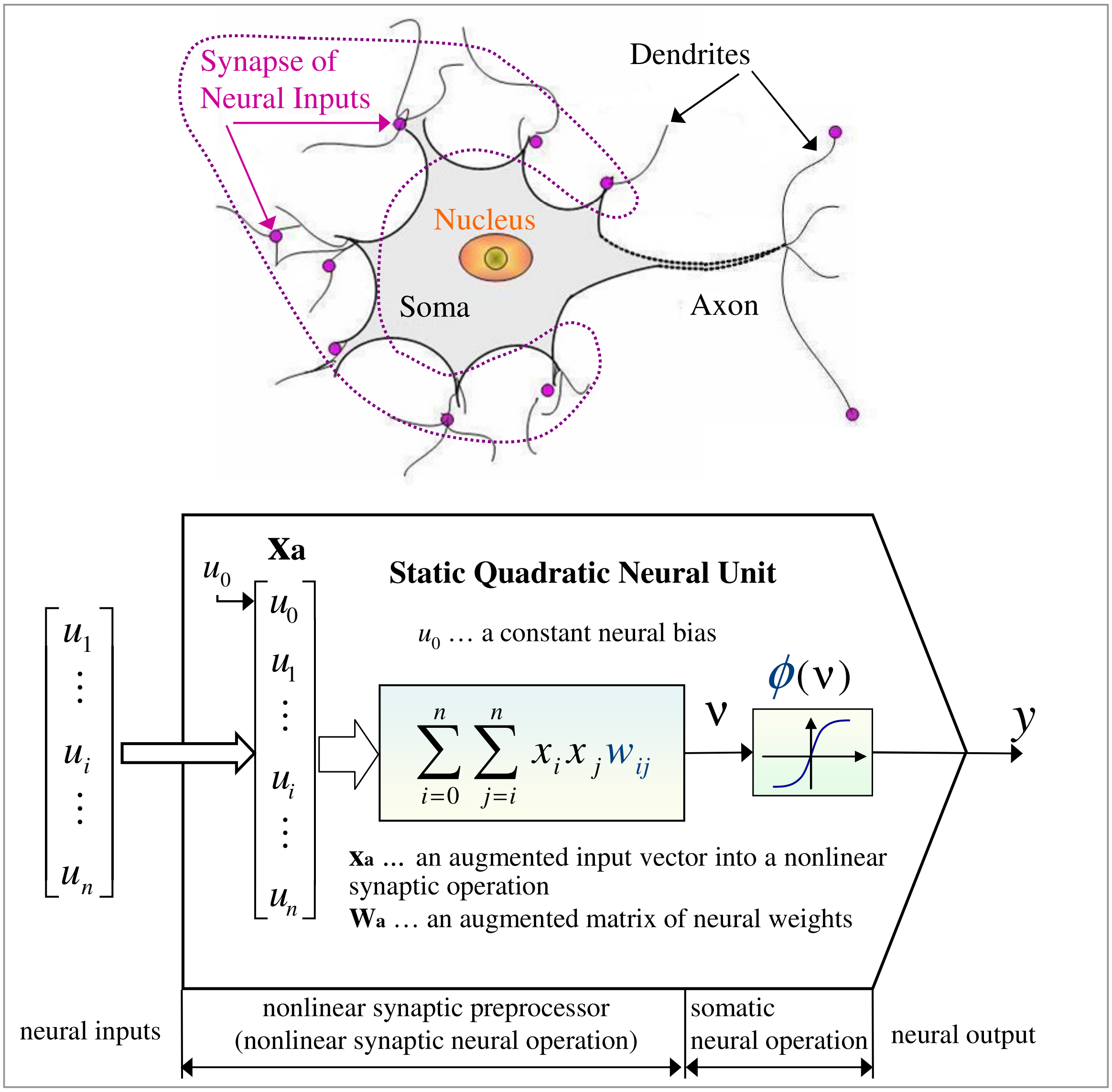

Synapse of
Neural Inputs
Dendrites
Nucleus
Soma
Axon
Xa
u_0
Static Quadratic Neural Unit
u_0 … a constant neural bias
$\phi(\nu)$
ν
y
u_1 ⋮ u_i ⋮ u_n
u_0 u_1 ⋮ u_i ⋮ u_n
$\sum_{i=0}^{n}\sum_{j=i}^{n} x_i x_j w_{ij}$
Xa … an augmented input vector into a nonlinear synaptic operation
Wa … an augmented matrix of neural weights
neural inputs
nonlinear synaptic preprocessor
(nonlinear synaptic neural operation)
somatic
neural operation
neural output

Figure 21: Structure of the static quadratic neural unit where quadratic polynomial is understood as merely a synaptic nonlinear preprocessor of the neural inputs (QNU, Gupta *et.al.* 2003);

[8] Further in the development discussion in chapter 4, the mathematical notation of static as well as dynamic HONNU, will be shown, as it provides us with a more insightful interpretation of its biological analogy.

Figure 21 shows the general neural architecture of static HONNUs where f_{HONNU} is a higher-order polynomial function or other nonlinear and piecewise smooth function[9], ν is the output of a synaptic neural operation, $\mathbf{x_a}$ is an augmented input vector into a nonlinear synaptic operation, $\mathbf{W_a}$ is an augmented matrix of neural weights, $\mathbf{u}$ is vector of neural inputs, u_0 is a constant neural bias, $\phi(\ )$ is a somatic neural operation and y is the neural output. In fact, Figure 21 (Gupta et al., 2003 [49]) is generalization of Figure 5 on page 16 and underlines the inspiration-from-biology approach as introduced in chapter 2 - State of the . The inverse approach is followed in further subsection that is concluding the analogies to a biological neuron upon inspiration from mathematic notation generally successful in technical applications. In other words, it will be shown how the state-space representation of nonlinear and time-delayed dynamic systems can correspond to the functionality of a biological neuron and thus it results in further development of HONNU architectures proposed in this work.

Further interpretation of the merge of mathematical and biological understanding to the dynamic HONNU and TmD-DNU is shown in subsection 4.1.2 as it belongs to the chapter on developments of this thesis.

[9] More precisely, a function that is piecewise smooth (piecewise differentiable) with respect to adaptable neural parameters (neural weights, adaptable time-delays, adaptable dynamic parameters).

4 Development of Nonconventional Artificial Neural Architectures: HONNU and TmD-DNU

In this section, artificial neural architectures called higher-order nonlinear neural units (HONNU), time-delay dynamic neural units (TmD-DNU), and from-their-combination resulting time-delay dynamic higher-order nonlinear neural units (TmD-DHONNU) will be introduced. They represent a novel class of neural architectures suitable for system approximation with both the minimum number of neural parameters and a simple internal architecture. The use of these units avoids the "gray-box" or even the "black-box" modeling effect of conventional neural networks because the internal mathematical structure of these neural models can be further investigated relatively easily. The complexity of conventional neural networks is reduced by the increased computational power of the nonconventional neural units developed and proposed in this work. Further in this chapter, the concept of nonconventional neural architectures called higher-order nonlinear neural units (HONNU) is discussed further. Contrary to conventional artificial neural units (neurons) (Figure 1) with linear aggregating function and their common network architectures (Figure 3, e.g.), the basic structures of HONNU have higher-order polynomials or other customizable nonlinear terms in the input-aggregating function f_{HONNU}, where internal neural state variables are also aggregated in their dynamic versions (e.g. TmD-DHONNU). Especially in subsections 4.1.1 and 4.2, the conception of the neural aggregating function is proposed. It is shown how both the synaptic junctions as well as the somatic part of a biological neuron can be understood as computationally active while maintaining both the computational power of a neuron and a simple mathematical structure. A technique of practical application of HONNU to system approximation and the monitoring of changes in dynamics is summarized in section 5.

4.1 STATIC HONNU

In this section, static HONNU with nonlinear aggregating function of neural inputs f_{HONNU} are introduced. It is indicated that the full notation of a polynomial aggregating function emphasizes computational capability of soma (and possibly nucleus) when compared to biological structure of neurons. Thus it may explain higher computational capabilities of biological neurons.

4.1.1 Synaptic and Somatic Neural Operation and Static HONNU

In this section, the class of novel neural units called HONNU, designed by M.M. Gupta (Figure 21) is further analyzed, and a theoretical proposal is made about the nonlinear aggregating function $f_{HONNU} = \nu$ of neural inputs u_i. Also a parallel with the biological counterpart of HONNU is drawn.

Recall, that Figure 4 on page 15 showed one of the conventional neural units that has incorporated single-argument nonlinear functions on neural inputs; the argument is the particular neural input (dendrite), and inputs are further linearly aggregated by weighted summation. Figure 21 shows QNU designed by M.M. Gupta *et al.* [49] [50] [51], where the nonlinear aggregating function of neural inputs is considered to be a nonlinear and merely synaptic preprocessor. The design of static HONNU (Figure 21) is still very simplified, but it significantly increases the computational power of a single artificial neuron. Below, it is proposed that not only does the nonlinear aggregating function of HONNU represent a pure synaptic preprocessor, but it should be viewed as a nonlinear aggregation of both neural synaptic and somatic operations. The aggregating function of neural inputs of HONNU can be understood as the interaction of neural synapses plus somatic neural operation according to HONNU's mathematical conception. This proposal is based on a theoretical deduction and agrees with the mathematical concept of static HONNU, which are good universal approximators of complex functions (see below). Let us imagine that only a single dendrite is connected to each nonlinear neural synapse junction as shown in Figure 22.

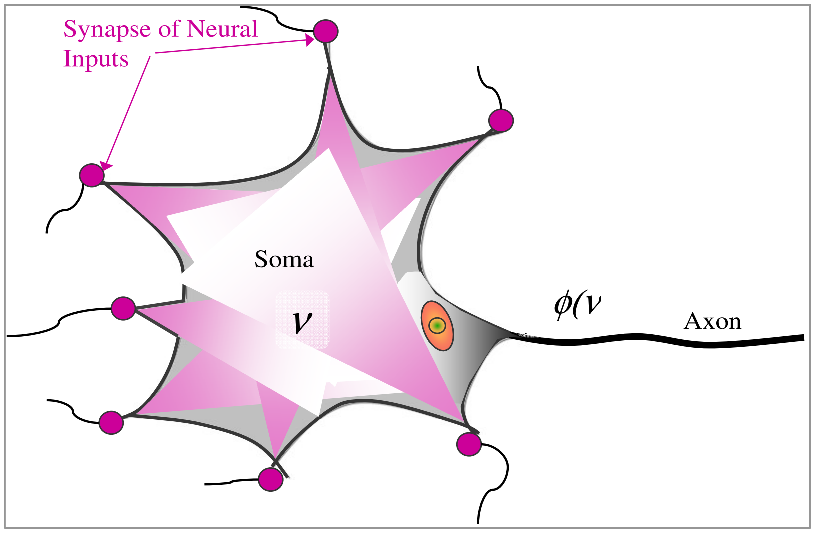


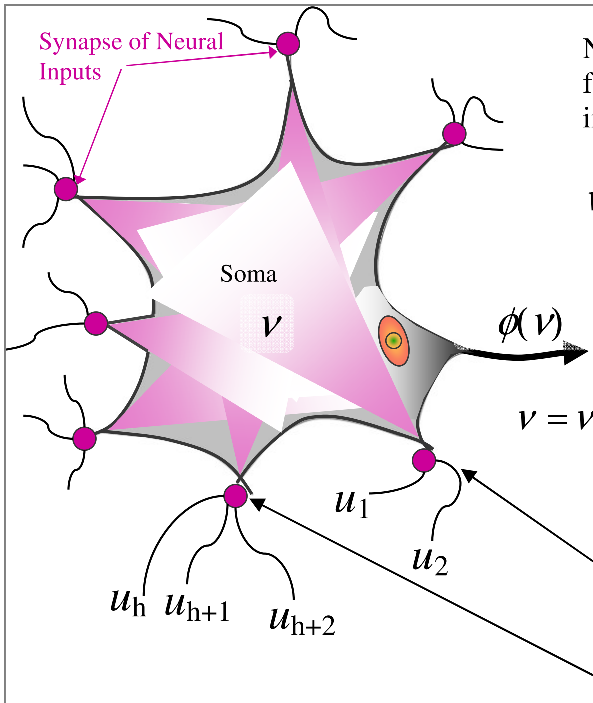
Figure 22: A simplified neuron with single dendrite on synaptic junction with somatic operation incorporating further interactions of neural inputs. The biological parallel of the decomposition of the aggregating function into synaptic and somatic part for quadratic neural unit (QNU) is in Eq.(4-1).

The aggregating function of the simplified single-dendrite neuron shown in Figure 22 is decomposed for the case of the quadratic neural unit in Eq.(4-1) as

$$
\begin{aligned}
\nu = f_{QNU} &= \nu_{synaptic} + \nu_{somatic} = \sum_{i=0}^{n}\sum_{j\geq i}^{n} u_i u_j w_{ij} = \\
&= \sum_{i=1}^{n}\left(w_{0i}u_i + w_{ii}u_i^{2}\right) \quad + \quad w_{00} + \sum_{i=1}^{n}\sum_{j>i}^{n}\left(w_{ij}u_i u_j\right) \quad = \\
&= \text{nonlinear aggregation of neural inputs on synaptic junctions } + \\
&+ \text{ nonlinear somatic operation (simplified effect of further neural input interactions} \\
&\quad \text{inside soma and nucleus that affects the signal transmitted into axon)}
\end{aligned}
\tag{4-1}
$$

where w_{ij} are neural weights, u_{ij} are neural inputs (dendrites of other neurons), and u_0 is neural bias.

To mimick a real neuron as in Figure 23, its synaptic junctions should incorporate more dendrites of other neurons.



Nonlinear somatic operation incorporating further influences and correlations of neural inputs; e.g., for CNU could instead be given as

$$\nu_{somatic} = w_{000}u_0 + \sum_{i=1}^{n}\sum_{j>i}^{n}\sum_{k>j}^{n} w_{ijk}u_i u_j u_k \cdot \alpha_{ijk}$$

where α_{ijk}=0 if the combination *ijk* has been already included in ν_{synaptic}

$$\nu = \nu_{synaptic} + \nu_{somatic} = \sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} w_{ijk}u_i u_j u_k$$

Aggregation of neural inputs of CNU on a synaptic junction with two dendrites:

$$w_{001}u_1 + w_{002}u_2 + w_{011}u_1^{\,2} + w_{012}u_1u_2 + w_{022}u_2^{\,2}$$
$$+w_{111}u_1^{\,3} + w_{112}u_1^{\,2}u_2 + w_{122}u_1u_2^{\,2} + w_{222}u_2^{\,3}$$

Nonlinear aggregation of neural inputs on the three-input synaptic junction of CNU :

$$\nu_{synaptic\ r} = \sum_{i=h}^{h+2}(w_{00i}u_i) + \sum_{i=h}^{h+2}\sum_{j=i}^{h+2}(w_{0ij}u_iu_j) + \sum_{i=h}^{h+2}\sum_{j=i}^{h+2}\sum_{k=j}^{h+2}(w_{ijk}u_iu_ju_k)$$

Figure 23: A simplified neuron with somatic operation incorporating further interactions of neural inputs; the biological parallel of the decomposition of the aggregating function into synaptic and somatic part for cubic neural unit (CNU).

Of course, more than one dendrite may be connected to each synaptic point of a biological neuron as shown in Figure 23. Then the decomposition of the aggregating function into synaptic and somatic parts is not as clearly apparent as in the case of the single-dendrite neuron shown in Figure 22 and Eq.(4-1), and this would be rather a research of another field of science.

However, mathematically this decomposition does not have to be considered when the concept of HONNU is introduced because it is aggregated within a single polynomial function $\nu = \nu_{\mathrm{synaptic}} + \nu_{\mathrm{somatic}} = f_{HONNU}$ and the neural weights of this aggregating function are found by the cognitive nature (learning algorithm) of the unit. With regards to understanding HONNU as function approximators, one way is through a function approximation by the Taylor polynomial. In principle, a quadratic neural unit, shown in Figure 5, is capable of identifying a nonlinear function by approximating it with the Taylor polynomial of the second order. Similarly, a cubic neural unit is capable of identifying a nonlinear system by approximating it with the Taylor polynomial of the third order. The relationship between HONNU and the Taylor polynomial is indicated in Eq.(4-2).

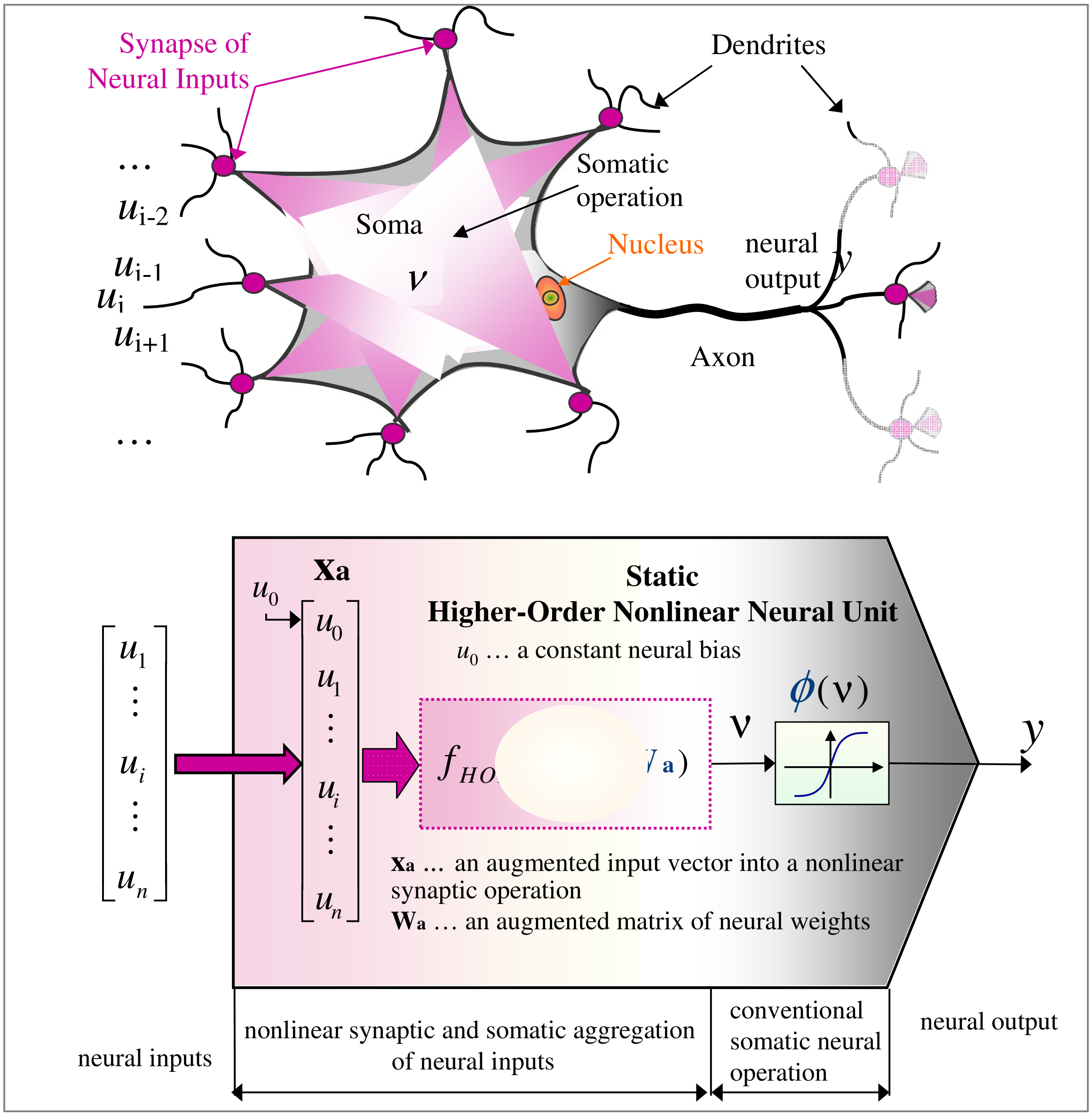

Synapse of
Neural Inputs
Dendrites
…
u_{i-2}
u_{i-1}
u_i
u_{i+1}
…
Somatic
operation
Soma
ν
Nucleus
neural
output
y
Axon
u_1
u_i
u_n
Xa
u_0
u_0
u_1
u_i
u_n
Static
Higher-Order Nonlinear Neural Unit
u_0 … a constant neural bias
f_{HO}
a)
ν
$\phi(\nu)$
y
Xa … an augmented input vector into a nonlinear synaptic operation
Wa … an augmented matrix of neural weights
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs
conventional somatic neural operation
neural output

Figure 24: General structure of a static higher-order nonlinear neural unit (HONNU), where polynomial is understood as simplified realization and interaction of both synaptic preprocessor of neural inputs and somatic neural operation

$$f(\mathbf{x}) \cong f(\mathbf{x}_0) + \sum_{i=1}^{n} \frac{\partial f(\mathbf{x})}{\partial x_i}(x_i - x_{i0}) + \frac{1}{2}\sum_{i=1}^{n}\frac{\partial^2 f(\mathbf{x})}{\partial x_i{}^2}(x_i - x_{i0})^2 + \frac{1}{3!}\sum_{i=1}^{n}\frac{\partial^3 f(\mathbf{x})}{\partial x_i{}^3}(x_i - x_{i0})^3 + \cdots$$

$$f(\mathbf{x}) \cong w_0 x_0 \; + \; \sum_{i=1}^{n} w_i x_i \; + \; \sum_{i=1}^{n}\sum_{j=i}^{n} w_{ij} x_i x_j \; + \; \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} w_{ij} x_i x_j x_k \; + \ldots$$

$$-----\textit{quadratic neural unit}--\textit{QNU}-----| \qquad (4\text{-}2)$$

$$-------------\textit{cubic neural unit}----\textit{CNU}-----------|$$

$$----------\textit{further higher}-\textit{order nonlinear neural units}--\textit{HONNU}--\ldots \; ;$$

where f denotes the nonlinear aggregating function f_{HONNU} as shown in Figure 24 and the derivatives are evaluated at $\mathbf{x}=\mathbf{x}_0$. The concept of the Taylor polynomial approximation is only one way to understand the nonlinear synaptic operation of HONNU in mathematical terms. A nonlinear synaptic operation may be of various structures and can include different types of nonlinearities (piecewise differentiable for the gradient learning backpropagation to be realized); thus, the synaptic operation of HONNU can be modified to fit a particular application.

4.1.2 Matrix Notation of Nonlinear Aggregating Function

The matrix calculus is widely used for analyzing important properties of various systems, such as the stability assessment. Therefore, the availability of the matrix notation of synaptic operation of HONNU Eq.(4-5) can become an important feature of HONNU, especially when the internal static and dynamic structure of HONNU is to be standardized in this work. In this chapter, we develop the matrix notation for cubic neural units as well as for even higher-order neural units with higher-order polynomials in their synaptic operation. Such notation could be useful for further analysis of HONNU properties in the future. Further development of such 'nonlinear-system-matrix' calculus focused on analyzing of static as well as dynamic HONNU exceeds the framework of this thesis and would deserve another proper research work.

The quadratic neural unit (QNU) is the simplest representative of the class of artificial neural architectures that we propose, further develop, and standardize both their notation and structure in this work, and that are called higher-order nonlinear neural units (HONNU). The simplest representation of the full synaptic operation of QNU (Gupta *et.al.*, [49] to [53]) is recalled in Eq.(4-3) as

$$\begin{aligned} v_{Quadratic} &= f_{QNU}(\mathbf{xa},\mathbf{Wa}) = \sum_{i=0}^{n}\sum_{j=i}^{n} w_{ij}x_i x_j \in \boldsymbol{R}^{\mathbf{1}}, \\ &\text{where } x_0 = 1 \qquad . \end{aligned} \tag{4-3}$$

where s is the number of neural inputs, x_0 is neural bias, $\mathbf{xa}$ is a bias-augmented vector of neural inputs into the synaptic neural operation f_{HONNU} (including f_{QNU} and f_{CNU}), and $\mathbf{Wa}$ is the augmented matrix of neural weights as shown in Eq.(4-4)

$$\mathbf{xa} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{Wa} = \begin{bmatrix} w_{00} & w_{01} & w_{02} & \cdots & w_{0n} \\ 0 & w_{11} & w_{12} & \cdots & w_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & w_{nn} \end{bmatrix}. \tag{4-4}$$

The synaptic neural operation of QNU in Eq.(4-3) can be also expressed in matrix notation as shown in Eq.(4-5)

$$v_{Quadratic} = \mathbf{xa}^T \cdot \mathbf{Wa} \cdot \mathbf{xa} \in \boldsymbol{R}^{\mathbf{1}}. \tag{4-5}$$

The full synaptic operation of Cubic Neural Unit (CNU) can be represented in a summation form as

$$v_{Cubic} = \sum_{i=0}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} w_{ijk} x_i x_j x_k =$$

$$= w_{000} x_0^{\;3} + w_{001} x_0^{\;2} x_1 + \cdots + w_{123} x_1 x_2 x_3 + w_{133} x_1 x_3^{\;2} + \cdots + w_{n-1nn} x_{n-1} x_n^{\;2} + w_{nnn} x_n^{\;3}, \;. \qquad (4\text{-}6)$$

where $x_0 = 1$.

The summation in Eq.(4-6) cannot be interpreted using common vector and two-dimensional matrix multiplications, as in the previous case of QNU in Eq.(4-5), where two instances of an augmented vector and an augmented matrix are used. Therefore, the appropriate multidimensional matrix notation should be used to express the synaptic operation of CNU or any higher-order of the polynomial, i.e., the synaptic operation in matrix notation. The matrix notation of the synaptic operation of a full CNU represents an interesting theoretical task that is not yet common in literature and that was solved in this thesis.

$$v_{HigherOrder} = \sum_{i=0}^{n} \sum_{j=i}^{n} \cdots \sum_{q=p}^{n} \sum_{r=q}^{n} w_{ij\cdots qr} x_i x_j \cdots x_q x_r =$$

$$= w_{00\cdots0}\, x_0^{\;n} + w_{00\cdots1}\, x_0^{\;n-1} x1 + \cdots + w_{n-1n\cdots n}\, x_{n-1} x_n^{\;n-1} + w_{nn\cdots n} x_n^{\;n}, \qquad (4\text{-}7)$$

where $x_0 = 1$.

The multidimensional matrix multiplication may be considered difficult to visualize. Below, an auxiliary mathematical technique is developed to allow us to perform multidimensional matrix multiplication in order to develop matrix notation for higher-order synaptic operation of HONNU. First, additional operators for matrix multiplications of higher-dimensional arrays need to be defined, ie., 3-D, 4-D or higher.

Operator [4-1]: Let the small dot "·" or no sign represent an operator for common two-dimensional matrix multiplication operator, as if two matrices m × n and n × p were multiplied and a two-dimensional matrix m × p would result.

Operator [4-2]: Let the large dot "•" represent multidimensional matrix multiplication operator analogous to the multiplication of a two-dimensional matrix m × n by a scalar b; however, where the elements of the two-dimensional matrix m × n are vectors or matrices and the element b becomes vector or matrix as well (**b**→b), (see example in Eq.(4-10)).

Operator [4-3]: Let the cross in a circle "⊗" represent multidimensional matrix multiplication operator analogous to the multiplication of two two-dimensional matrices m × n and n × p, where the elements of the matrices are vectors or matrices as well, (see example in Eq.(4-11)).

Let us denote common matrices and vectors as

$$\begin{aligned} \mathbf{A_1} &= \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, \quad \mathbf{A_2} = \begin{bmatrix} a_5 & a_6 \\ a_7 & a_8 \end{bmatrix}, \quad \mathbf{b_1} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\ \mathbf{A_3} &= \begin{bmatrix} a_9 & a_{10} \\ a_{11} & a_{12} \end{bmatrix}, \quad \mathbf{A_4} = \begin{bmatrix} a_{13} & a_{14} \\ a_{15} & a_{16} \end{bmatrix}, \quad \mathbf{b_2} = \begin{bmatrix} b_3 \\ b_4 \end{bmatrix} \;. \end{aligned} \qquad (4\text{-}8)$$

Examples of multidimensional matrices are then

$$\mathbf{A}=\begin{bmatrix}\mathbf{A_1} & \mathbf{A_2}\\ \mathbf{A_3} & \mathbf{A_4}\end{bmatrix},\ \mathbf{B}=\begin{bmatrix}\mathbf{b_1}\\ \mathbf{b_2}\end{bmatrix}. \tag{4-9}$$

Then, the second introduced operation *“• “* represents matrix multiplication as

$$\mathbf{A}\bullet\mathbf{b_1}=\begin{bmatrix}\mathbf{A_1}\cdot\mathbf{b_1} & \mathbf{A_2}\cdot\mathbf{b_1}\\ \mathbf{A_3}\cdot\mathbf{b_1} & \mathbf{A_4}\cdot\mathbf{b_1}\end{bmatrix},\text{where for example } \mathbf{A_1}\cdot\mathbf{b_1}=\begin{bmatrix}[a_1b_1+a_2b_2]\\ [a_3b_1+a_4b_2]\end{bmatrix} \tag{4-10}$$

Following the example shows the other defined sign *“⊗”,* which represents matrix multiplication as

$$\mathbf{A}\otimes\mathbf{B}=\begin{bmatrix}\mathbf{A_1} & \mathbf{A_2}\\ \mathbf{A_3} & \mathbf{A_4}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{b_1}\\ \mathbf{b_2}\end{bmatrix}=\begin{bmatrix}[\mathbf{A_1}\cdot\mathbf{b_1}+\mathbf{A_2}\cdot\mathbf{b_2}]\\ [\mathbf{A_3}\cdot\mathbf{b_1}+\cdot\mathbf{A_4}\cdot\mathbf{b_2}]\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}a_1b_1+\ a_2b_2\ +a_5b_3\ +\ a_6b_4\\ a_3b_1+\ a_4b_2\ +\ a_7b_3\ +\ a_8b_4\end{bmatrix}\\ \begin{bmatrix}a_9b_1+a_{10}b_2\ +a_{13}b_3+a_{14}b_4\\ a_{11}b_1+a_{12}b_2+a_{15}b_3+a_{16}b_4\end{bmatrix}\end{bmatrix} \tag{4-11}$$

The additional operators for multidimensional matrix multiplication defined in [4-1] [4-2] and [4-3] allow us to express the synaptic operation of HONNU (3-D, 4-D, and even higher) by the multiplication of multidimensional matrix composed from (n+1)×(n+1), ie., 2-D, augmented neural-weight matrices and (n+1)×1 vector of n neural inputs. It should be emphasized that a stricter and more rigorous mathematical notation of dimensions might be required by mathematicians. However, this simplified multidimensional matrix multiplication technique can be applied to solve the problem of the matrix notation of synaptic operation of HONNU such as CNU or even of higher-nonlinearity-order units as shown below. The summation form of the synaptic operation of CNU, shown in Eq.(4-6), can be composed of either the summation or the matrix notation form of QNU, as seen in Eq.(4-3) and Eq.(4-5), and this is shown below in Eq.(4-12).

$$\begin{aligned} v_{Cubic} &= \sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} w_{ijk}x_ix_jx_k=\sum_{i=0}^{n}x_i\sum_{j=i}^{n}\sum_{k=j}^{n}w_{ijk}x_jx_k=\sum_{i=0}^{n}x_iv_i\ = \\ &= \sum_{i=0}^{n}x_i\cdot(\mathbf{xa}^{\mathbf{T}}\cdot\mathbf{W_{Qi}}\cdot\mathbf{xa})=\mathbf{x}^{\mathbf{T}}\cdot\begin{bmatrix}\mathbf{xa}^{\mathbf{T}}\cdot\mathbf{W_{Q_0}}\cdot\mathbf{xa}\\ \mathbf{xa}^{\mathbf{T}}\cdot\mathbf{W_{Q_1}}\cdot\mathbf{xa}\\ \vdots\\ \mathbf{xa}^{\mathbf{T}}\cdot\mathbf{W_{Q_n}}\cdot\mathbf{xa}\end{bmatrix}. \end{aligned} \tag{4-12}$$

where n is the number of elements of an augmented vector $\mathbf{xa}$, and $\mathbf{W_{Qi}}$ is an appropriate augmented (n+1) × (n+1) weight matrix with elements selected by rule:

$$\begin{aligned}\mathbf{W_{Q_i}}[j,k] &= w_{ijk} \quad \text{if} \quad k\geq j\geq i,\ \text{for}\ i=0,1,...,n\\ &= 0 \quad \text{else.}\end{aligned} \tag{4-13}$$

Applying the matrix operations introduced in definitions [4-1] [4-2] [4-3] on page 49, the matrix notation of CNU in Eq.(4-12) can be further simplified as

$$v_{Cubic} = \mathbf{xa}^{\mathbf{T}} \otimes \begin{bmatrix} \mathbf{xa}^{\mathbf{T}} \cdot \mathbf{W}_{\mathbf{Q}_{\mathbf{0}}} \cdot \mathbf{xa} \\ \mathbf{xa}^{\mathbf{T}} \cdot \mathbf{W}_{\mathbf{Q}_{\mathbf{1}}} \cdot \mathbf{xa} \\ \vdots \\ \mathbf{xa}^{\mathbf{T}} \cdot \mathbf{W}_{\mathbf{Q}_{\mathbf{n}}} \cdot \mathbf{xa} \end{bmatrix} \cdot = \mathbf{xa}^{\mathbf{T}} \otimes [\mathbf{xa}^{\mathbf{T}} \bullet \begin{bmatrix} \mathbf{W}_{\mathbf{Q}_{\mathbf{0}}} \\ \mathbf{W}_{\mathbf{Q}_{\mathbf{1}}} \\ \vdots \\ \mathbf{W}_{\mathbf{Q}_{\mathbf{n}}} \end{bmatrix} \bullet \mathbf{xa}] = \mathbf{xa}^{\mathbf{T}} \otimes [\mathbf{xa}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{C}} \bullet \mathbf{xa}] = \qquad (4\text{-}14)$$

$$= \mathbf{xa}^{\mathbf{T}} \cdot [\mathbf{xa}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{C}} \bullet \mathbf{xa}] \in R^1 .$$

Because the term $[\mathbf{xa}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{C}} \bullet \mathbf{xa}]$ represents a single column vector whose elements are scalars, the sign "$\otimes$" in the last row can be changed to "$\cdot$"; thus, the synaptic operation of CNU can be expressed using the new multidimensional matrix $\mathbf{W}_{\mathbf{C}}$, which appears as $(n+1)\times 1$ array (a column vector with matrices as its elements), as follows

$$v_{Cubic} = \sum_{i=0}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} w_{ijk} x_i x_j x_k = \mathbf{x}^{\mathbf{T}} \cdot [\mathbf{x}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{C}} \bullet \mathbf{x}] \in R^1, \text{ where } x_0 = 1 , \qquad (4\text{-}15)$$

where we refer to $\mathbf{W}_{\mathbf{C}}$ as the "*upper weight matrix* " (or "*upper column vector*). Consequently, the elements of *upper matrix* $\mathbf{W}_{\mathbf{C}}$ are *sub-matrices* $\mathbf{W}_{\mathbf{Q_i}}$ for $i = 0,1,...,n$. Sometimes in literature, the submatrices $\mathbf{W}_{\mathbf{Q_i}}$ as elements of the multidimensional array are called "*pages*" (e.g., in *Manual for Matlab, Humusoft*) .

The full CNU matrix notation has been derived in Eq.(4-15). Using this approach, the matrix notation for general higher-order polynomial (higher than CNU that is of order n=3*)* neural synaptic operations can be developed. For example, for the fourth-order polynomial in the synaptic operation of HONNU

$$v_{Fourth\,Order} = \sum_{i=0}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} \sum_{l=k}^{n} w_{ijkl} x_i x_j x_k x_l , \qquad (4\text{-}16)$$

steps analogous to the case of developing CNU can be followed; that is, the formula can be expressed as a summation of cubic terms

$$v_{Fourth\,Order} = \sum_{i=1}^{n} x_i \cdot v_{i\,Cubic} = \sum_{i=0}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} \sum_{l=k}^{n} w_{ijkl} x_i x_j x_k x_l = \mathbf{x}^{\mathbf{T}} \otimes \begin{bmatrix} \mathbf{x}^{\mathbf{T}} \cdot [\mathbf{x}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{C}_{\mathbf{0}}} \bullet \mathbf{x}] \\ \mathbf{x}^{\mathbf{T}} \cdot [\mathbf{x}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{C}_{\mathbf{1}}} \bullet \mathbf{x}] \\ \vdots \\ \mathbf{x}^{\mathbf{T}} \cdot [\mathbf{x}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{C}_{\mathbf{n}}} \bullet \mathbf{x}] \end{bmatrix} =$$

$$= \mathbf{x}^{\mathbf{T}} \otimes [\mathbf{x}^{\mathbf{T}} \bullet [\mathbf{x}^{\mathbf{T}} \bullet \begin{bmatrix} \mathbf{W}_{\mathbf{C}_{\mathbf{0}}} \\ \mathbf{W}_{\mathbf{C}_{\mathbf{1}}} \cdot \\ \vdots \\ \mathbf{W}_{\mathbf{C}_{\mathbf{n}}} \end{bmatrix} \bullet \mathbf{x}]] = \mathbf{x}^{\mathbf{T}} \cdot [\mathbf{x}^{\mathbf{T}} \bullet [\mathbf{x}^{\mathbf{T}} \bullet \mathbf{W}_{\mathbf{F}} \bullet \mathbf{x}]] \in R^1$$

(4-17)

The technique can be used again to upgrade the matrix notation to a higher-order polynomial synaptic function, and the appropriate form of weight matrices can be easily derived using previously shown results and is shown in Eq.(4-17).

The elements of the fourth-order neural unit weight matrix $\mathbf{W_F}$ in Eq.(4-17) are CNU sub-matrices $\mathbf{W_{C_i}}$ (3-D) that can be decomposed further into QNU weight sub-matrices (2-D), as in Eq.(4-14). According to Eq.(4-14) and Eq.(4-17), the matrixes $\mathbf{W_C}$ and $\mathbf{W_F}$ can be expressed as sparse 2-D matrices. The higher the sparseness, the higher the order of the polynomial of $f_{HONNU;}$ that is, the $\mathbf{W_C}$ is less sparse than $\mathbf{W_F}$, and so on.

To summarize section 4.1.2, auxiliary operators for multidimensional matrix multiplication were introduced in definitions [4-1], [4-2] and [4-3] and illustrated in examples using Eq.(4-8) to Eq.(4-11). The matrix notation was developed for the summation notation of the neural synaptic operation of CNU in Eq.(4-6) or for the general higher-polynomial-order neural synaptic operation shown in Eq.(4-7). Except for the development of the matrix notation of the synaptic operation of HONNU in general, the approach introduces the decomposition of multidimensional matrix[10] multiplication into multiplication of 2-D matrices using the defined operators "·" and "•" and "⊗" [11]. It also displays the relative simplicity of multidimensional matrix multiplication, which can be assumed to be too irrational at first sight. Establishing the matrix notation of synaptic operations of HONNU in this work can serve as a foundation for a new stability assessment of HONNU by matrix calculus; its particular development, however, exceeds the scope of this thesis.

Further in this thesis, static HONNU are further shown as useful in finding the initial weights for dynamic HONNU and, therefore, problems with the stability of the learning algorithm of dynamic HONNU can be simply avoided.

4.2 THE DYNAMIC NATURE OF A BIOLOGICAL NEURON AND DYNAMIC HONNU

In this section, the parallel between HONNU and the simplified understanding to the functionality of its biological counterpart is further proposed and even more emphasized due to the dynamic nature of the neural unit.

The notion regarding the importance not only of the synaptic but also of somatic neuronal part has been introduced in section 4.1.1 as deduced from the full notation of HONNU that maintains high computational capability of a unit as well as its simple mathematical structure. The computational role of the somatic part becomes even more evident for dynamic HONNU which structure is shown in Figure 25.

The neural state is represented by the variable ξ which represent the level of signal carried through axon forward to neural outputs.

[10] By a multidimensional matrix, we mean 3-D or more dimensional matrices, i.e., arrays of dimensions (m×n×k×...).

[11] "⊗" becomes "·" for HONNU, because only vectors with scalar elements are to be multiplied, as in the end of Eq.(4-12).

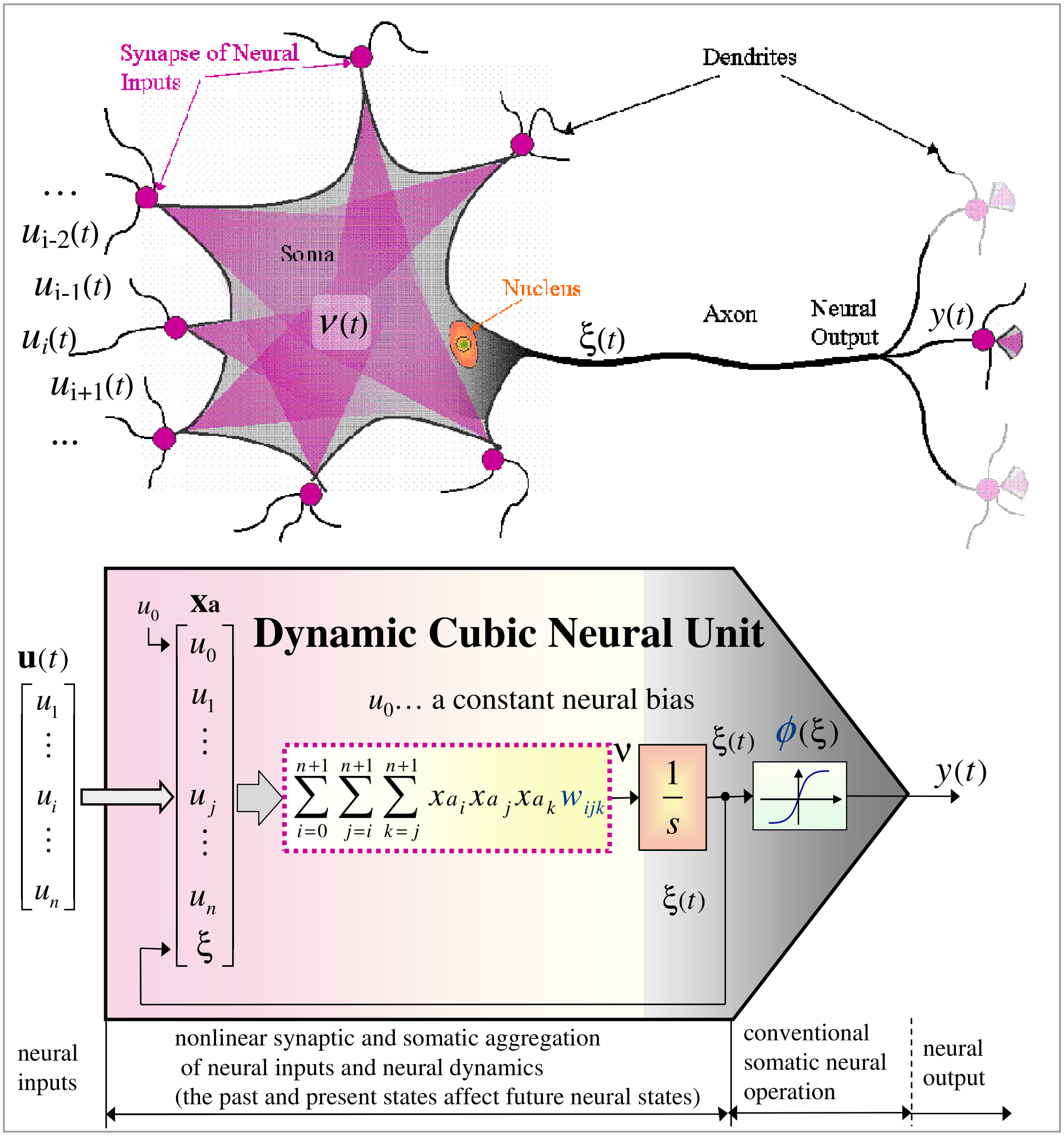

Synapse of Neural Inputs
Dendrites
…
$u_{i-2}(t)$
$u_{i-1}(t)$
$u_i(t)$
$u_{i+1}(t)$
…
Soma
$v(t)$
Nucleus
$\xi(t)$
Axon
Neural Output
$y(t)$
$\mathbf{u}(t)$
u_1 ⋮ u_i ⋮ u_n
u_0
Xa
u_0 u_1 ⋮ u_j ⋮ u_n ξ
Dynamic Cubic Neural Unit
u_0… a constant neural bias
$\sum_{i=0}^{n+1}\sum_{j=i}^{n+1}\sum_{k=j}^{n+1} xa_i xa_j xa_k w_{ijk}$
v
$\frac{1}{s}$
$\xi(t)$
$\phi(\xi)$
$y(t)$
$\xi(t)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics (the past and present states affect future neural states)
conventional somatic neural operation
neural output

Figure 25: General structure of the dynamic cubic neural unit (CNU) compared to its biological counterpart.

In the next section, the adaptable time-delays are introduced into dynamic neural units, inspired by successful applications of time delays into state-space models for system identification and control [45] [46] [56].

4.3 CONTINUOUS TIME-DELAY DYNAMIC NEURAL UNITS (TmD-DNU)

Before introducing nonlinear dynamic neural units with adaptable time delays (TmD-DHONNU), the concept of linear time-delay dynamic neural units (TmD-DNU) will be introduced. TmD-DNU represents a useful tool for the approximation of linear higher-order dynamic systems both with and without time delays. These units can also be used to identify precisely time delays within real linear plants.

The nature of the time-delay dynamic neural units (TmD-DNU) originates from linear dynamic neural units that can also be viewed as a particular adaptive mechanism capable of approximation of a dynamic system in the form of a linear differential equation. The analogy to a differential equation indicates that we are going to deal with continuous dynamic neural units (DNUs), which are working in continuous-time, where the fastest sampling period of a whole neural architecture is practically determined by the capabilities of a specific numerical method. The simplicity and user customizable structure of TmD-DNU are among the main advantages of these novel neural architectures. They also represent a direct analogy to linear differential equations with (optional) time delays. From the user's practical point of view, the appropriate structure of DNU can be defined according to prior information about the approximated dynamic system resulting from more detailed mathematical analyses, or according to less exact pragmatic speculations on system features. More various architectures of TmD-DNUs may be defined by an investigator and tested which would fit best the character of an investigated complex behaving dynamic system. TmD-DNU will be further classified into two major types: Type 1 (Figure 26, Figure 27) and Type 2 (Figure 29).

4.3.1 Time-Delay Dynamic Neural Units (TmD$_1$-DNU, TmD$_2$-DNU)

The TmD-DNU–Type 1 (TmD$_1$-DNU) might be viewed as the constituting element of conventional Time-Delay Neural Networks (TmDNN); however, we are focused on adaptable time delays within a single neural unit and not on the delays in inter-neuron connections that are common with tapped-delay neural networks (TDNN). This difference will become more apparent further in comparison to TmD$_2$-DNU. The architecture of TmD$_1$-DNU corresponds to the structure of the first-order differential equation with time-delayed input, shown in Eq.(4-19).

$$\begin{gathered}\frac{d\xi(t)}{dt}\cdot(\tau+\tau_{\min})+\xi(t)=\sum_{i=0}^{n} w_i\, u_i(t-T_i)\\ y(t)=\phi(\xi(t))\end{gathered} \tag{4-19}$$

The Laplace transfer function of the internal dynamic structure of TmD$_1$-DNU is derived in Eq.(4-20); this transfer function has been used to develop the learning algorithm for continuous time TmD$_1$-DNU according to Eq.(4-28).

$$\Xi(s)=\frac{\sum\limits_{i=0}^{n} w_i\, U_i\, e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}=\sum_{i=0}^{n}\frac{w_i\, e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}U_i=\sum_{i=0}^{n} G_i(s)\cdot U_i,\ \text{where}\ L\{\xi(t)\}=\Xi(s) \tag{4-20}$$

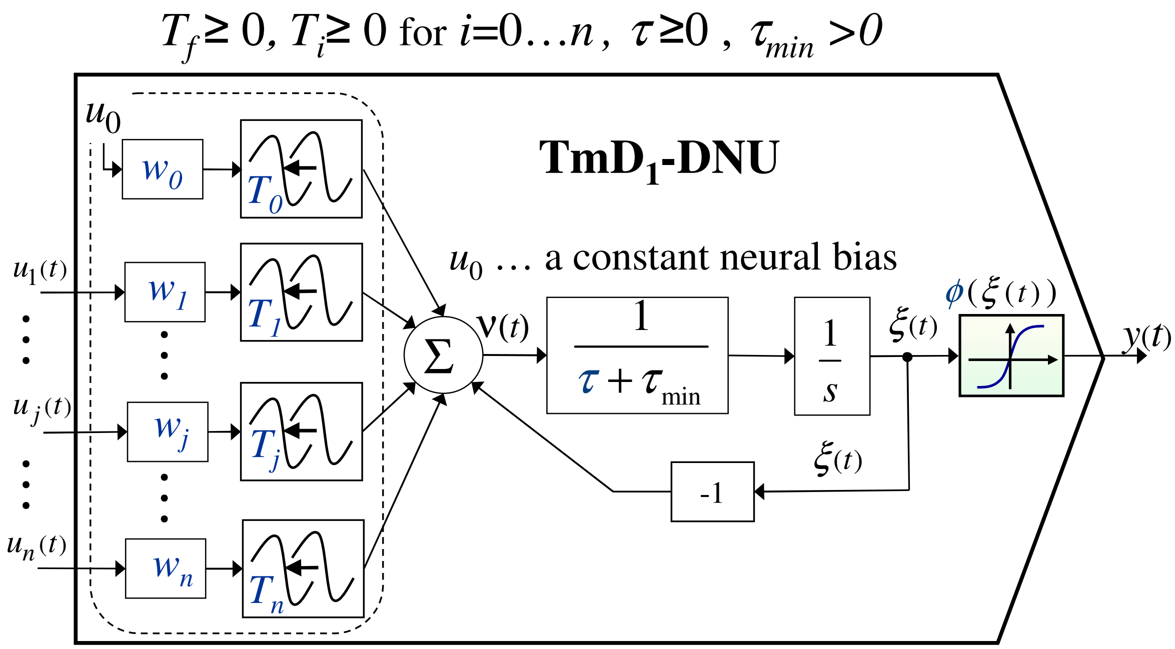


Figure 26: Linear Time-Delay Dynamic Neural Unit - Type 1 (TmD$_1$-DNU) with adaptable time delays T_i in i^{th} neural input where i=1…n.

Let us consider modification of a single input DNU given in Figure 26, where τ_1, τ_2 and w_i (i=$1…n$) are neural weights, n is the number of neural inputs, T_j represents directly the time delay of j^{th} neural input, constant τ_{min} is a minimum (positive) time constant of the unit, $\boldsymbol{u}(t)$ is vector of neural inputs, $\xi(t)$ is an internal state variable, $\phi()$ is a somatic (transfer) function, $y(t)$ is the neural output, and s is the Laplace operator. To assure the maximum stability of TmD-DNU, a simple quadratic substitution has been used. For simplicity and identification purposes, the neural somatic operation $\phi()$ of a neuron will be kept as linear. The neural architecture in Figure 26 (i^{th} neural unit) represents a dynamic system analogical to a plant with time delays T_j on its input. It is shown as a single-input and unbiased unit in the block diagram in Figure 27 using a simple resulting transfer function $G_{TmD_1\text{-}DNU}$ (s) with Laplace operator s and neural somatic operation $\phi()$.

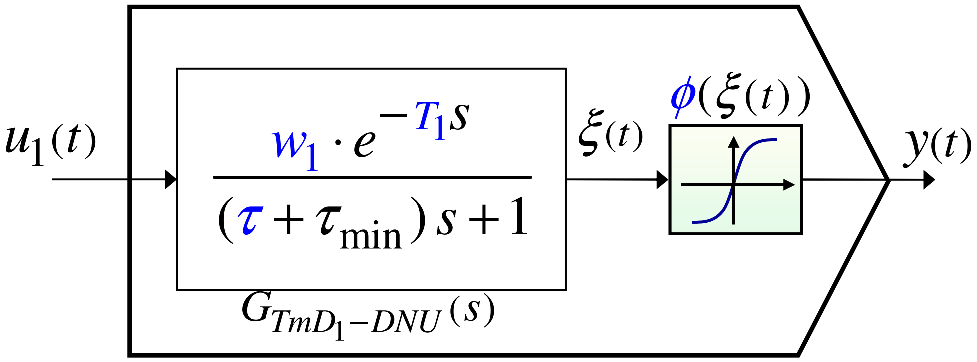


Figure 27: The analogy of a single input TmD$_1$-DNU (without a bias) to a linear dynamic plant with time-delayed input (Figure 26).

It should be emphasized that this proposed structure of TmD$_1$-DNU, as in Figure 26 or Figure 27, with an appropriately (small and positive) chosen time constant τ_{min} and with introduced positive adaptable neural parameters τ and T_1, assures the dynamic nature of the unit and its stability. It is also a preventive measure that improves the stability of the whole learning algorithm and possibly of the whole TmD-DNU networks.

In further reading, the time-delay dynamic structure is further extended in order to improve the single-neural-unit approximating capabilities of TmD-DNU for even higher-order dynamic systems.

$T_f \geq 0, T_i \geq 0$ for $i=0\ldots n$, $\tau \geq 0$, $\tau_{min} > 0$

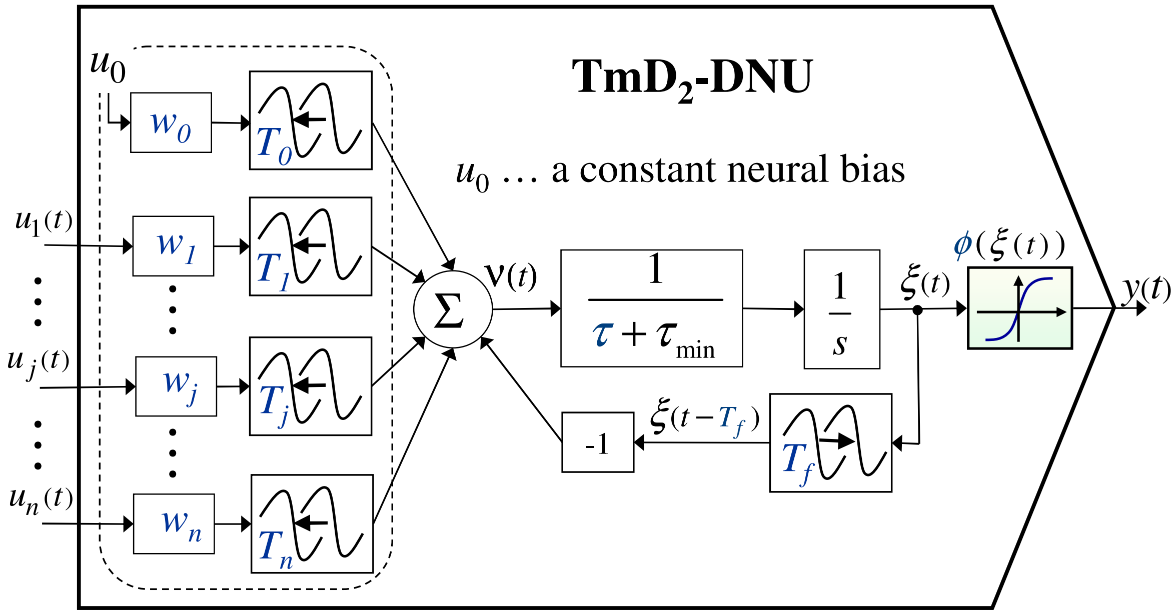


Figure 28: Linear Time-Delay Dynamic Neural Unit – Type 2 (TmD_2-DNU) with adaptable time delays T_i on its input and with adaptable delay T_f in state feedback of a unit.

Researchers dealing with control engineering applications may sometimes deal with linear dynamic systems containing not only the input delays but also the time-delayed state variables. The reason for that can be seen in the introduction of time delay, here denoted as T_f , into the state feedback of a first-order system with input time delay as

$$\frac{d\xi(t)}{dt}\cdot(\tau+\tau_{\min})+\xi(t-T_f)=\sum_{i=0}^{n} w_i\, u_i(t-T_i)$$
$$y(t)=\phi(\xi(t)) \qquad . \tag{4-21}$$

The Laplace transfer function of the internal dynamic structure of TmD_2-DNU is derived n Eq.(4-22); this transfer function has been used to develop the learning algorithm for continuous time TmD_2-DNU in Eq.(4-28) and in Eq.(4-22) .

$$\Xi(s)=\sum_{i=0}^{n}\frac{w_i\, e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+e^{-s\,T_f}}U_i=\sum_{i=0}^{n}G_i(s)\cdot U_i, \quad \text{where} \quad L\{\xi(t)\}=\Xi(s) \tag{4-22}$$

The introduction of adaptable time delay T_f results in the increase of approximating capability for higher-order dynamic systems as shown in [45] and [46], thus it potentially results in more robust dynamic neural networks with an even fewer number of neural parameters and a less complicated structure for a given problem. We propose the above dynamic structure in Eq.(4-21) to be the basis of the second type of time-delay dynamic neural units denoted as TmD$_2$-DNU.

As an example, the learning algorithm enhanced with the adaptable continuous-time-delay parameter T_f of the state variable ξ is shown later in Eq.(4-35) and Eq.(4-36) and in Figure 34.

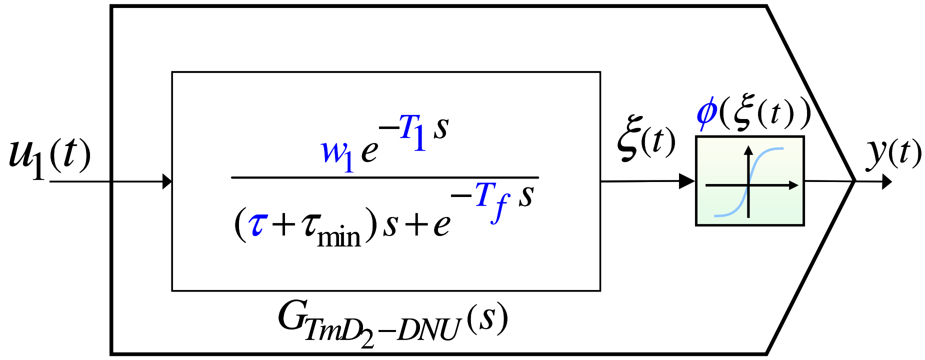
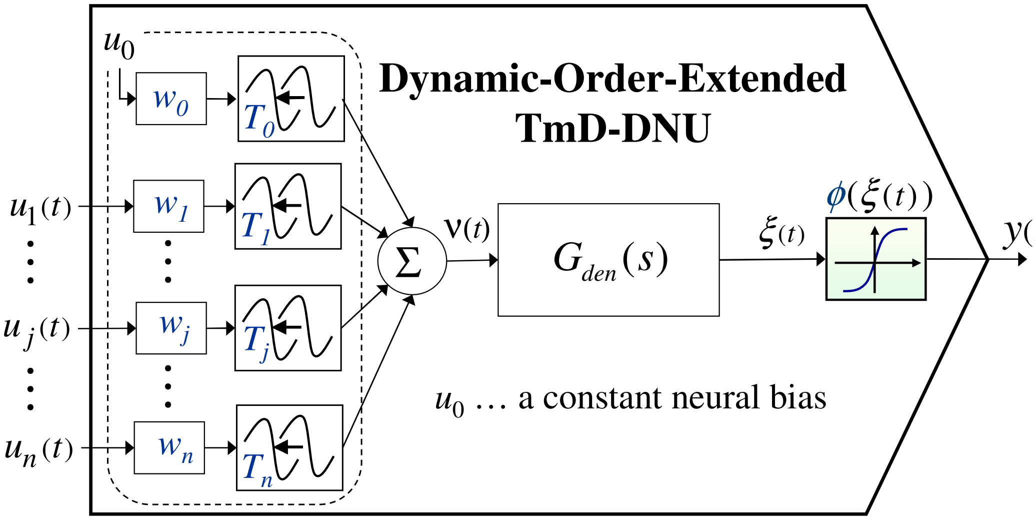


Figure 29: The analogy of a single input time-delay dynamic neural unit – type 2 (TmD$_2$-DNU) to a linear dynamic plant (Eq.(4-21)) with both time-delayed inputs and state variable $\xi(t)$.

4.3.2 Dynamic-Order-Extended Time-Delay Dynamic Neural Units



$$G_{den}(s)=\frac{1}{\left((\tau_1+\tau_{\min})s+e^{-(T_f\,s)}\right)((\tau_2+\tau_{\min})s+1)}$$

where $T_0,\ldots,T_j,\ldots T_n \geq 0,\ T_f \geq 0,\ T_j \geq 0\ for\ j=0\ldots n,\ \tau_1 \geq 0, \tau_2 \geq 0,\ \tau_{\min} > 0,$

Figure 30: Dynamic-Order-Extended TmD-DNU with higher approximating capabilities resulting from higher density of poles and zeros in complex plain due to the adaptable input delays, second-order internal dynamics, and the adaptable time delay T_f (or more than one delay) in state feedback of a unit.

$$\Xi(s)=\frac{\sum_{i=0}^{n} w_i\, U_i\, e^{-s\,T_i}}{\left((\tau_1+\tau_{\min})\,s+e^{-(T_f\,s)}\right)\left((\tau_2+\tau_{\min})\,s+1\right)}=\sum_{i=0}^{n} G_i(s)\cdot U_i,\ \text{ where }\ L\{\xi(t)\}=\Xi(s) \qquad (4\text{-}23)$$

4.3.3 Time-Delay Dynamic Higher-Order Nonlinear Neural Units

Time-delay dynamic higher-order nonlinear neural unit (TmD-DHONNU) are one of the most computationally powerful stand-alone neural units. They maintain the minimal number of neural parameters due to a combination of adaptable time-delays in neural inputs and in state feedback(s) of the unit, enhanced with the nonlinear aggregation of the inputs and state variable(s). This thesis is focused on the approximation of a complex system by as simple a model as possible, so the stand-alone TmD-DHONNU architecture represents one of the most powerful units for system approximation within the frame of the proposed research work. The general structure of time-delay dynamic cubic neural units as representative of TmD-DHONNU is given as

$$\frac{d^m\xi(t)}{dt^m}=f_{TmD-CNU}\left(\mathbf{xa}\right)=\sum_{i=0}^{m+n}\sum_{j=i}^{m+n}\sum_{k=j}^{m+n} x_i(t-T_i)\,x_j(t-T_j)\,x_k(t-T_k)\,w_{ijk}=$$

$$y(t)=\phi(\xi(t))$$

$$\text{where }\ \mathbf{xa}=\left[x_0\ \ \ldots\ \ x_i\ \ \ldots\ \ x_{m+n+1}\right]^T=$$

$$=\left[1\quad u_1(t-T_1)\quad\ldots\quad u_j(t-T_j)\quad\ldots\quad u_n(t-T_n)\quad \xi(t-T_{n+1})\quad \frac{d\xi(t-T_{n+2})}{dt}\quad\ldots\quad \frac{d^{m-1}\xi(t-T_{m+n})}{dt^{m-1}}\right]^T, \qquad (4\text{-}24)$$

where $u_i(t)$ are neural inputs, $u_0=x_0=1$ is neural bias, $y(t)$ is neural output, $\xi(t)$ and its m-1 derivatives represent m internal neural state variables, w_{ijk} are neural weights, $T_0\ \ldots T_n$ are adaptable delays of n neural inputs, and $T_{\mathrm{n+1}}\ \ldots T_m$ are adaptable delays in m state feedbacks. Blue color again highlights the adaptable neural parameters.

For further exploration of on TmD-DHONNU, see subsection 4.4.4, p.68.

4.4 DEVELOPMENT OF THE LEARNING ALGORITHM OF HONNU, TMD-DNU AND TMD-DHONNU

In this section, the supervised learning algorithm is shown for each type of the neural architectures proposed in this thesis.

4.4.1 Learning Algorithm for Static HONNU

The backpropagation (BP) gradient learning algorithm for static HONNU shown in Figure 31 can be applied for its simplicity and applicability to various classes of artificial neural systems.

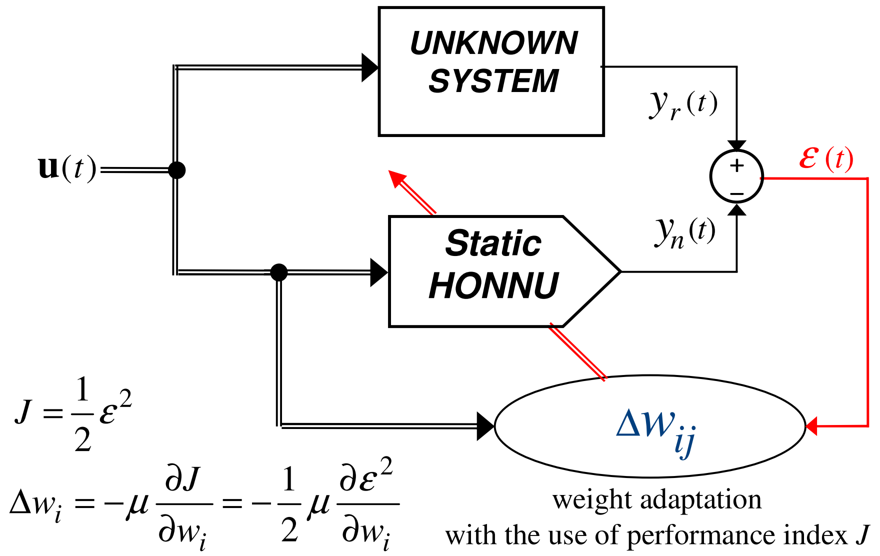

UNKNOWN SYSTEM
$y_r(t)$
$\mathbf{u}(t)$
+
−
$\varepsilon(t)$
$y_n(t)$
Static HONNU
$J = \frac{1}{2}\varepsilon^2$
Δw_{ij}
$\Delta w_i = -\mu \frac{\partial J}{\partial w_i} = -\frac{1}{2}\mu \frac{\partial \varepsilon^2}{\partial w_i}$
weight adaptation
with the use of performance index J

Figure 31: Supervised learning algorithm for static HONNU that can have either the scalar or the vector input; neural input **u** enters into the calculation of neural weight increments Δw_{ij}.

In the simplest form, the neural weights are adapted against the gradient of the performance index shown in Eq.(4-25)

$$\Delta w_i = -\mu \frac{\partial J}{\partial w_i}, \quad J = \frac{1}{2}\varepsilon^2, \qquad (4\text{-}25)$$

where Δw_i is a neural-weight increment, J is the performance index, μ is the learning rate, and ε is error between the neural and the desired signal output. The form of the backpropagation learning algorithm can be applied to static HONNU, shown in Figure 21, and the neural-weight increments are calculated by the set of Eq.(4-26).

$$\begin{aligned}
&w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k)\\
&\Delta w_{ij}(k+1) = -\frac{1}{2}\mu \frac{\partial(\varepsilon)^2}{\partial w_{ij}} = -\mu\,\varepsilon(t)\frac{\partial(y_r - y)}{\partial w_{ij}} = \mu\,\varepsilon(t)\frac{\partial \phi(\nu)}{\partial w_{ij}},\\
&\Delta w_{ij}(k+1) = \mu\,\varepsilon \frac{\partial \phi}{\partial \nu}\frac{\partial \nu}{\partial w_{ij}} = \mu\,\varepsilon \frac{\partial \phi}{\partial \nu}\frac{\partial\, f_{HONNU}(\mathbf{xa},\mathbf{Wa})}{\partial w_{ij}},
\end{aligned} \qquad (4\text{-}26)$$

In particular, f_{HONNU}, **xa**, and **Wa** are evaluated for quadratic neural units in section 4.1.2.

$$\begin{aligned}
\Delta w_{ij}(k+1) &= -\frac{1}{2}\mu \frac{\partial \varepsilon^2}{\partial w_i} = -\mu\,\varepsilon \frac{\partial(y_r(k) - y_n(k))}{\partial w_i} = \mu\,\varepsilon \frac{\partial y_n(k)}{\partial w_i} =\\
&= \mu\,\varepsilon \frac{\partial \phi(\xi(k-1))}{\partial w_i} = \mu\,\varepsilon \frac{\partial \phi}{\partial \xi}\frac{\partial f_{HONNU}(\mathbf{xa},\mathbf{Wa})}{\partial w_{ij}} =\\
&= \mu\,\varepsilon \frac{\partial \phi}{\partial \xi}\frac{\partial}{\partial w_{ij}}\left[\sum_{i=0}^{n+1}\sum_{j=i}^{n+1} x_i x_j w_{ij}\right] = \mu\,\varepsilon \frac{\partial \phi}{\partial \xi} x_i x_j
\end{aligned} \qquad (4\text{-}27)$$

where $\mathbf{xa} = \begin{bmatrix} x_0 & \dots & x_i & \dots & x_{n+1}\end{bmatrix}^T = \begin{bmatrix} 1 & u_1(k-1) & \dots & u_j(k-1) & \dots & u_n(k-1)\end{bmatrix}^T$

4.4.2 Learning Algorithm for Linear Time-Delay Dynamic Neural Units (TmD-DNU)

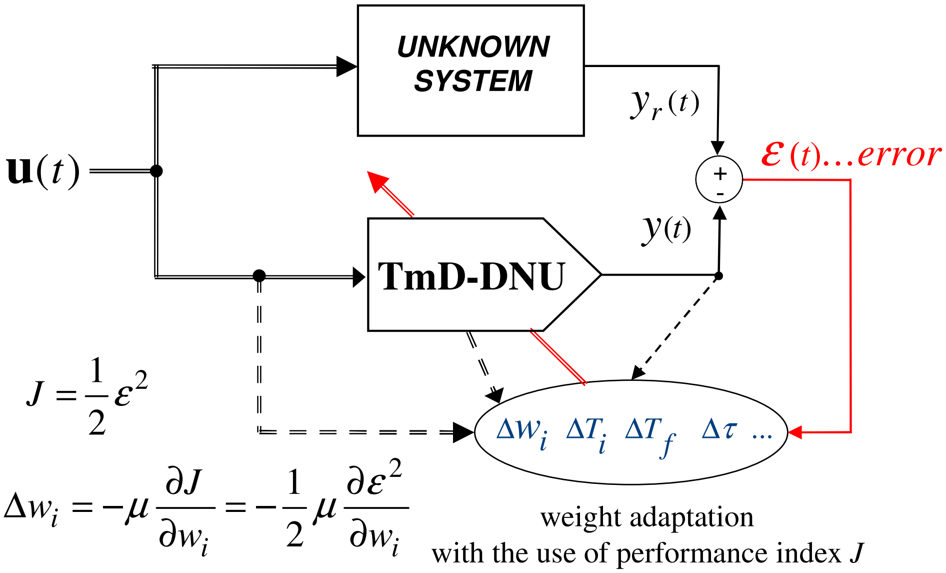

UNKNOWN SYSTEM
$\mathbf{u}(t)$
$y_r(t)$
$\varepsilon(t)\ldots error$
+
-
$y(t)$
TmD-DNU
$J=\frac{1}{2}\varepsilon^2$
$\Delta w_i \;\; \Delta T_i \;\; \Delta T_f \;\; \Delta\tau \;\ldots$
$\Delta w_i=-\mu\frac{\partial J}{\partial w_i}=-\frac{1}{2}\mu\frac{\partial\varepsilon^2}{\partial w_i}$
weight adaptation
with the use of performance index J

Figure 32: The principle of the supervised learning algorithm for all TmD-DNU, dynamic HONNU, and TmD-DHONNU; neural input u, internal neural state variables, and neural outputs enter into the calculation of neural-parameter increments Δw_i , ΔT_i , ΔT_f , $\Delta\tau$,...

Because the internal dynamic structure of TmD-DNU (Figure 26 to Figure 28) represents a linear time-delay dynamic system, the neural-weight increment can be evaluated by the rules of Eq.(4-25); however, using the Laplace transfer function of the internal neural dynamics as shown in set of Eq.(4-28),

$$w_i(k+1)=w_i(k)+\Delta w_i(k)$$

$$\Delta w_i(k+1)=-\frac{1}{2}\mu\frac{\partial(\varepsilon(t))^2}{\partial w_i}=-\mu\, e(t)\frac{\partial(y_r(t)-y(t))}{\partial w_i}=\mu\,\varepsilon(t)\frac{\partial\phi(\xi(t))}{\partial w_i} \tag{4-28}$$

$$\Delta w_i(k+1)=\mu\,\varepsilon(t)\left[\frac{\partial\phi(t)}{\partial\xi(t)}\frac{\partial\xi(t)}{\partial w_i}\right]=\mu\,\varepsilon(t)\left[\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{\partial}{\partial w_i}\left(\sum_{i=0}^{n}G_i(s)\,U_i(s)\right)\right\}\right],$$

where G_i (s) is the i^{th} component of the Laplace transfer function of the internal linear dynamic structure of a linear TmD-DNU corresponding to i^{th} neural input u_i , as derived in Eq.(4-20) and Eq.(4-22); t is a continuous parameter of time, L^{-1} is the inverse Laplace transform, $\xi(t)$ is the internal neural state variable, and w_i represents the neural weight, n is the number of neural inputs, and u_0 corresponds to an internal neural bias. It can be substituted by another adaptable neural parameter such as the adaptable time delay (T_i or T_f) or other adaptable dynamic parameters (such as τ) of TmD-DNU (eg., Figure 28),

By substituting Eq.(4-20) into Eq.(4-28), the learning algorithm for neural weights of TmD_1-DNU can be derived in Eq.(4-29).

$$\begin{aligned}\Delta w_i(k+1) &= \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}\frac{\partial\xi(t)}{\partial w_i} = \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}\frac{\partial}{\partial w_i}\left[L^{-1}\left\{(\sum_{i=0}^{n} G_i(s)\,U_i(s))\right\}\right] = \\ &= \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{\partial}{\partial w_i}\left[\frac{\sum\limits_{i=0}^{n} w_i\,U_i\;e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}\right]\right\} = \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{U_i\;e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}\right\}\end{aligned}\tag{4-29}$$

Another neural parameters T_i (where i=0...n) representing input time delays can similarly be adapted as shown below

$$\begin{aligned}\Delta T_i(k+1) &= \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{\partial}{\partial T_i}\left[\frac{\sum\limits_{i=0}^{n} w_i\,U_i\;e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}\right]\right\} = \\ &= -\mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{s\,w_i\,U_i\;e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}\right\},\end{aligned}\tag{4-30}$$

where τ representing the internal dynamics is adapted as

$$\begin{aligned}\Delta\tau(k+1) &= \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{\partial}{\partial\tau}\left[\frac{\sum\limits_{i=0}^{n} w_i\,U_i\;e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}\right]\right\} = \\ &= \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{-\frac{s\sum\limits_{i=0}^{n} w_i\,U_i\;e^{-s\,T_i}}{\left((\tau+\tau_{\min})\cdot s+1\right)^2}\right\} = \\ &= \mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{-s}{(\tau+\tau_{\min})\cdot s+1}\cdot\frac{\sum\limits_{i=0}^{n} w_i\,U_i\;e^{-s\,T_i}}{(\tau+\tau_{\min})\cdot s+1}\right\} = \\ &= -\mu\,\varepsilon(t)\frac{\partial\phi(t)}{\partial\xi(t)}L^{-1}\left\{\frac{s}{(\tau+\tau_{\min})\cdot s+1}\cdot L\{\xi(t)\}\right\}\end{aligned}\tag{4-31}$$

The examples of evaluation of particular neural-parameter increments for a single-input linear TmD-DNU without bias are shown further in this section.

In the text below, the learning rules for adaptable time-delay parameters are shown in detail for each of the two proposed neural architectures TmD$_1$-DNU and TmD$_2$-DNU.

Considering the dynamic structure of TmD$_1$-DNU as shown in Figure 26 and as further simplified in Figure 27, the neural output of a single-input TmD-DNU can be expressed using continuous-time transfer function as

$$y_n(t)=\phi(\xi(t))=\phi(L^{-1}\left\{G_1(s)\,U(s)\right\})=\phi(L^{-1}\left\{\frac{w_1\,e^{-T_1\,s}}{(\tau+\tau_{\min})\,s+1}U(s)\right\}), \qquad (4\text{-}32)$$

where e is the Euler's number and usually $\phi(\xi)=\xi$ for a case of stand-alone implementation of a unit. The increments of neural parameters are evaluated according to Eq.(4-28). In particular, the actual increment of the neural time-delay parameter T_1 can then be derived as

$$\begin{aligned}\Delta T_1(k+1)&=\mu\;\varepsilon(t)\frac{\partial\phi\left(L^{-1}\left\{G_1(s)\,U(s)\right\}\right)}{\partial T_1}=\\&=-\mu\;\varepsilon(t)\,L^{-1}\left\{\frac{w_1\;s\;e^{-T_1\,s}}{(\tau+\tau_{\min})\,s+1}U(s)\right\}.\end{aligned} \qquad (4\text{-}33)$$

The implementation of neural weight increment ΔT_1 in Eq.(4-33), which represents the adaptable time-delay parameter on neural input, is further depicted in Figure 33.

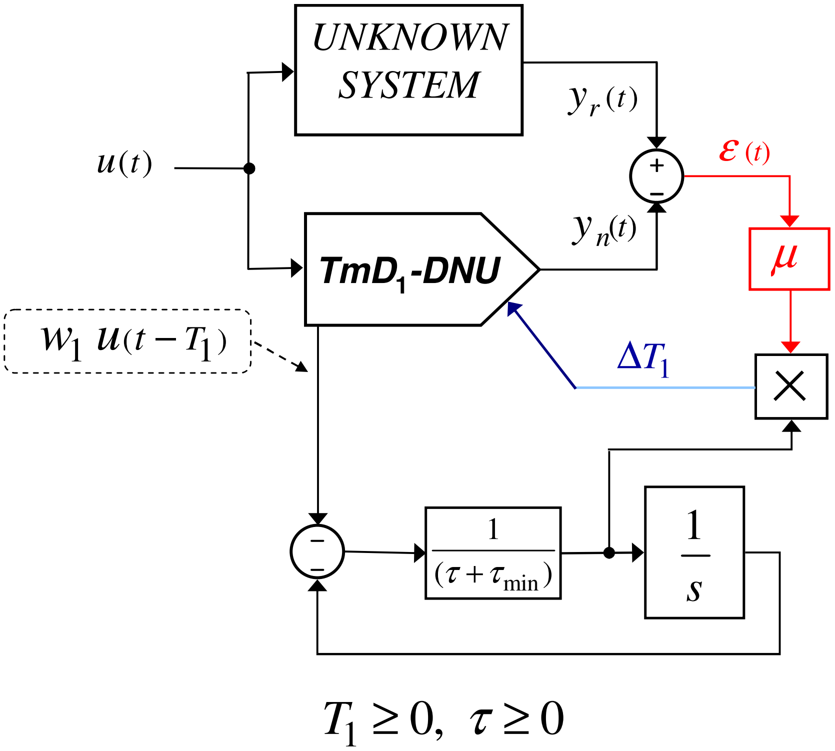


Figure 33: The mechanism for generating the neural weight increment ΔT_1 of neural weight T_1 that represents the adaptable input time delay of TmD$_1$-DNU (Figure 26, Figure 27).

The increments of the other neural parameter τ and of neural weights w_j of TmD$_1$-DNU shown in Eq.(4-19) and in Figure 26 and Figure 27 are generated similarly, that is by following principles in Eq.(4-29) and Eq.(4-31).

The learning algorithm for the defined structure of TmD$_2$-DNU and adaptable time-delay neural parameters τ and T_f is shown, as it displays good error convergence with excellent stability of a unit for a reasonably chosen learning rate μ and initial conditions. The maximum values of μ maintaining stable adaptation were up to μ=0.01, and the size of μ also depended on type of input signal[12]. To derive the learning algorithm for TmD$_2$-DNU, the same rules in Eq.(4-28) may be followed while the neural output shown in Eq.(4-32) yields

$$Y(s) = \phi(G_2(s)\,U(s)) = \frac{w_1\, e^{-T_1 s}}{(\tau + \tau_{\min})\, s + e^{-T_f s}}\, U(s)\,, \qquad (4\text{-}34)$$

where e is the Euler's number and $\phi(\xi)=\xi$ for simplicity.

Considering the simplification in Eq.(4-34), the increment of weight ΔT_f, representing the continuous-time adaptable delay of the state variable of a neural unit, can be evaluated as follows

$$\begin{aligned}\Delta T_f &= \mu\,\varepsilon(t)\,\frac{\partial y(t)}{\partial T_f} = \mu\,\varepsilon(t)\,\frac{\partial \phi\big(\xi(t)\big)}{\partial T_f}\bigg|_{\phi(\xi)=\xi} = \\ &= \mu\,\varepsilon(t)\, L^{-1}\left\{\frac{\partial\big\{G_2(s)\;U(s)\big\}}{\partial T_f}\, U(s)\right\} = \\ &= \mu\,\varepsilon(t)\, L^{-1}\left\{\frac{w_1\, e^{-T_1 s}\, e^{-T_f s}\, s}{\Big((\tau+\tau_{\min})\, s + e^{-T_f s}\Big)^2}\; U(s)\right\}.\end{aligned} \qquad (4\text{-}35)$$

The resulting transfer function in Eq.(4-35) can be further decomposed into two time-delayed plants in series. The implementation of the weight increment ΔT_f may then be realized with the use of the already-generated neural unit output $y(t)$ as

$$\begin{aligned}\Delta T_f &= \mu\,\varepsilon(t)\, L^{-1}\left\{\frac{e^{-T_f s}\, s}{(\tau+\tau_{\min})\cdot s + e^{-T_f s}}\; G_{TmD_2-DNU}(s)\, U(s)\right\} = \\ &= \mu\,\varepsilon(t)\, L^{-1}\left\{\frac{e^{-T_f s}\; s}{(\tau+\tau_{\min})\, s + e^{-T_f s}}\; Y(s)\right\}.\end{aligned} \qquad (4\text{-}36)$$

The implementation of Eq.(4-36) is then depicted in Figure 34 below.

[12] In general, the smoother input signal, the higher μ could have been used and the adaptation of a unit was stable and converging.

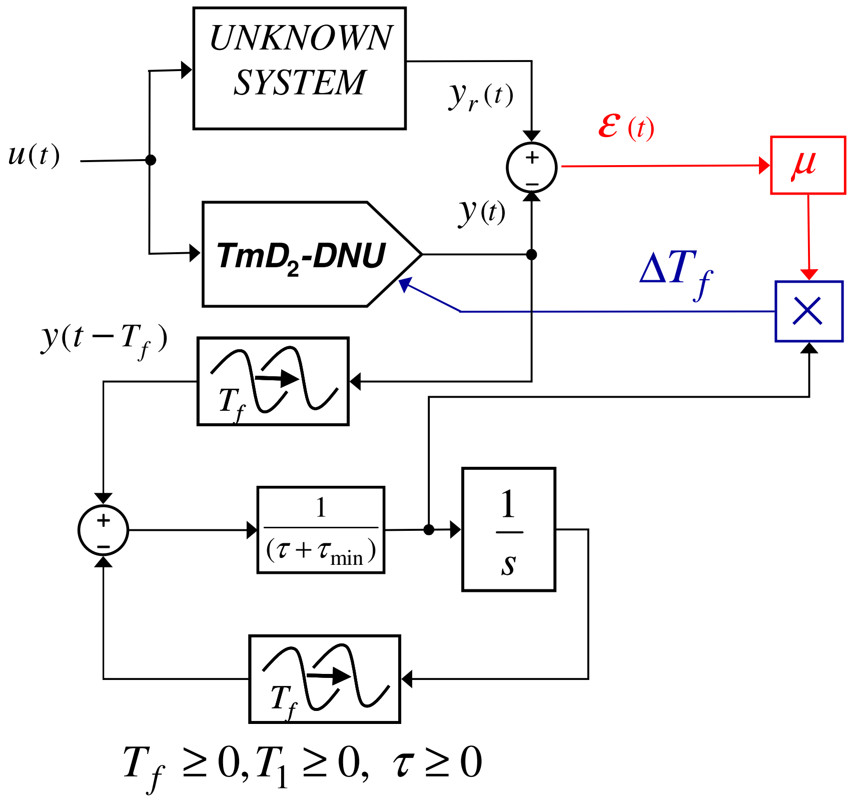


Figure 34: The mechanism for generating the neural weight increment ΔT_f of neural parameter T_f in Eq.(4-34), Eq.(4-35), and Eq.(4-36), which represents the adaptable time delay in state feedback of TmD_2-DNU (Figure 28).

Similarly, the remaining neural increments of TmD_2-DNU, such as w_i , τ, and T_i , can be evaluated by appropriately modified rules in Eq.(4-29) to Eq.(4-31), where Eq.(4-32) yields Eq.(4-34)

Besides the purposely designed stable internal dynamic structure of both TmD_1-DNU and TmD_2-DNU, no special measures were taken to assure the stability of the learning algorithm except for the appropriate choice of learning rate and the initial conditions. The choice of the initial neural parameters for TmD_2-DNU, Eq.(4-21), should fulfill the condition

$$\frac{T_f}{\tau+\tau_{\min}} < \frac{\pi}{2}, \quad (4\text{-}37)$$

which is an important stability condition for such a class of systems [45]. Based on the simulation experiments, the following simple guidelines were considered in setting the initial weights and the individual learning rates for each neural weight to ensure the adaptation stability of TmD_2-DNU:

1. Set the initial weights and neural parameters τ, T_1 , and T_f to be of random but, e.g., equal value $\tau(0)=T_1(0)=T_f(0)$ (however fulfilling Eq.(4-37)).
2. Set the appropriate initial learning rate (around $\mu = 0.001$).
3. Run the adaptation of the unit and observe which neural weights adapt too fast (i.e., which weights run toward much larger values than expected or oscillates in larger intervals than expected).

4. Decelerate learning of these significantly fast adapting neural weights by assigning them a lower value of the individual learning rate. In our experiments, the neural weight representing the static gain w_1 was the one that had to be decelerated.
5. Because the single TmD$_2$-DNU have been displaying robust approximating capability (for higher-order linear dynamic systems), more simulations from distinct sets of initial conditions are recommended to achieve a better degree of accuracy (i.e., a convergence toward the local minimum, providing the unit with an even higher degree of accuracy).

TmD-DNUs are focused upon as stand-alone neural units in this work, and a simple and pure learning rule (the dynamic backpropagation) is shown. Other, more advanced methods, such as learning with weight decay or other modifications of the learning rule, can be utilized as they are introduced [30].

4.4.3 Learning Algorithm for Discrete Dynamic HONNU

For the purpose of implementing discrete dynamic HONNU (Figure 25, Figure 35), the backpropagation gradient learning algorithm was extended to its dynamic form. It will be introduced in chapter 5 that static and dynamic BP algorithms were combined to prevent instability problems of learning algorithms for the general class of nonlinear (unbounded) functions as synaptic and somatic aggregation operations of HONNU. This was the simplest, most practical preventive measure to use to run a stable adaptation of dynamic HONNU.

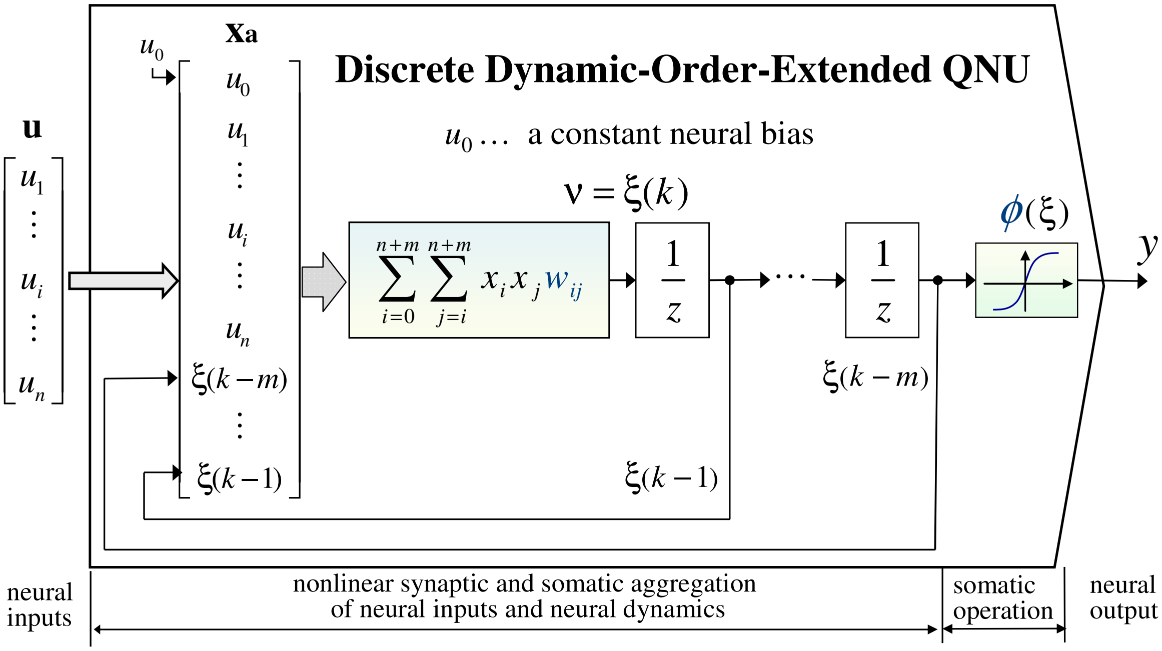

Discrete Dynamic-Order-Extended QNU
$u_0 \dots$ a constant neural bias
$\nu = \xi(k)$
u
u_1
u_i
u_n
Xa
u_0
u_0
u_1
u_i
u_n
$\xi(k-m)$
$\xi(k-1)$
$\sum_{i=0}^{n+m}\sum_{j=i}^{n+m} x_i x_j w_{ij}$
$\frac{1}{z}$
$\frac{1}{z}$
$\xi(k-m)$
$\xi(k-1)$
$\phi(\xi)$
y
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics
somatic operation
neural output

Figure 35: General structure of the discrete dynamic quadratic neural unit with the internal dynamics of order *m*.

A simple example of the learning rule for discrete dynamic QNU is given in its general form as follows

$$\Delta w_{ij}(k+1) = -\frac{1}{2}\mu\frac{\partial\varepsilon^2}{\partial w_i} = -\mu\varepsilon\frac{\partial(y_r(k)-y(k))}{\partial w_i} = \mu\varepsilon\frac{\partial y(k)}{\partial w_i} = \mu\varepsilon\frac{\partial\phi(\xi(k-n))}{\partial w_i} =$$

$$= \mu\varepsilon\frac{\partial\phi}{\partial\xi}\frac{\partial}{\partial w_{ij}}\cdot\left[D^m\{f_{HONNU}(\mathbf{xa},\mathbf{Wa})\}\right] =$$

$$= \mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[D^m\left\{\frac{\partial}{\partial w_{ij}}f_{HONNU}(\mathbf{xa},\mathbf{Wa})\right\}\right] = \qquad (4\text{-}38)$$

$$= (\text{for the case of QNU}) = \mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[D^m\{x_i x_j\}\right],$$

where $\mathbf{xa} = [x_0 \quad \ldots \quad x_i \quad \ldots \quad x_{m+n+1}]^T =$

$$= [1 \quad u_1(k-1) \quad \ldots \quad u_j(k-1) \quad \ldots \quad u_n(k-1) \quad \xi(k-m) \quad \ldots \quad \xi(k-1)]^T,$$

and D represents the delay operator as

$$D\{x(k)\} = x(k-1), \text{ or more generally } D\{x(i)\cdot x(j)\} = x(i-1)\cdot x(j-1). \qquad (4\text{-}39)$$

The discrete form of a system, obtained, for example, by simple differentiation, can be

$$y(k) + f(y(k-1), y(k-2), \ldots, y(k-m), u(k-m), \mathbf{Wa},) = 0, \qquad (4\text{-}40)$$

corresponding to a general discrete-time dynamic structure for which HONNU can be designed. The implementation of the learning algorithm of the discrete HONNU approximating system in Eq.(4-40) for m=2 is shown in Figure 36, where y_m= y_r is the output from a real plant.

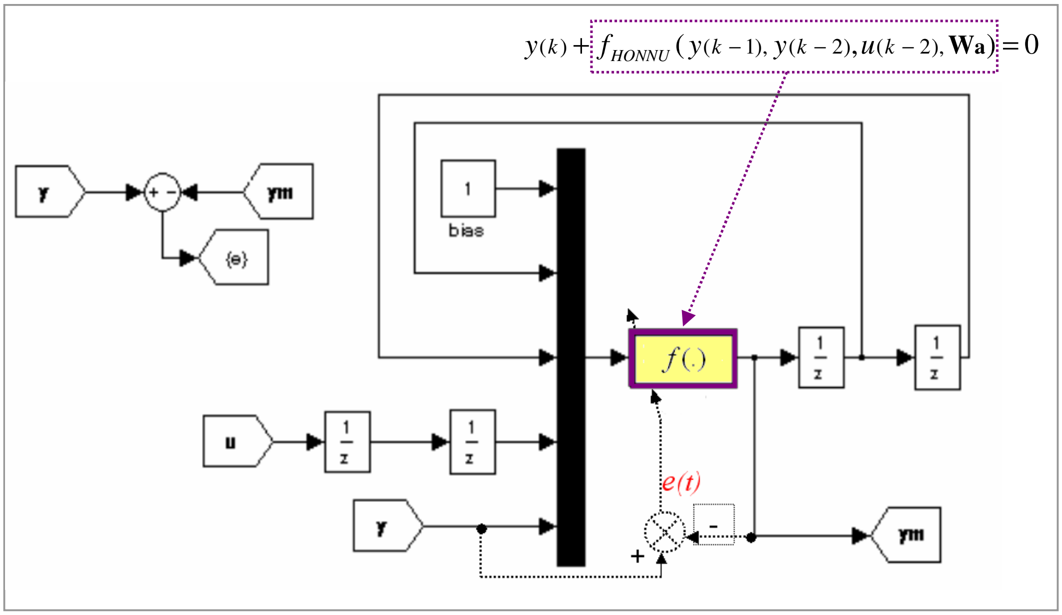


Figure 36: Sketch of the implementation of learning algorithm of stand-alone discrete single input dynamic HONNU with linear neural output operation $\phi(\,)$, m=2, unbiased by w_{00} =0.

4.4.4 Learning Algorithm for Continuous Dynamic HONNU and TmD-DHONNU

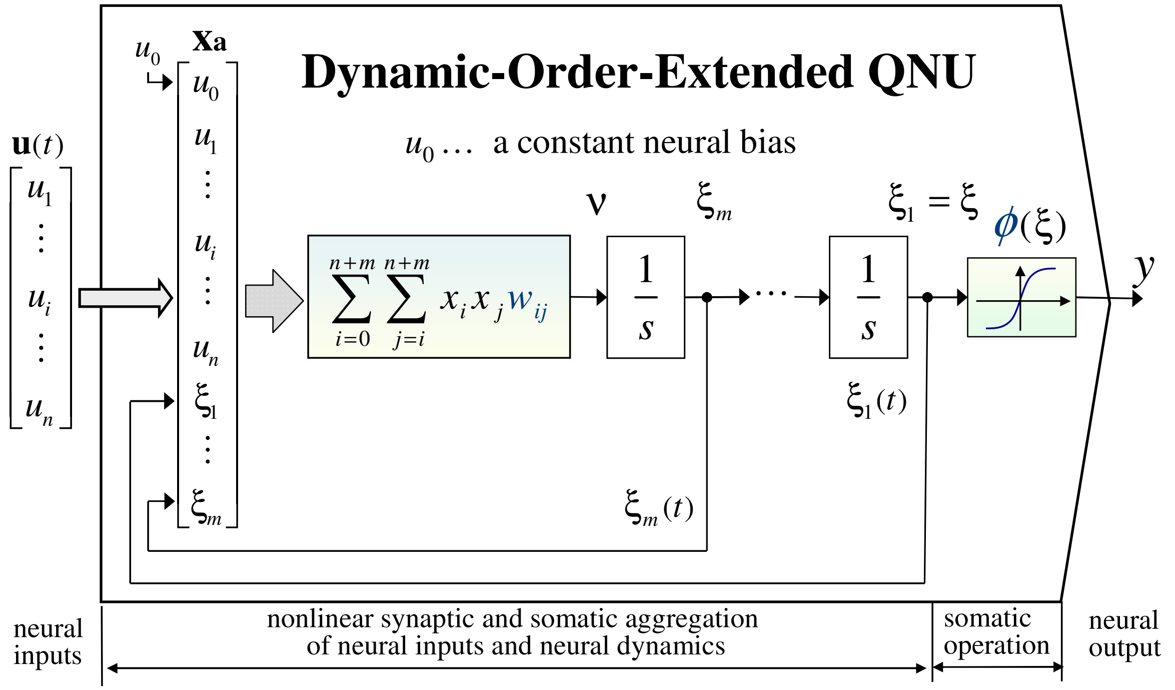


Figure 37: Dynamic quadratic neural unit approximating with the dynamic order of *m*.

The neural output of the dynamic-order-extended QNU, shown in Figure 37, is given by state-space representation in

$$\frac{d^m\xi(t)}{dt^m} = f_{HONNU}(\mathbf{x}\mathbf{a}) = \sum_{i=0}^{m+n}\sum_{j=i}^{m+n} x_i x_j w_{ij} = f_{HONNU}\left(\xi(t), \frac{d\xi(t)}{dt}, \ldots, \frac{d^{m-1}\xi(t)}{dt^{m-1}}, \mathbf{u}(t)\right)$$

$$y(t) = \phi(\xi(t))$$

$$\text{where } \mathbf{x}\mathbf{a} = \begin{bmatrix} x_0 & \ldots & x_i & \ldots & x_{m+n+1} \end{bmatrix}^T = \qquad (4\text{-}41)$$

$$= \begin{bmatrix} 1 & u_1(t) & \ldots & u_j(t) & \ldots & u_n(t) & \xi(t) & \frac{d\xi(t)}{dt} & \ldots & \frac{d^{m-1}\xi(t)}{dt^{m-1}} \end{bmatrix}^T .$$

The neural output of continuous dynamic HONNU in Eq.(4-41), shown as QNU in Figure 37, then results from integrating it with respect to time as

$$y(t) = \phi(\xi(t)) = \phi\left(\int_0^t\int_0^t \ldots \int_0^t f_{HONNU}(\mathbf{x}\mathbf{a}, \mathbf{W}\mathbf{a})\, d\tau^m\right) =$$

$$= \phi\left(\int_0^t\int_0^t \ldots \int_0^t f_{HONNU}\left(\xi(\tau), \frac{d\xi(\tau)}{d\tau}, \ldots, \frac{d^{m-1}\xi(\tau)}{d\tau^{m-1}}(\tau), \mathbf{u}(\tau)\right) d\tau^m\right), \qquad (4\text{-}42)$$

where symbol τ has been used here only for purpose of integration[13].

[13] Contrary to architectures of linear TmD-DNU where τ represents an adaptable neural parameter of neural dynamics, *e.g.*, Eq.(4-19) and Eq.(4-21).

Then, the weight increment for continuous dynamic HONNU can be evaluated for general *m*, according to the same principle of the learning algorithm as

$$\begin{aligned}\Delta w_{ij}(k+1) &= -\frac{1}{2}\mu\frac{\partial\varepsilon^2}{\partial w_i} = -\mu\varepsilon\frac{\partial(y_r(t)-y(t))}{\partial w_i} = \mu\varepsilon\frac{\partial y(t)}{\partial w_i} = \mu\varepsilon\frac{\partial\phi(\xi(t))}{\partial w_i} = \\ &= \mu\varepsilon\frac{\partial\phi}{\partial\xi}\frac{\partial}{\partial w_{ij}}\cdot\left[\int_0^t\int_0^t\dots\int_0^t f_{HONNU}\left(\mathbf{xa}(\tau),\mathbf{Wa}\right)d\tau^m\right] = \\ &= \mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\int_0^t\int_0^t\dots\int_0^t\left(\frac{\partial}{\partial w_{ij}}f_{HONNU}\left(\mathbf{xa}(\tau),\mathbf{Wa}\right)\right)d\tau^m\right] = \\ &= (\text{for the case of QNU}) = \mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\int_0^t\int_0^t\dots\int_0^t\left(x_i(\tau)x_j(\tau)\right)d\tau^m\right].\end{aligned} \tag{4-43}$$

Next, the learning algorithm for adaptable time delays will be expressed for a continuous time-delay dynamic QNU, shown in Figure 38; however, it is possible to generally apply this learning rule to time-delay HONNU implementations in general.

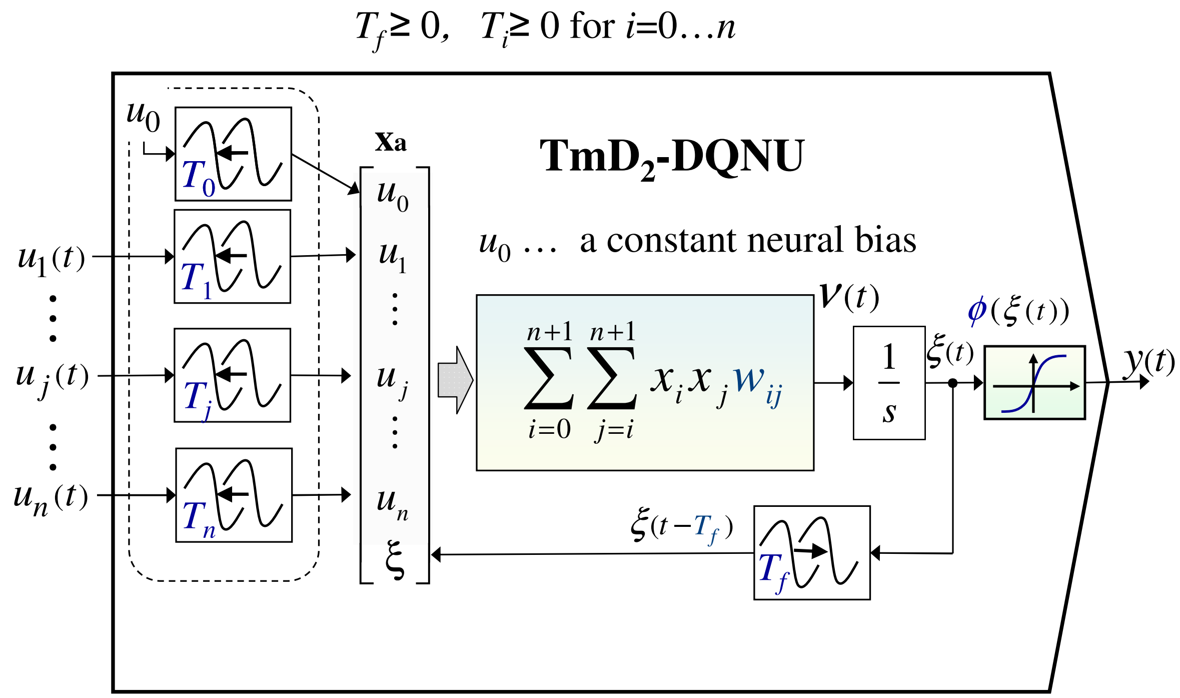


Figure 38: Type-2 continuous time-delay quadratic neural unit; type-2 indicates an adaptable time delay in state feedback of a unit.

Similar to the continuous-time type of HONNU, e.g., in Figure 37 and Eq.(4-41), and due to its nonlinear nature, the design of the learning rule for TmD-DHONNU cannot utilize the Laplace transform while expressing the neural parameter increments, e.g., as in Eq.(4-33), Eq.(4-34), or Eq.(4-35). However, the advantage of the equality shown in Eq.(4-44) can be taken into account while developing the learning rule:

$$\frac{\partial}{\partial T} y(t-T) = \frac{\partial}{\partial T} L^{-1}\left\{Y(s)\cdot e^{-T\cdot s}\right\} = L^{-1}\left\{\frac{\partial}{\partial T}(s)\cdot e^{-T\cdot s}\right\} = \\ = L^{-1}\left\{-s\cdot Y(s)\cdot e^{-T\cdot s}\right\} = -\frac{\partial}{\partial t} y(t-T), \tag{4-44}$$

where the use of the inverse Laplace transform L^{-1} validates the fact that the derivative of time dependent variable $y(t\text{-}T)$ with respect to its time-delay parameter T generally equals the negative value of its derivative with respect to time t.

Simplifying the demonstration of the single-input unit (system identification), the notation of TmD-QNU, shown in Figure 38, is as

$$\frac{d\xi(t)}{dt} = f_{TmD_2-DQNU}\left(\mathbf{xa}, \mathbf{Wa}\right) = \sum_{i=0}^{2}\sum_{j=i}^{2} x_i x_j w_{ij} = f_{TmD_2-DQNU}\left(\xi(t-T_f), u(t-T_1)\right) = \\ = w_{01}u(t-T_1) + w_{02}\xi(t-T_f) + w_{11}u^2(t-T_1) + w_{12}u(t-T_1)\xi(t-T_f) + w_{22}\xi^2(t-T_f) \\ y(t) = \phi(\xi(t)), \tag{4-45}$$

where $\mathbf{xa} = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} 1 & u(t-T_1) & \xi(t-T_f) \end{bmatrix}^T$ and $w_{00}(k)$=0 for any k.

Considering Eq.(4-44) and the fact that the integral of summed terms equals the sum of the integrated terms, the increment of the adaptable time delay T_f in state feedback of QNU from Eq.(4-45) can be evaluated as

$$\Delta T_f(k+1) \approx \mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\int_0^t\left(\frac{\partial}{\partial T_f} f_{TmD_2-DQNU}\left(\mathbf{xa}(\tau), \mathbf{Wa}\right)\right)d\tau\right] \approx$$

$$\approx -\mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\int_0^t\left(w_{02}\dot{\xi}(\tau-T_f) + w_{12}u(\tau-T_1)\dot{\xi}(\tau-T_f) + 2w_{22}\xi(\tau-T_f)\dot{\xi}(\tau-T_f)\right)d\tau\right]$$

$$\Delta T_f(k+1) = -\mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\xi(t-T_f)\left(w_{02} + w_{22}\xi(t-T_f)\right) + \int_0^t\left(w_{12}u(\tau-T_1)\dot{\xi}(\tau-T_f)\right)d\tau\right], \tag{4-46}$$

where $\dot{\xi}(t-T_f) = \nu(t-T_f)$ and

$w_{ij} \rightarrow w_{ij}(t)$ and $T_1 \rightarrow T_1(t)$ (i.e., w_{ij} and T_1 are considered constants except for the integration)

The neural weights w_{ij} vary in time (converge) during adaptation $w_{ij} = w_{ij}(t)$; thus, it is important to explain the simplification in Eq.(4-46). The learning rule works even better if w_{ij} are thought of as being constants contrary to the analytical result in Eq.(4-46), that suggests evauating ΔT_f as

$$\Delta T_f(k+1) \approx \\ \approx -\mu\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\int_0^t\left(w_{02}(\tau)\dot{\xi}(\tau-T_f) + w_{12}(\tau)u(\tau-T_1)\dot{\xi}(\tau-T_f)d\tau + 2w_{22}(\tau)\xi(\tau-T_f)\dot{\xi}(\tau-T_f)\right)d\tau\right]. \tag{4-47}$$

The explanation revealed here results from the nature of adaptation by the gradient learning algorithm itself.

The adaptation of the neural unit starts with initial weights $w_{ij}(0)$, which are presumably chosen as distinct from the desired accurate weights. During the adaptation, the neural weights $w_{ij}(t)$ and other neural parameters converge toward desired, i.e., more accurate, values. If more accurate initial neural weights, represented by $w_{ij}(t)$ for t>0, had been known before adaptation started (t=0), then they should be set as initial weights $w_{ij}(0) \leftarrow w_{ij}(t)$. The integration in Eq.(4-47) would naturally result in even more accurate results for ΔT_f , and in faster learning as well. It is better to consider w_{ij} as more accurate constants rather than less accurate time-varying (converging) parameters. It is even desirable to consider weights w_{ij} (and other neural parameters) as constant when calculating ΔT_f in Eq.(4-46), as well as in the learning rules in general. The precise mathematical notation for evaluating the increment of the neural parameter ΔT_f is

$$\Delta T_f(k+1) = -\mu\,\varepsilon(t)\frac{\partial\phi}{\partial\xi}\cdot$$

$$\cdot\left[\xi(t-T_f(t))\Big(w_{02}(t)+w_{22}(t)\cdot\xi(t-T_f(t))\Big)+\int_0^t\Big(w_{12}(\tau)\cdot u(\tau-T_1(\tau))\cdot\dot{\xi}(\tau-T_f(\tau))\Big)d\tau\right] \qquad (4\text{-}48)$$

where t is a continuous parameter of time.

Considering the neural dynamic architecture in Figure 38, all variables in Eq.(4-48) are directly available, including the time derivative of $\xi(t)$, which is measured directly on the integrator input, where it is denoted as $\nu(t)$ (the output from the aggregation function in Figure 38).

Further, the learning rule for time delay in neural input into a unit T_1 is shown. The learning rule can be simplified again, for the case of single input TmD-DHONNU, and for the neural architecture given in Eq.(4-45) it is evaluated as

$$\Delta T_1(k+1) \approx \mu\,\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\int_0^t\left(\frac{\partial}{\partial T_1}f_{TmD_2-DQNU}\big(\mathbf{xa}(\tau),\mathbf{Wa}\big)\right)d\tau\right]\approx$$

$$\approx -\mu\,\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[\int_0^t\Big(w_{01}\dot{u}(\tau-T_1)+2w_{11}u(\tau-T_1)\dot{u}(\tau-T_1)+2w_{12}\dot{u}(\tau-T_1)\xi(\tau-T_f)\Big)d\tau\right]$$

$$\Delta T_1(k+1) = -\mu\,\varepsilon\frac{\partial\phi}{\partial\xi}\cdot\left[u(t-T_1)\Big(w_{01}+w_{11}u(t-T_1)\Big)+\int_0^t\Big(w_{12}\dot{u}(\tau-T_1)\xi(\tau-T_f)\Big)d\tau\right], \qquad (4\text{-}49)$$

where $w_{ij} \to w_{ij}(t)$ and $T_f \to T_f(t)$ (i.e. w_{ij} and T_f are considered constant except for the integration).

In order to avoid the time derivative of the input $u(t)$, the derivative of the input signal may be unavailable for stand-alone neural units or for units in the first layer of a network, the integration term in Eq.(4-49) can be modified by applying per-partes rule,

$$u(t-T_1)\cdot\xi(t-T_f) = \int_0^t\Big(w_{12}u(\tau-T_1)\dot{\xi}(\tau-T_f)\Big)d\tau+\int_0^t\Big(w_{12}u(\tau-T_1)\dot{\xi}(\tau-T_f)\Big)d\tau. \qquad (4\text{-}50)$$

Then Eq.(4-49) yields

$$\Delta T_1(k+1) =$$

$$= -\mu\,\varepsilon \frac{\partial \phi}{\partial \xi} \cdot \left[u(t-T_1)\Big(w_{01} + w_{11}u(t-T_1) + w_{12}\xi(t-T_f)\Big) - \int_0^t \Big(w_{12}u(\tau-T_1)\dot{\xi}(\tau-T_f)\Big)\,d\tau \right], \tag{4-51}$$

where again no signal has to be derived with respect to time because all variables, including $\dot{\xi}(t)$, can be measured directly within the neural unit.

5 Discrete HONNU and an Adaptive Approach to the Monitoring of Variability of Complex Time-Series

In this section, the approximating capabilities of adaptive HONNU are utilized for monitoring the actual changes in variability of signals generated by complex (chaotic) dynamic systems.

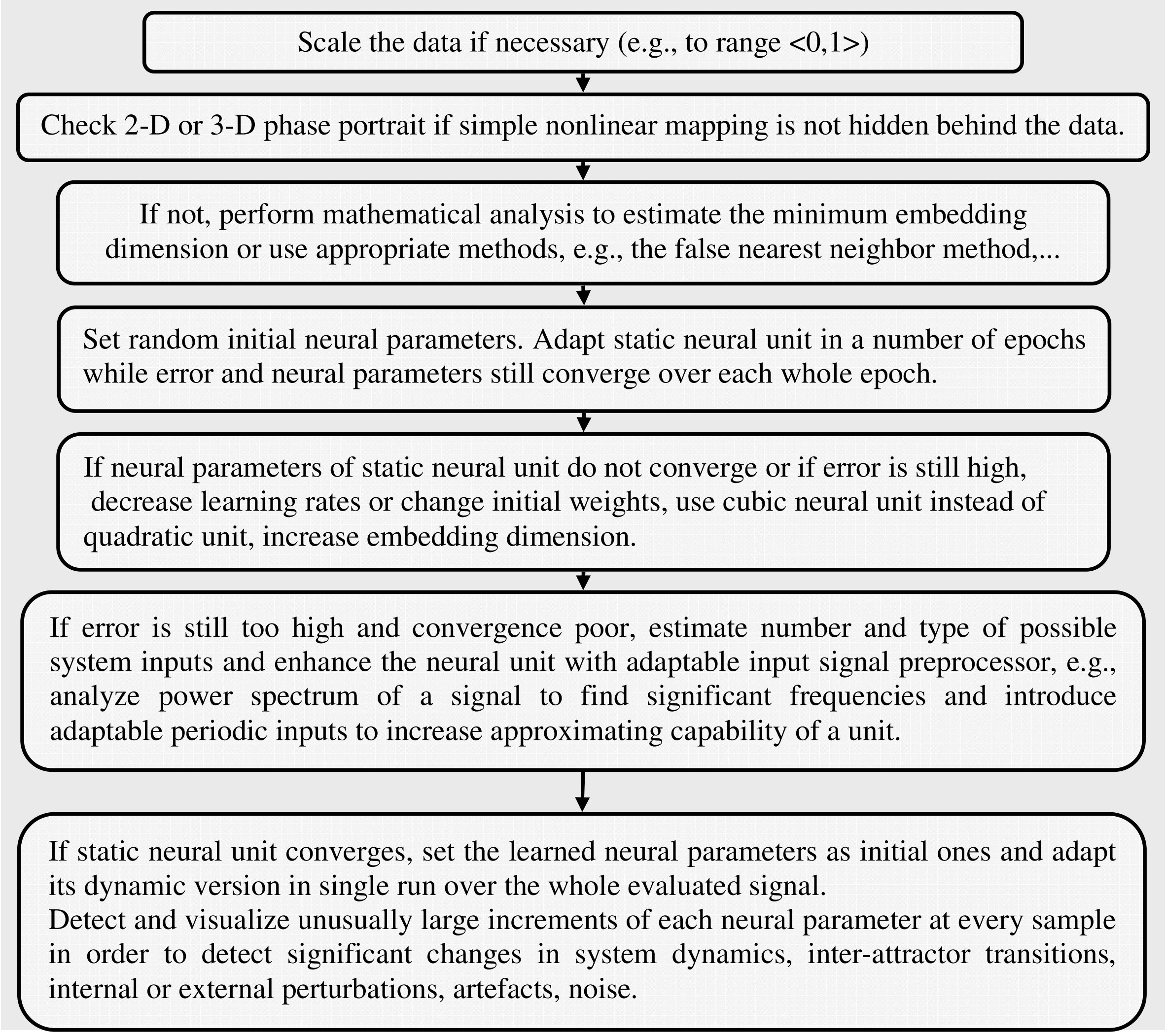
There are two basic approaches to retrieve the underlying dynamics of a system. The first is when the system inputs and outputs are available, and the number of state variables can be found or estimated, e.g., by analyzing the system. The second occurs when only output signals are available, such as with systems evaluations based on time series; the dimension of the state space than has to be reconstructed by appropriate methods, such as the false nearest neighbors method (Kennel *et. al.*1992).

As it can be concluded from section 3.1.2, systems with a potential occurrence of complex behavior shall be approximated by models performing at least in 3-D in order to develop chaotic behavior that is not simply periodic or quasiperiodic (e.g., such as with forced oscillators). However, even non-input single-state-variable nonlinear systems can become chaotic if they are discrete (recurrent), e.g., the well known logistic equation.

A practical technique for the approximation of dynamic systems with stand-alone HONNU has been developed and applied. This technique helps avoid the major instability issues of dynamic HONNU during learning, i.e., during the fine online tuning of a nonlinear dynamic sytem. Neural weights and other neural parameters of HONNU and TmD-DNU converged during adaptation if the neural architecture were of appropriate mathematical structure (such as the polynomials in the case of QNU or CNU), and the initial neural parameters and learning rates were reasonably chosen. It can be deduced from the general properties of the learning algorithm shown in Eq.(4-26), and it was also usually observed during simulation experiments with HONNU and TmD-DNU. Moreover, static HONNU has displayed an excellent ability to converge for the general class of nonlinear aggregating functions without limitations upon the monotonousness and bounding, as with the use of higher-order polynomials or exponential functions. During simulation experiments, the static variations of HONNU (Figure 24) were trained to identify the structure and coefficients of a nonlinear governing equation. Assuming that a nonlinear governing equation is contained as a subset of the synaptic and somatic aggregation operation f_{HONNU}, the learning process is stable for manually chosen learning rates and for random initial neural weights within the range (-0.5 , 0.5). The chosen learning rates and initial neural weights mean such values for which the learning algorithm was stable, neural parameters converged, and the values can be found on the first attempt or can be found manually by a few trial and error attempts.

Since the mathematical structure of HONNU performs polynomial approximation, the unit can be enhanced further with additional mathematical functions other than polynomials, and a stand-alone HONNU can be used for approximations of wide range of complex systems. Dynamic HONNU also copes with stability problems during adaptation because of the generally nonlinear and unbounded function f_{HONNU} implemented in its synaptic operation.

This observation has lead to the combination of static and dynamic HONNU in order to provide dynamic HONNU with stability during learning. The initial values for dynamic HONNU can be approached first by those of static HONNU. The dynamic HONNU, which corresponds to the already adapted static one, can be then run from those initial values. Then, the stable and converging learning process of a dynamic HONNU can be observed. This second step also provides finer tuning of neural weights by dynamic modification of the backpropagation learning algorithm. The following diagram roughly summarizes the steps for approximating systems by discrete HONNU and consequently for detecting changes of signal variability, i.e., to approximate systems for which discrete complex output signals (time series) are available, such as heart beat tachograms,…



Scale the data if necessary (e.g., to range <0,1>)

Check 2-D or 3-D phase portrait if simple nonlinear mapping is not hidden behind the data.

If not, perform mathematical analysis to estimate the minimum embedding dimension or use appropriate methods, e.g., the false nearest neighbor method,...

Set random initial neural parameters. Adapt static neural unit in a number of epochs while error and neural parameters still converge over each whole epoch.

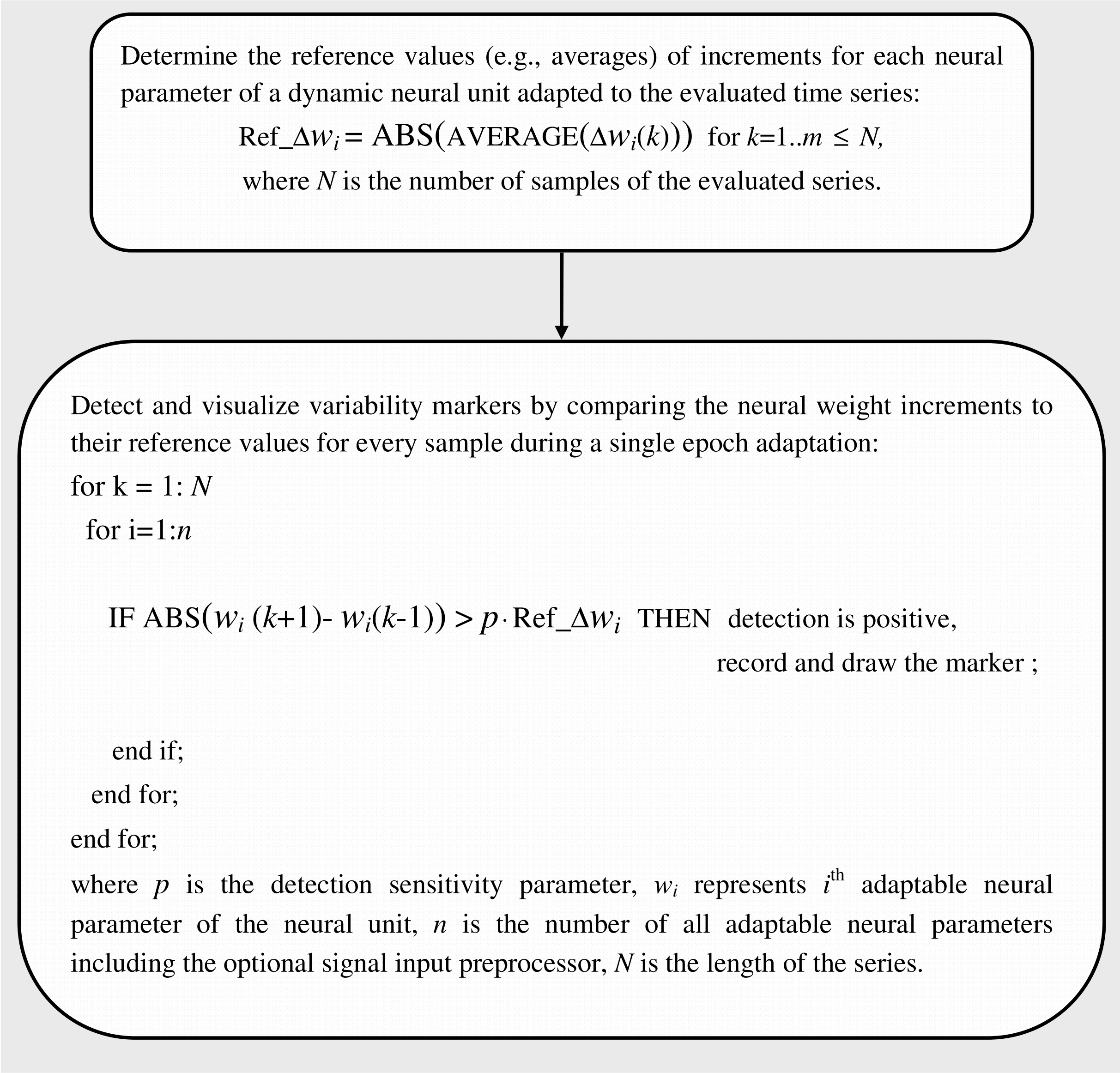
If neural parameters of static neural unit do not converge or if error is still high, decrease learning rates or change initial weights, use cubic neural unit instead of quadratic unit, increase embedding dimension.

If error is still too high and convergence poor, estimate number and type of possible system inputs and enhance the neural unit with adaptable input signal preprocessor, e.g., analyze power spectrum of a signal to find significant frequencies and introduce adaptable periodic inputs to increase approximating capability of a unit.

If static neural unit converges, set the learned neural parameters as initial ones and adapt its dynamic version in single run over the whole evaluated signal.
Detect and visualize unusually large increments of each neural parameter at every sample in order to detect significant changes in system dynamics, inter-attractor transitions, internal or external perturbations, artefacts, noise.

Figure 39: The technique for system approximation and consequent monitoring of changes of signal variability by both discrete static and dynamic HONNU or HRV-HONNU (see Figure 40).

The detection of every unusual increment of each neural parameter in discrete dynamic neural units according to the last point in Figure 39 is described in Figure 40. The parameter p in Figure 40 is called the detection sensitivity parameter; the higher the value of p, the more significant the weight increments detected, but the less the number of markers drawn.



Determine the reference values (e.g., averages) of increments for each neural parameter of a dynamic neural unit adapted to the evaluated time series:

$$\text{Ref\_}\Delta w_i = \text{ABS}(\text{AVERAGE}(\Delta w_i(k))) \;\text{ for } k=1..m \le N,$$

where N is the number of samples of the evaluated series.

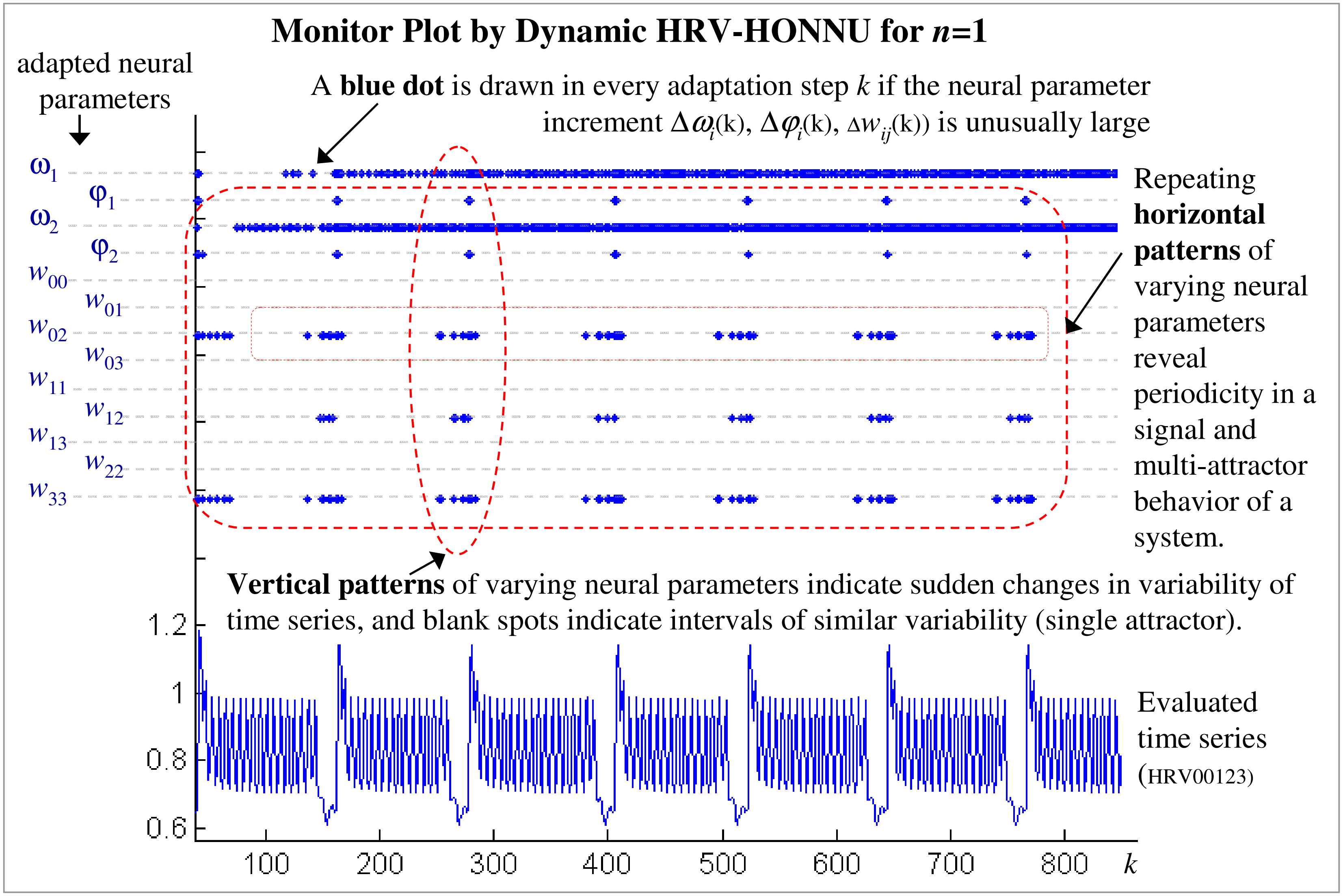
Detect and visualize variability markers by comparing the neural weight increments to their reference values for every sample during a single epoch adaptation:

```
for k = 1: $N$
  for i=1:$n$

    IF $\text{ABS}(w_i\,(k+1)\text{-}\,w_i(k\text{-}1)) > p\cdot\text{Ref\_}\Delta w_i$ THEN detection is positive,
                                                                record and draw the marker ;

    end if;
  end for;
end for;
```

where p is the detection sensitivity parameter, w_i represents i^{th} adaptable neural parameter of the neural unit, n is the number of all adaptable neural parameters including the optional signal input preprocessor, N is the length of the series.

Figure 40: The technique to detect and record variability markers (blue dots) for the monitor plot in .

The plot of markers detecting the distinct neural parameter increments is called the monitor plot in this work. The monitor plot and its functionality are explained in Figure 41.



Monitor Plot by Dynamic HRV-HONNU for *n*=1

adapted neural parameters

A **blue dot** is drawn in every adaptation step *k* if the neural parameter increment $\Delta\omega_i$(k), $\Delta\varphi_i$(k), Δw_{ij}(k)) is unusually large

Repeating **horizontal patterns** of varying neural parameters reveal periodicity in a signal and multi-attractor behavior of a system.

Vertical patterns of varying neural parameters indicate sudden changes in variability of time series, and blank spots indicate intervals of similar variability (single attractor).

Evaluated time series (HRV00123)

Figure 41: The monitor plot for the detection of changes in variability, transititions to another attractor, artefacts, system perturbations, (LLE=0.9 and CD=0.88 by the program Dataplore).

6 An Adaptive Approach to the Evaluation of Heart Rate Variability: HRV-HONNU

Based on the neural architectures developed in chapter 4 and on the achievements shown in chapter 3, the principle of the novel adaptive method for real-time monitoring of changes in the dynamics of complex (chaotic) dynamic systems is introduced in this chapter. The method focuses on monitoring of actual changes in the dynamics of heart rate variability for diagnostic purposes. As the method is designed for complex (biological) systems, it is also very promising for applications with technical systems, especially for adaptive control and fault detection; the results on system identification and control underline this proposal [50] to [52] [54] [56] to [59] .

All of the proposed neural units can test an unknown system for the existence of the deterministic nature of its behavior by observing the actual convergence of the neural parameters. A deterministic system shall result in converging neural weights of an adapted neural unit; the simpler the system or system output (i.e., the simpler the approximating model), the better the convergence of the neural parameters expected during approximation by a neural unit.

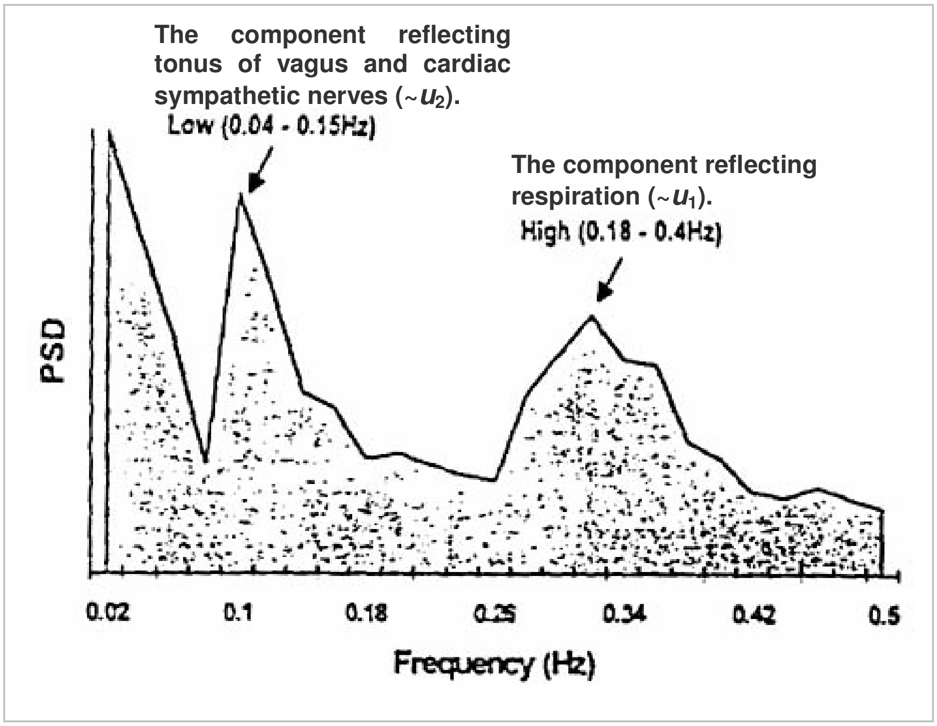


Figure 42: Power spectral density of heart beat tachogram (picture copied and modified from : John D. and Catherine T. MacArthur Research Network on Socioeconomic Status and Health. http://www.macses.ucsf.edu/Research/Allostatic/notebook/heart.rate.html, retrieved 01/2007).

In section 3.2, it was proposed that the dynamics of the cardiovascular system has a significant deterministic-chaos component due to the fast beat-by-beat control influences of the autonomous nervous system (ANS). In other words, it has been shown that distinct physiological time delays in the transfer of information from baroreceptors to the brain and back to the heart tissue result in significantly distinct heart rate variability, such as from periodic to highly chaotic [18] to [23].

Previous experiments [53] have displayed convergence of stand-alone static HONNU and consequently of its discrete dynamic modification (discrete dynamic HONNU) during adaptation to heart rate time series; however, the error of the adapted unit was still significant. That error is currently lowered by implementing additional *apriori* knowledge into the HONNU, resulting in the neural unit called heart rate variability HONNU (HRV-HONNU) shown in Figure 43.

It is well known amongst physiologists that heart rate periodically reflects breathing frequency and displays an intrinsic frequency component due to the tonus of vagus nerves (Figure 42). This knowledge can be implemented into HRV-HONNU (Figure 43 and Figure 44) as input signals featuring the two remarkable frequencies.

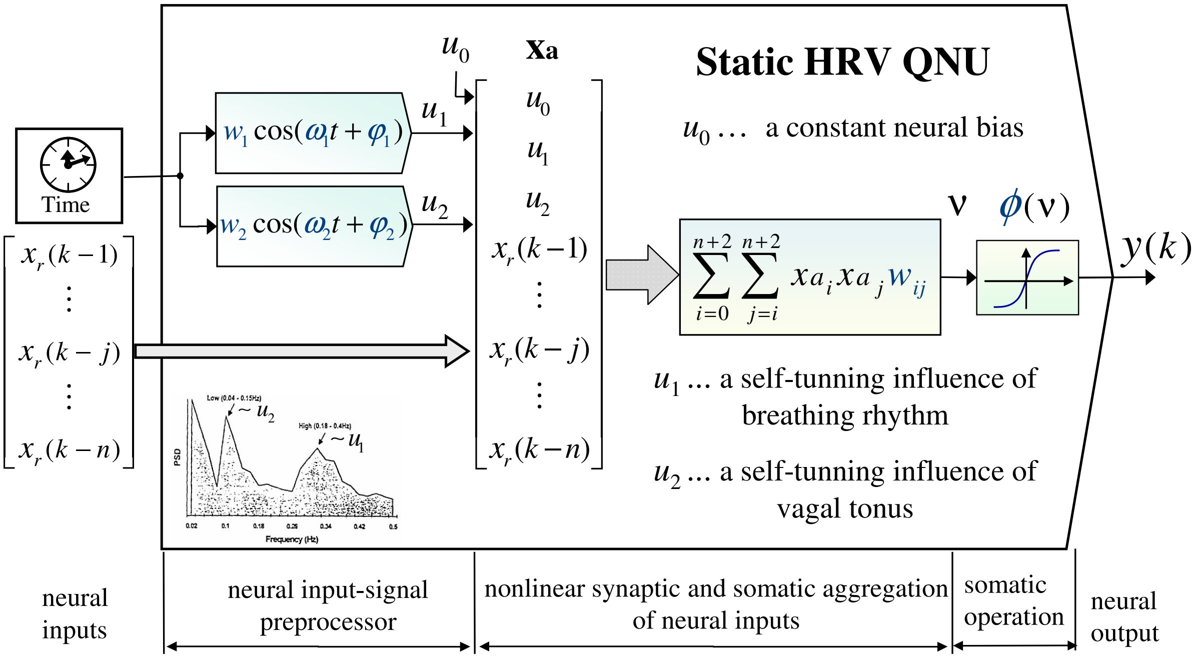

Static HRV QNU
Time
$x_r(k-1)$
$x_r(k-j)$
$x_r(k-n)$
$w_1\cos(\omega_1 t+\varphi_1)$
$w_2\cos(\omega_2 t+\varphi_2)$
u_0
u_1
u_2
Xa
u_0
u_1
u_2
$x_r(k-1)$
$x_r(k-j)$
$x_r(k-n)$
u_0 … a constant neural bias
$\sum_{i=0}^{n+2}\sum_{j=i}^{n+2} xa_i xa_j w_{ij}$
ν
$\phi(\nu)$
$y(k)$
u_1 ... a self-tunning influence of breathing rhythm
u_2 ... a self-tunning influence of vagal tonus
Low (0.04 - 0.15Hz)
$\sim u_2$
High (0.18 - 0.4Hz)
$\sim u_1$
PSD
0.02
0.1
0.18
0.26
0.34
0.42
0.5
Frequency (Hz)
neural inputs
neural input-signal preprocessor
nonlinear synaptic and somatic aggregation of neural inputs
somatic operation
neural output

Figure 43: Static heart rate variability quadratic neural unit HRV-QNU. Again, the blue color highlights the adaptable neural parameters. Input-signal preprocessor introduces breathing (u_1) and vagal tonus (u_2) frequencies.

The exact breathing frequency and vagal tonus are not known when only the heart rate is being recorded. The phase-delay between heart beat, breathing, and vagal tonus is not known, nor can the amplitude of these signals be easily measured. Excelent cognitive ability allows HRV-HONNU to learn the appropriate configuration of these parameters by the backpropagation learning rule in the neural-input-signal preprocessor in Figure 43 and Figure 45. The approximating capability of HRV-HONNU is improved for adaptation to the dynamics of heart beat rhythm, contrary to the results in [53] where pure HONNU were used only. This improvement of HONNU, by the signal preprocessor, has its physiological analogy to cardiovascular dynamics (Figure 42).

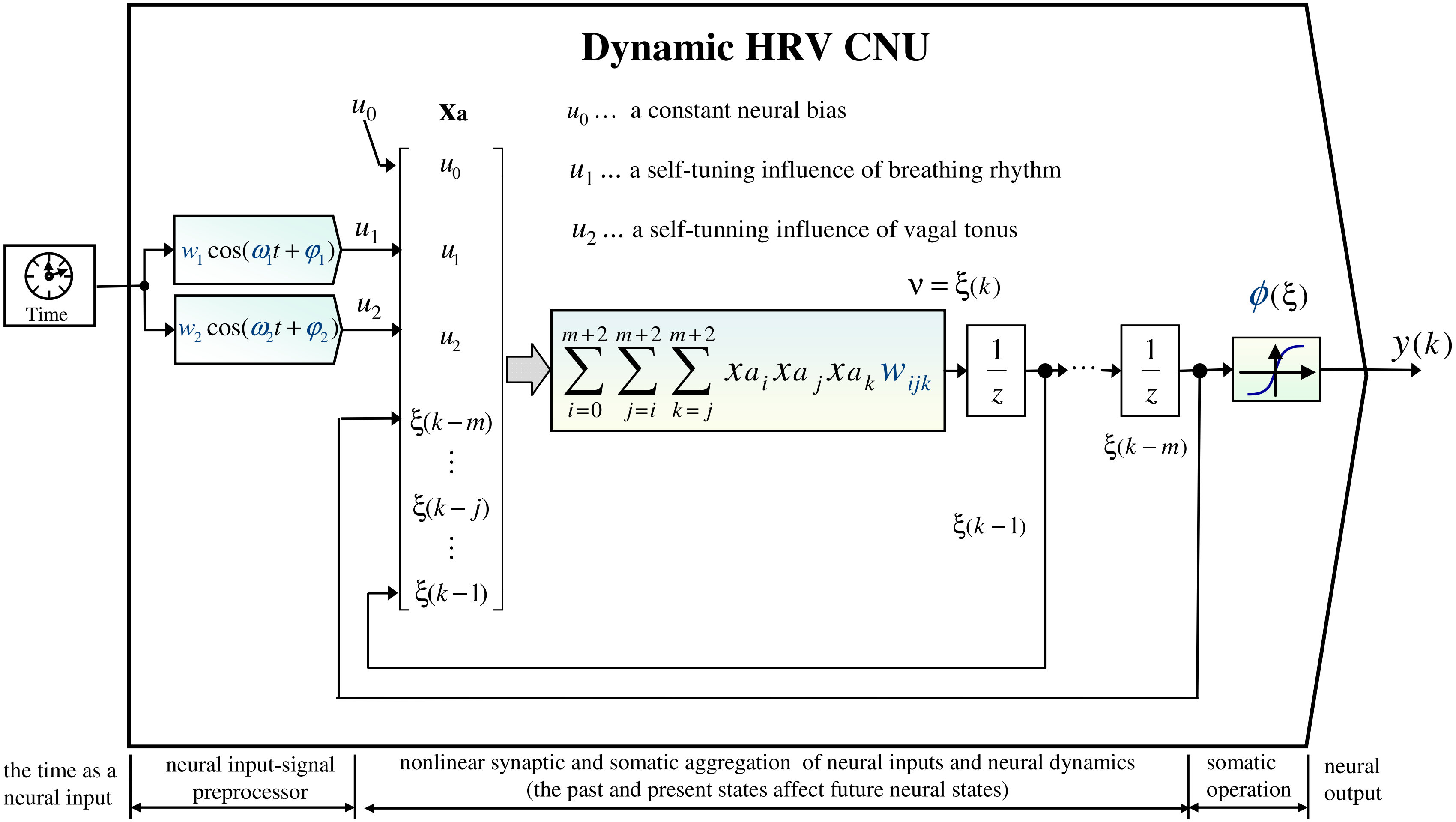

Dynamic HRV CNU
Time
$w_1\cos(\omega_1 t+\varphi_1)$
$w_2\cos(\omega_2 t+\varphi_2)$
u_0
u_1
u_2
$\mathbf{xa}$
u_0
u_1
u_2
$\xi(k-m)$
$\xi(k-j)$
$\xi(k-1)$
u_0 … a constant neural bias
u_1 … a self-tuning influence of breathing rhythm
u_2 … a self-tunning influence of vagal tonus
$\nu=\xi(k)$
$\sum_{i=0}^{m+2}\sum_{j=i}^{m+2}\sum_{k=j}^{m+2} xa_i\, xa_j\, xa_k\, w_{ijk}$
$\frac{1}{z}$
$\frac{1}{z}$
$\xi(k-m)$
$\xi(k-1)$
$\phi(\xi)$
$y(k)$
the time as a neural input
neural input-signal preprocessor
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics (the past and present states affect future neural states)
somatic operation
neural output

Figure 44: Dynamic heart rate variability HONNU. Input-signal preprocessor introduces breathing (u_1) and vagal tonus (u_2) frequencies.

The learning rule for HRV-HONNU follows the principles shown in subsection 4.4.3, where the input-signal preprocessor is adapted by the same backpropagation rules.

Both the static and the dynamic HRV-HONNU were used by the technique introduced in section 5. The significant variations of adaptable neural parameters and the observation of their patterns in the monitor plot **(),** as well as their possible evaluation during the adaptation of dynamic HRV-HONNU, establishes a novel method for both monitoring sudden changes in HRV and revealing intervals of similar HRV in heart beat tachograms. In fact, it does not matter if the changes of system dynamics are caused by external or internal perturbations or transitions among attractors for constant system parameters. According to the actual accuracy of HRV-HONNU approximating a particular system, this method is suitable to distinguish between the transition to another attractor and the internal parameter perturbations by the simultaneous observation of both the actual values of the adapted neural parameters as well as their variations during adaptation. Moreover, external perturbations to the system may be detected using the adaptable input-signal preprocessor of dynamic HRV-HONNU shown in Figure 44. Both of the previously mentioned capabilities of the proposed method are a matter for further research.

The results on the monitor plot presented in section 7.3 demonstrate the capability of this methodology to reveal sudden changes and to reveal intervals of a similar level of chaos even in such complex (chaotic) time series that common nonlinear methods have not achieved reliable results.

In section 7.3 (page 96), the monitor plots of both pure deterministic time series of a high level of chaos (section 3.2 above, [23]) and real R-R diagrams from the MIT Arrhythmia database [17] are shown.

7 Applications and Results

7.1 IDENTIFICATION OF TIME DELAYS AND SYSTEM APPROXIMATION USING TmD-DNU

Promising results were obtained during approximation of linear systems with dynamics of high orders and systems with time delays. These results are briefly presented below and underline the good approximating capability of time-delayed neural units as well as the excellent stability of the learning algorithm.

7.1.1 Identification of Time Delays in Linear Systems

An example demonstrating the capabilities of the stand-alone TmD_1-DNU shown in Eq.(4-19) (Figure 26) (Figure 27) follows. The dynamic system to be identified by TmD_1-DNU in this example is a linear plant with input time-delay as follows

$$2y_r'(t) + y_r(t) = 2.5u(t-1.5)\,. \tag{7-1}$$

Parameters of the plant in Eq.(7-1) will be identified by the proposed TmD_1-DNU in Eq.(4-19), and are also shown in Figure 26 and Figure 27.

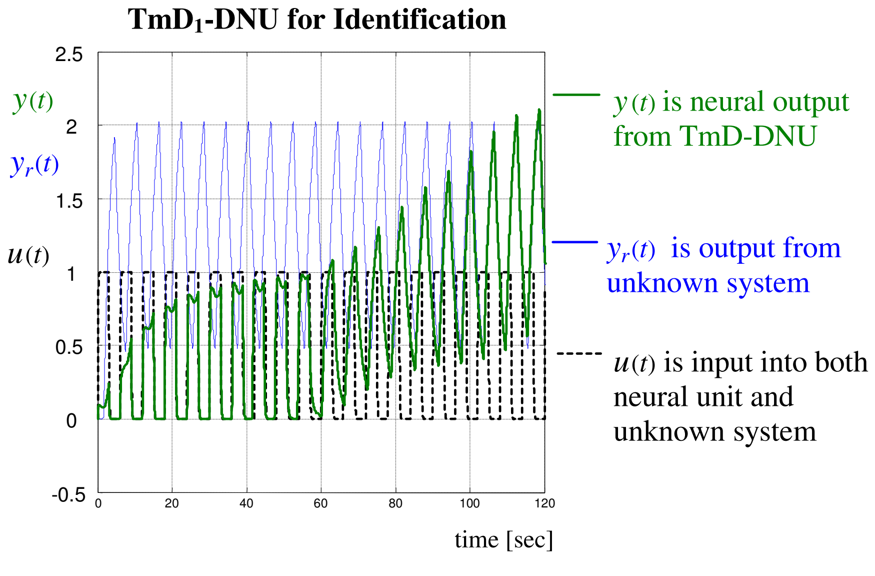


Figure 45: Adaptation of TmD-DNU–Type 1, Eq.(4-19), (Figure 26, Figure 27), with linear neural somatic operation $\phi(\,)$. The neural unit performs identification of a dynamic system with time-delayed input in Eq.(7-1).

The initial neural parameters, which are the learning rate μ,□ the minimum time constant τ_{min}, the initial neural weight $w_1(0)$, and other neural parameters $\tau(0)$ and $T_1(0)$ that represents the adaptable time delay, were set as follows

$$\mu = 10^{-3}, \quad \tau_{\min} = 10^{-3}\,\text{sec}, \quad w_1(0) = \tau(0) = T_1(0) \; = 0.1. \tag{7-2}$$

In this example, the common input $u(t)$ into both the identified plant and the dynamic neural unit, Eq.(4-19), is a periodic square signal measured on the output of a linear plant with a fast time constant (a simple low-pass filter) in order to achieve a smoother input signal. The neural weights were adapted by the learning rules described in Eq.(4-29). The neural parameter T_1 representing the adaptable time-delay parameter was adapted by Eq.(4-30), and is shown in Eq.(4-33) and in Figure 33. The process of adaptation of the unit in Eq.(4-19) for this example is shown in Figure 45.

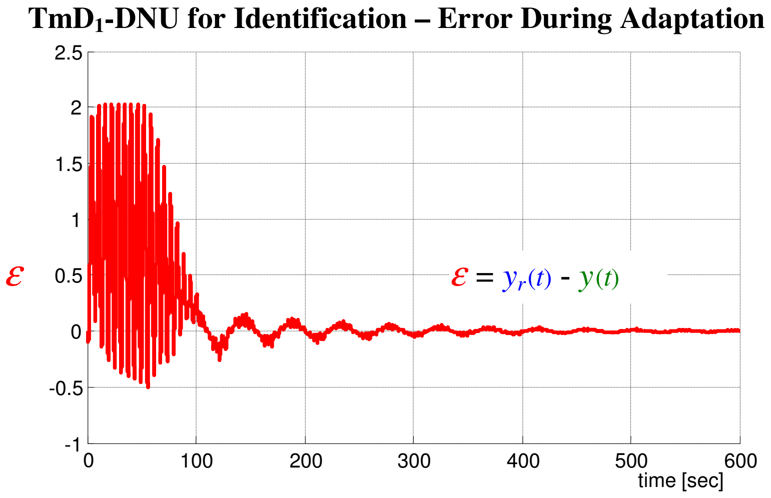

TmD₁-DNU for Identification – Error During Adaptation

Figure 46: Error convergence of TmD-DNU–Type 1 for the example of identification according to Figure 45.

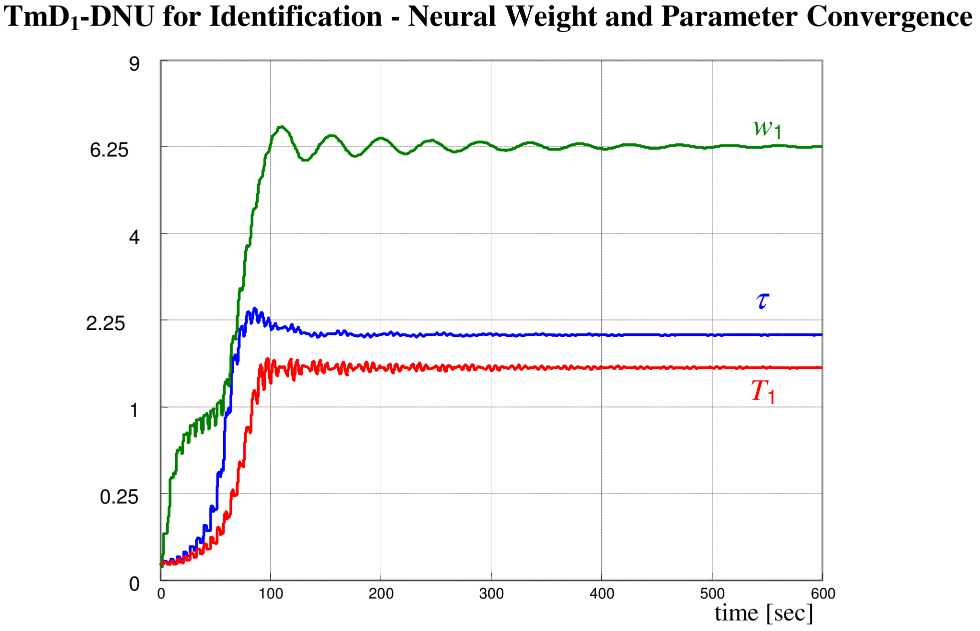

TmD₁-DNU for Identification - Neural Weight and Parameter Convergence

Figure 47: Convergence of neural weight and parameters of TmD₁-DNU for identification of the system in Eq.(7-1) according to Figure 45; neural parameter T_1 represents the continually adaptable parameter of time delay ($T_1 =$ 1.5 sec).

7.1.2 Approximation of Linear Higher-Order Dynamic Systems

Another example of approximating capabilities of the stand-alone TmD$_1$-DNU in Eq.(4-19) and (Figure 26) (Figure 27) is shown in this subsection. A real system to be approximated is a linear higher-order dynamic plant represented by a continuous transfer function (7-3) without time delays.

$$G_r(s) = \frac{Yr(s)}{U(s)} = \frac{360}{s^4+17\cdot s^3+84\cdot s^2+48\cdot s+80}\,. \tag{7-3}$$

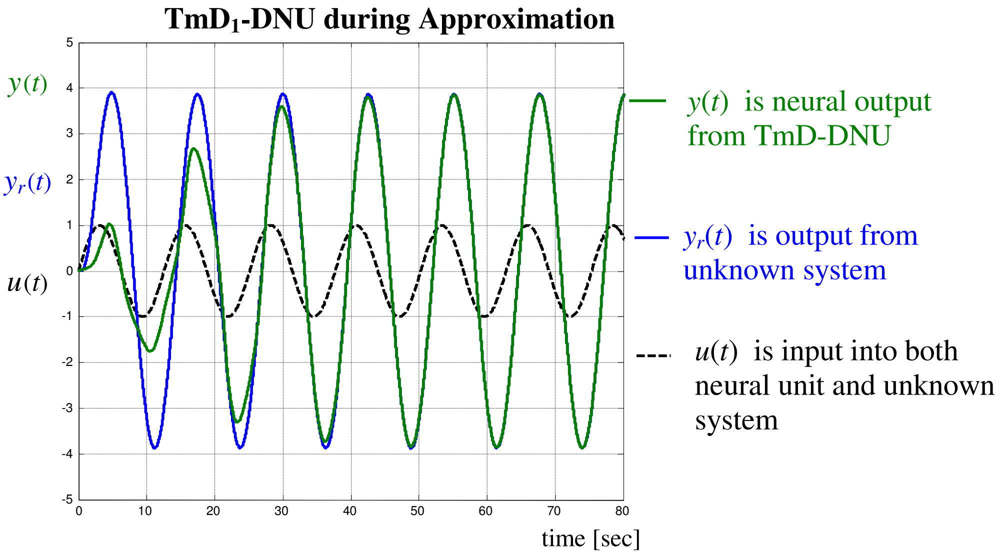


Figure 48: The approximation of a 4th-order dynamic system, Eq.(7-3), by unbiased (w_0 = 0) TmD$_1$-DNU (Figure 26) with a linear neural somatic operation $\phi()$.

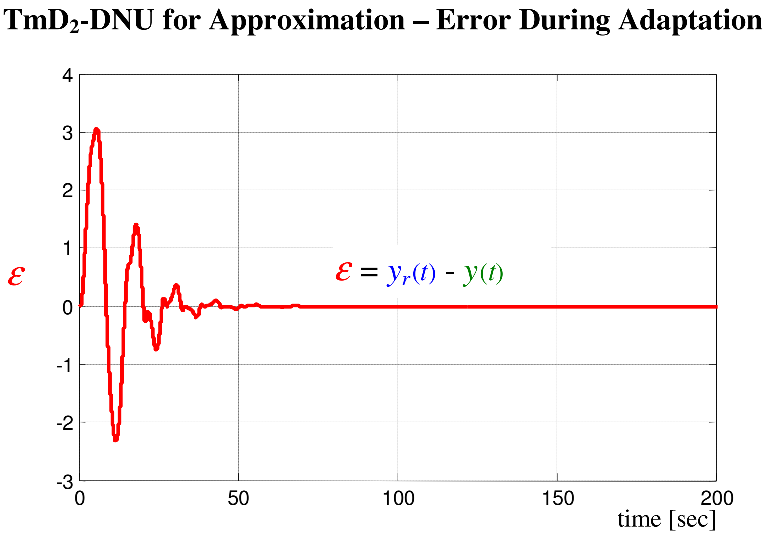


Figure 49: Error convergence of TmD$_1$-DNU for the example of approximation with the sinus input signal according to Figure 48.

The plant, Eq.(7-3), was approximated with TmD$_1$-DNU, representing a different internal dynamic structure in this case. The initial neural parameters of the unit were similar to those in the previous example Eq.(7-2). The common input $u(t)$ into the identified plant and the dynamic neural unit (Figure 31) is a periodic sinus signal (Figure 48).

TmD$_1$-DNU for Approximation - Neural Weight and Parameter Convergence

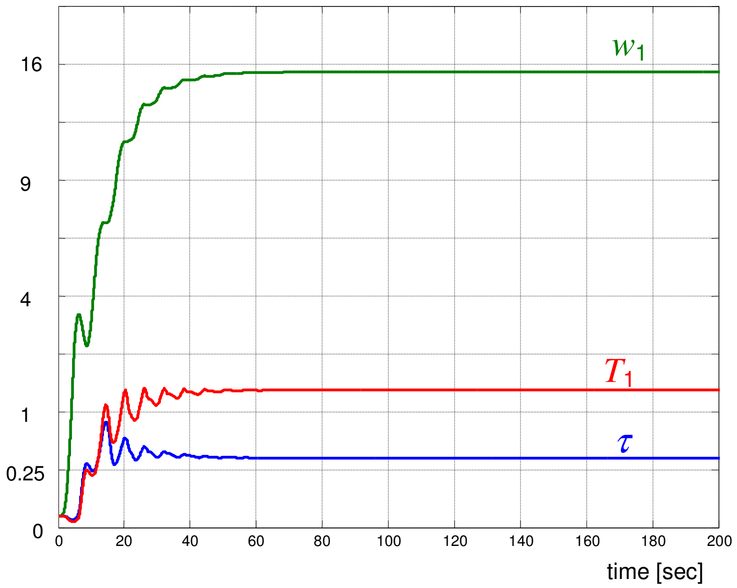


Figure 50: Convergence of neural weights of TmD$_1$-DNU for approximation according to Figure 49. Neural weight T_1 represents the continually adaptable parameter of time delay.

7.1.3 Case 2.b – Agreement in Step Response of Approximated System and Adapted TmD-DNU–Type 1

Even though Case 2.a displays the good error and weight convergence (Figure 49 and Figure 50) of the Type 1 unit (Figure 26) (Figure 27) while approximating system Eq.(7-3) with the sinus input signal (Figure 48), the step response of the learned unit still does not agree precisely (Figure 51) with the actual step response of the real system Eq.(7-3). It could be assumed that the disagreement (Figure 51) of adapted neural unit with the approximated system Eq.(7-3) might have been improved through out the input signal. The periodic square input signal was applied to the next adaptation of the neural unit, and the converged values of previously adapted neural weights were applied as further initial values (Figure 50)

$$\mu = 10^{-5}, \quad \tau_{\min} = 10^{-3} \text{ sec}, \quad \tau = 0.36, \quad w_1 = 15.463, \; T_1 = 1.4087\,. \qquad (7\text{-}4)$$

The choice of significantly lower values of the learning rate μ improved both the stability of the learning algorithm and the ability of neural weights to converge sufficiently close to the minimum of the performance index J (e.g., Figure 31). The lower value of μ and the square input signal resulted in a significantly longer adaptation process (Figure 52, Figure 53); however, the increase in the accuracy of the approximation is crucial (Figure 54).

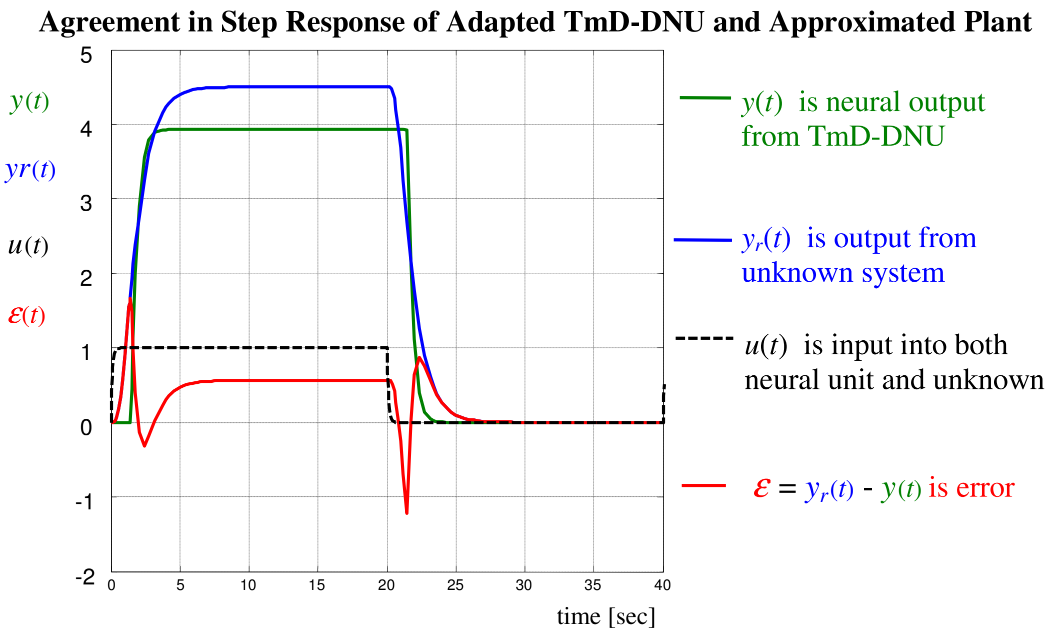


Figure 51: The step response of adapted TmD$_1$-DNU shown in Figure 26 adapted to system in Eq.(7-3) with sinus input signal according to Figure 48. Further adaptation resulted in better agreement (Figure 54).

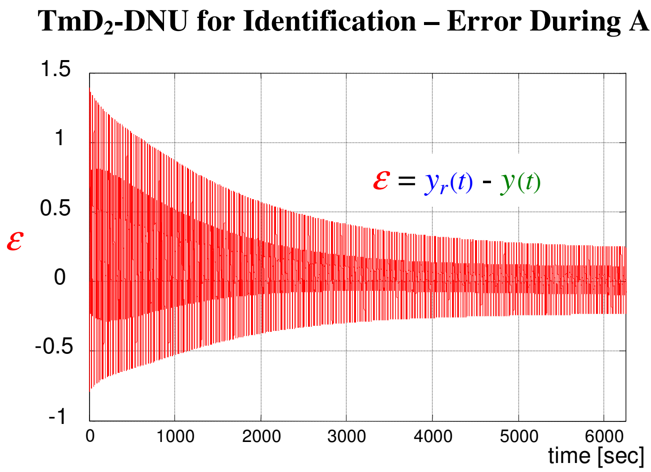


Figure 52: Error convergence of TmD$_1$-DNU adapted to system Eq.(7-3) with periodic square input signal and initial neural weights as in Eq.(7-4).

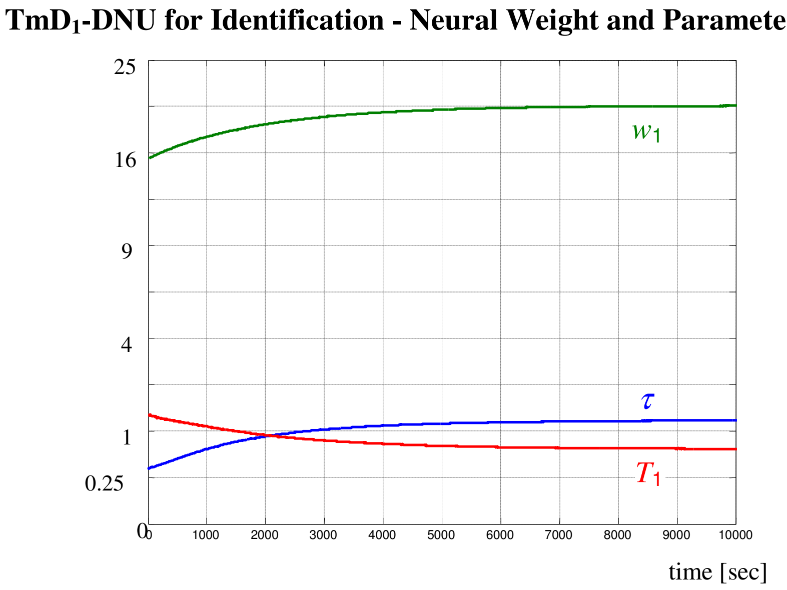


Figure 53: Convergence of neural weights of TmD$_1$-DNU adapted to system Eq.(7-3) with periodic square input signal and initial neural weights as in Eq.(7-4). Neural weight T_1 represents the continually adaptable parameter of time delay.

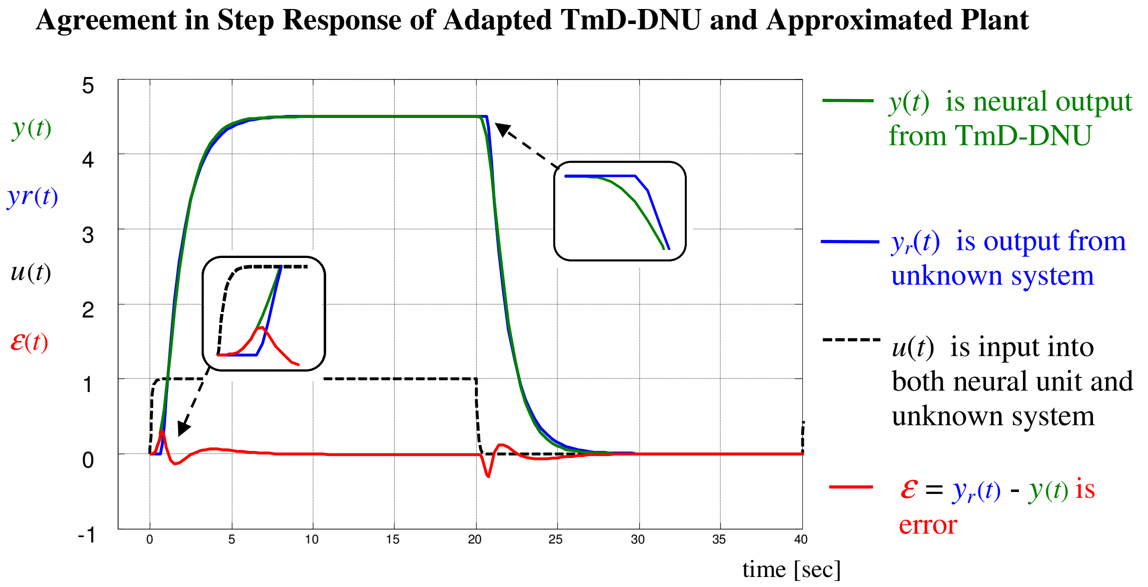


Figure 54: The step response of TmD$_1$-DNU adapted to system Eq.(7-3) with periodic square input signal and initial neural weights Eq.(7-4) obtained from previous adaptation with sinus input signal (Figure 50).

7.1.4 Identifying Capabilities of Time Delay Neural Units – Type 2 (TmD$_2$-DNU)

In this part, the error convergence of the TmD$_2$-DNU will be shown, as the unit Eq.(4-21) has been adapted to system

$$
\begin{gathered}
14.8\,\frac{d\xi(t)}{dt}+\xi(t-6.04)=u(t-11.25)\\
y(t)=\phi(\xi(t))=\xi(t),
\end{gathered}
\tag{7-5}
$$

which represents the same dynamic structure as the TmD$_2$-DNU (Eq.(4-21)) (Figure 29).

Following the stability condition in Eq.(4-37) for time-delay systems as by Eq.(4-21) (introduced in [45] [46]), the initial neural parameters for adaptation of TmD$_2$-DNU were set as in Eq.(7-6).

$$
\begin{gathered}
\mu_{\tau}=\mu_{T_1}=\mu_{T_f}=1\mathrm{e}-3,\ \mu_{w_1}=1\mathrm{e}-4\\
\tau_{\min}=0.1,\quad \tau(0)=T_1(0)=T_f(0)=5,\ w_1(0)=0.5\ \ .
\end{gathered}
\tag{7-6}
$$

The TmD$_2$-DNU started from the initial settings thst represented the following system

$$
\begin{gathered}
25.1\,\frac{d\xi(t)}{dt}+\xi(t-25)=0.5\,u(t-25)\\
y(t)=\phi(\xi(t))=\xi(t)\ \ .
\end{gathered}
\tag{7-7}
$$

The training input signal ($u(t)$ in Figure 55) consisted of a quasiperiodic sinus wave component and a random square wave component (two sinus waves with mutually incommensurate frequencies of 1/5 and 1/8 rad/sec plus one random square wave component with mean = 0, variance = 0.5 and sample time = 10 sec) . The adaptation is depicted in Figure 55 to Figure 57.

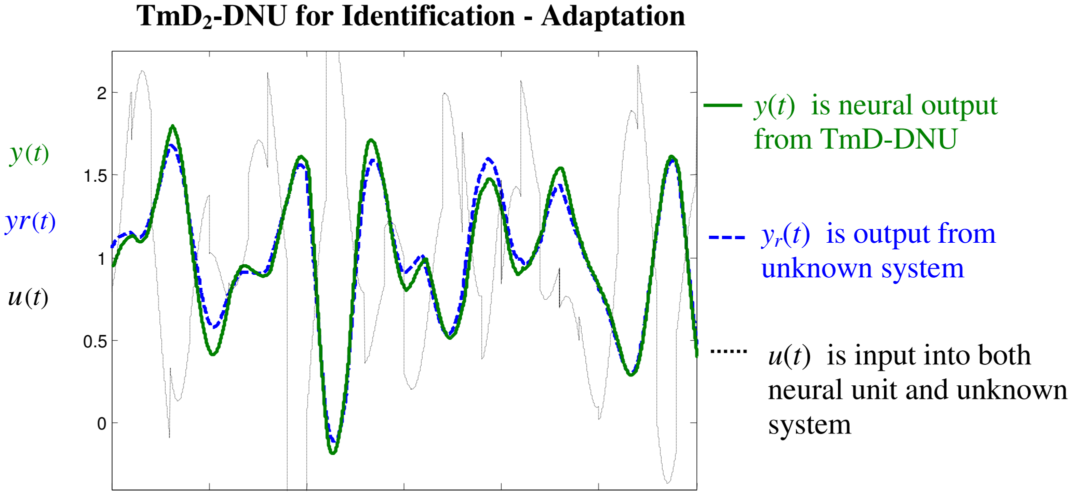
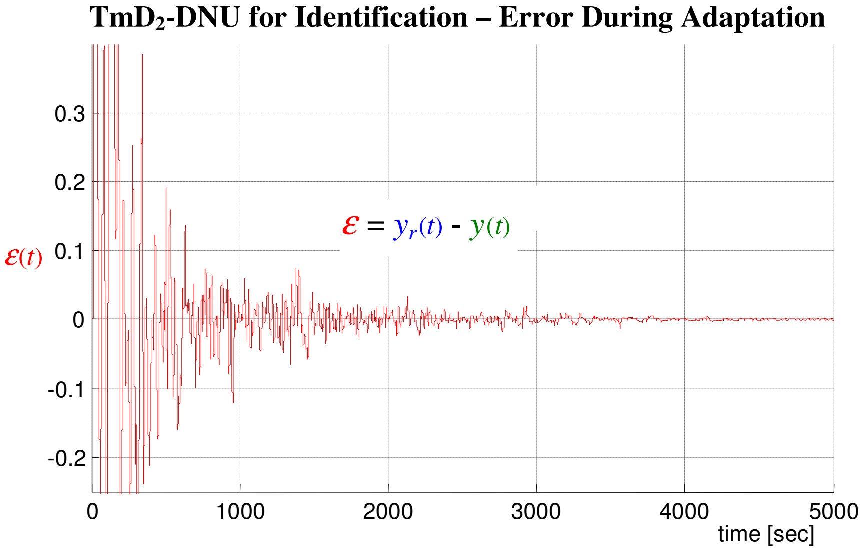
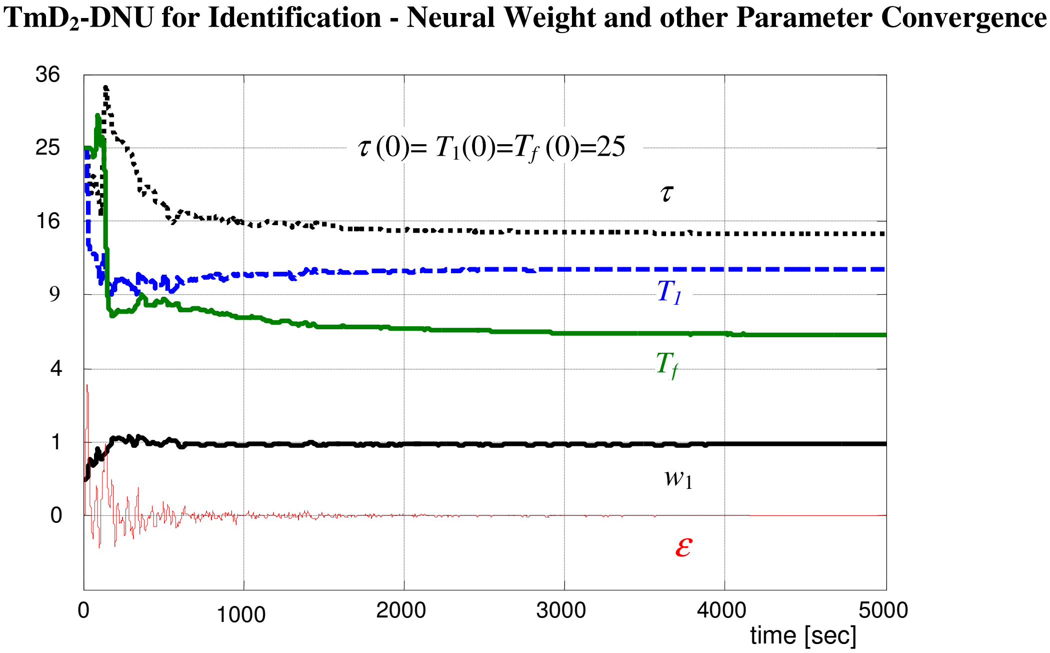


Figure 55: Detail of adaptation of TmD$_2$-DNU with linear output function $\phi(\ )$. The neural unit performs identification of a dynamic system Eq.(7-5) with both time-delayed input and state variable starting from initial conditions in Eq.(7-6).



TmD$_2$-DNU for Identification – Error During Adaptation

Figure 56: Output error during the convergence of TmD$_2$-DNU adapted to system Eq.(7-5) starting from initial neural weights as in Eq.(7-6) according to Figure 55.



TmD$_2$-DNU for Identification - Neural Weight and other Parameter Convergence

Figure 57: Convergence of neural weights of TmD$_2$-DNU for identification of Eq.(7-5) according to Figure 55 and Figure 56. Neural weights T_1 and T_f represent the continually adaptable parameters of time delays (see Table 2)

<table>
<tr><th colspan="3">Identification by TmD-DNU–TYPE 2 - Adaptation Accuracy</th></tr>
<tr><td><b>TmD$_2$-DNU:</b></td><td>$(\tau+\tau_{\min})\dfrac{d\xi(t)}{dt}+\xi(t-T_f)=w_1\cdot u(t-T_1)$</td><td><b>(4-21)</b></td></tr>
<tr><td><b>To be adapted to:</b></td><td>$14.8\,\dfrac{d\xi(t)}{dt}+\xi(t-6.04)=u(t-11.25)$</td><td><b>(7-5)</b></td></tr>
<tr><td colspan="2"><b>Adaptation Mode (Figure 55 - Figure 57)</b></td><td><b>Test Mode (Figure 58)</b></td></tr>
<tr><td><b>Initial Neural Parameters Eq.(7-6)</b></td><td><b>Adapted Weights</b></td><td><b><i>J</i></b>
$\int[e(t)]^2\,dt$</td></tr>
<tr><td>$\tau(0)=T_1(0)=T_f(0)=25$</td><td>$\tau=14.722,\quad T_1=11.236$
$T_f=6.069$</td><td>2.5×10^{-5}</td></tr>
</table>

Table 2: Adapted weights of TmD$_2$-DNU with linear somatic operation $\phi(\,)$, Eq.(4-21), (Figure 29) for time of adaptation of 5000 seconds, constant learning rates, Eq.(7-6) and quadratic integral error criteria *J* evaluated for input signal *u*(*t*) (Figure 58) of system in Eq.(7-5) .

Agreement in Response of Adapted TmD$_2$-DNU and the Approximated Plant

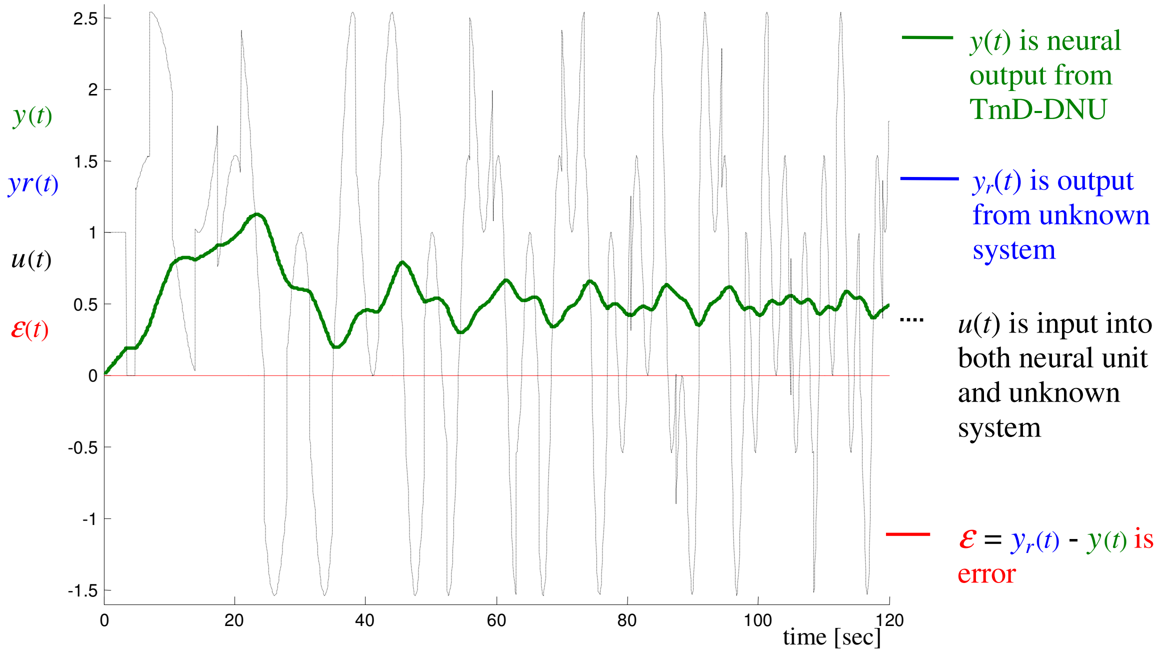


Figure 58: The response of adapted TmD$_2$-DNU, Eq.(4-21), (Figure 29) adapted to system Eq.(7-5) with input signal *u*(*t*) and initial neural weights $\tau(0)=T_1(0)=T_f(0)=25$ and further settings as in Eq.(7-6).

7.1.5 Approximating Capabilities of Time Delay Neural Units– Type 2 (TmD$_2$-DNU)

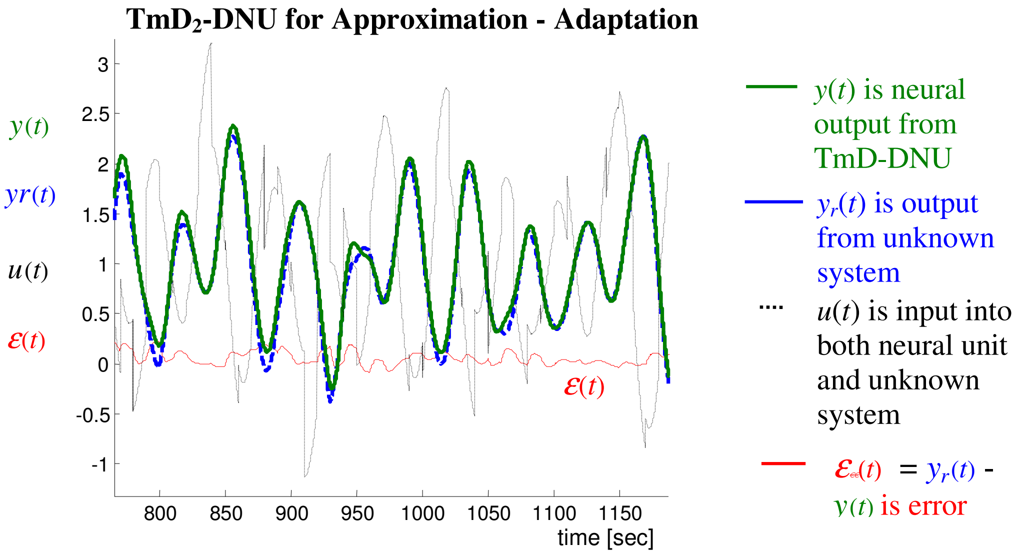


Figure 59: Example of detail of adaptation of TmD$_2$-DNU with linear output function $\phi(\;)$. The neural unit performs approximation of a dynamic system Eq.(7-8) with both time-delayed both input and state variable (Table 3, case B).

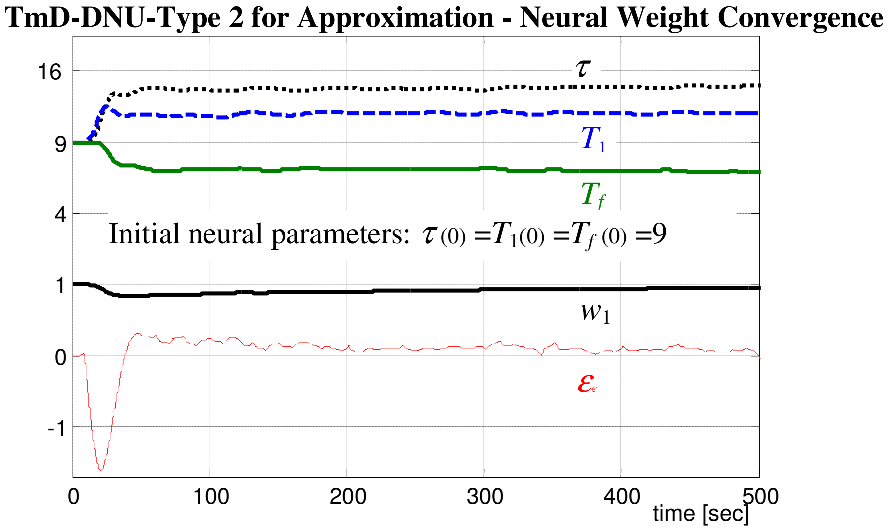


Figure 60: Example of convergence of neural weights of TmD$_2$-DNU for approximation of system of 10th order (Eq.(7-8)). Neural weights T_1 and T_f represent the continually adaptable parameters of time-delays (see Table 3, case C).

In this experimental part, the ‘real’ system to be approximated is chosen as a linear plant of the 10th order with the transfer function

$$G(s)=\frac{1}{(2s+1)^{10}}, \qquad (7\text{-}8)$$

where s is the Laplace operator. In fact, the system in Eq.(7-8) has been selected because it can be approximated in both time and frequency domains by systems in Eq.(7-5) and in Figure 28 and Figure 29 [45] [46].

Approximation of TmD-DNU–TYPE 2 - Adaptation Accuracy			
Structure of TmD$_2$-DNU :	$(\tau+0.1)\frac{d\xi(t)}{dt}+\xi(t-T_f)=w_1\,u(t-T_1)$ **(4-21)**		
To be adapted to:	$G(s)=\frac{1}{(2\cdot s+1)^{10}}$ **(7-8)**		
Adaptation Mode **(5000 seconds)**			**TEST MODE**
Case	**Initial Neural Weights** $w_1(0)=0.9$	**Adapted Weights**	$\boldsymbol{J} =$ $\int[e(t)]^2\,dt$
A	$\tau(0)=\ T_1(0)=T_f(0)=25$	$\tau=17.81,\ T_1=9.57$ $T_f=9.70$	**0.871**
B	$\tau(0)=\ T_1(0)=T_f(0)=16$	$\tau=16.30,\ T_1=10.17$ $T_f=7.85$	**0.566**
C	$\tau(0)=\ T_1(0)=T_f(0)=9$	$\tau=14.99,\ T_1=11.49$ $T_f=6.49$	**0.582**
D	$\tau(0)=\ T_1(0)=T_f(0)=4$	$\tau=10.08,\ T_1=13.31$ $T_f=2.10$	**3.16**

Table 3: Adapted weights of TmD$_2$-DNU, Eq.(4-21), for various initial conditions, time of adaptation of 5000 seconds, constant learning rates, Eq.(7-6), and quadratic integral error criteria J evaluated for response to input signal of system Eq.(7-5) .

It shall be emphasized that for all four sets of initial conditions depicted in two left columns of Table 2, all error plots converged toward zero, and all neural weights converged steadily to constant values for the constant learning rates in Eq.(7-6).

The following table compares the results of adaptation in order to choose the most appropriate result. The unit with adapted weights from Table 3 was tested for three distinct sets of adapted weights, shown in Table 4. These weights were obtained by adaptation from various sets of initial conditions (Table 3). Apparently, the single input TmD-DNU–Type 2 adapted from initial conditions C or B (Table 3 and Table 4) provides the most appropriate approximations of system Eq.(7-8) (at least for the currently used learning parameters and applied inputs).

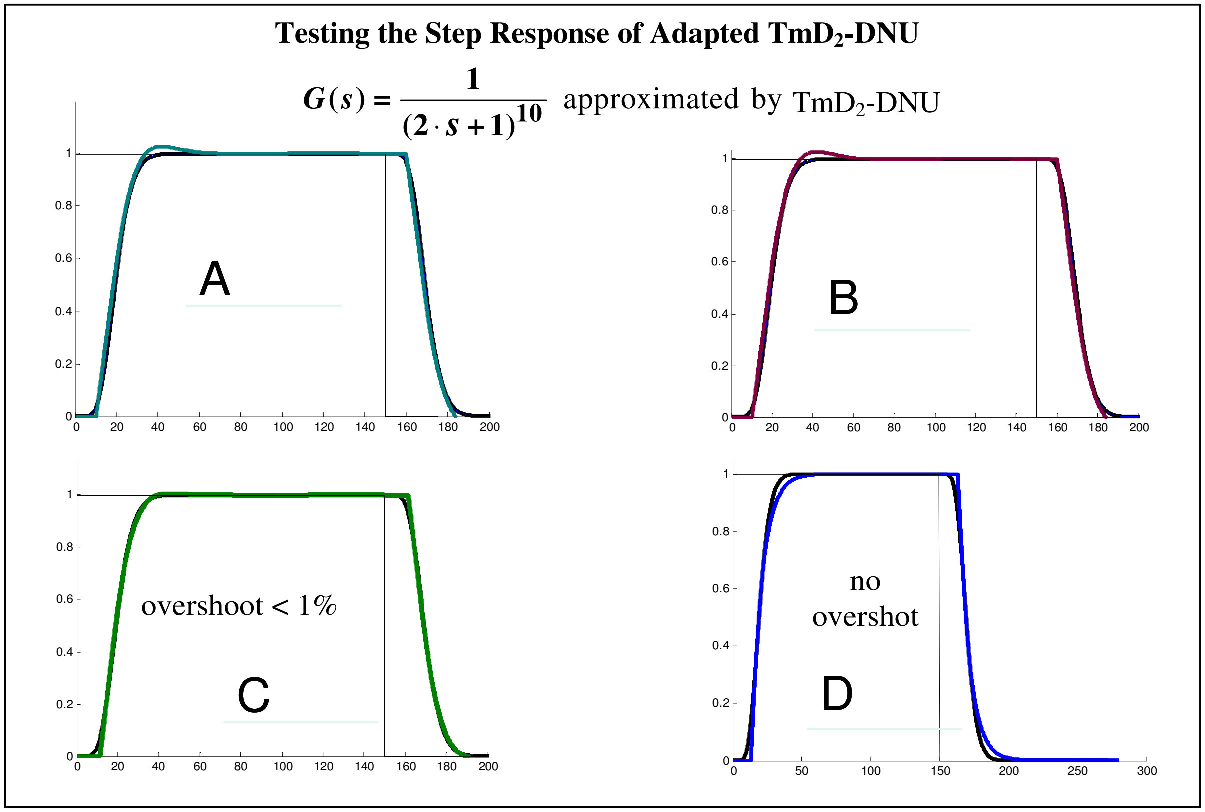


Figure 61: Testing of the adapted TmD$_2$-DNU as it fits the system in Eq.(7-8); the agreement in a step response is shown.

TEST MODE: Values of Quadratic Integral Error Criteria $\boldsymbol{J} = \int [e(t)]^2 \, dt$	Values of ***J*** for adapted TmD-DNU–Type 2 (Table 2)			
	(7-5) p. 86	A	B	C
square input (step response)	0.0349	0.2711	0.0489	0.0233
sinus chirp signal input (increasing frequency), time range = 360 sec, ω= <0.01, 1> [rad/sec]	0.3398	0.3449	0.2674	0.2870
Quasiperiodic sinus wave + square wave input	1.05	1.63	0.79	0.72

Table 4: Testing of the adapted TmD$_2$-DNU as it fits the system in Eq.(7-8). The values of *J* for neural unit adapted from various sets of initial conditions (Table 3) are compared also with *J* generated by system given in Eq.(7-5).

7.1.6 Conclusions on Linear TmD-DNU

Two types of linear (continuous-time) time delay dynamic neural units including adaptable time delays have been proposed in this work. The units are denoted TmD-DNU - Type 1 and TmD-DNU - Type 2 (TmD$_1$-DNU and TmD$_2$-DNU, respectively). Their stand-alone applications with a simplified linear neural output (somatic) operation have been the focus of this work in order to demonstrate their abilities to identify time delays in dynamic systems or to approximate dynamic systems with dynamics of higher orders.

TmD$_1$-DNU has been shown capable of both identifying time delays within linear plants and approximating higher-order systems through the input delays.

TmD$_2$-DNU has been introduced as an extension of the unit dynamic structure where another adaptable time delay is introduced into the state variable of TmD-DNU; therefore, the approximation capabilities of TmD-DNU are enhanced.

The disadvantage of using TmD-DNU for identification of time delays and system approximation is its (relatively) long time of adaptation which can be, however, significantly reduced by choosing another set of initial conditions or by re-running the adaptation in multiple epochs. In parallel, the problem of weight convergence toward local minima of error function is naturally reduced due to the robust approximating capability of TmD$_2$-DNU and can be further eliminated by choosing another set of initial weights for adaptation. According to our experiments, the basin of initial neural weights that make the units converge to an appropriate degree of accuracy, i.e., to some local minimum which provides the approximation with a sufficient degree of accuracy, is practically large enough (Table 3, Table 4). It is also necessary to find appropriate learning rates for which the learning algorithm is still stable for a particular problem; nevertheless, this was not difficult to overcome, and a simple approach for finding such values has been mentioned.

Amongst the major advantages of TmD-DNU is the capability to identify time delays within linear dynamic systems and its robust ability to approximate linear dynamic systems with dynamics of higher orders (e.g., the approximation of the 10th order is shown) in a continuous-time domain. Further, both the attractive simplicity of the learning algorithm and the excellent stability during adaptation due to the proposed design of neural structures (Figure 28 or Eq.(4-21)) are also advantages of using TmD-DNU.

7.2 APPLICATION OF HONNU TO COMMON ENGINEERING PROBLEMS

7.2.1 Identification of Technical Systems using HONNU

A special subset of HONNU, extended by pragmatically introduced exponential nonlinearity (shown in Eq.(7-9)), was applied to identification of the dynamics of parallel manipulator in Figure 62, [47] [54] (Model provided courtesy of the Department of Mechanics, U12105, FS ČVUT, Grant # VZ MSM 212200008). For example, the following equation is the structure of HONNU adapted to the dynamics of leg a)

$$y'' \approx f_a(\mathbf{x},\mathbf{W}) = w_{a_0}u_1 + \sum_{i=1}^{3} w_{ai}\,y_i + \sum_{i=1}^{3} w_{ai}\,y'_i + (u_2 - w_{a_7})\sum_{i=1}^{3}\sum_{j=i}^{3} w_{aij}\,\frac{y'_i y'_j}{e^{(-wae_{ij}(y_i - y_j)^2)}}, \qquad (7\text{-}9)$$

where y_1 is the position of the leg a) (Figure 62), which is generated as state variable inside the unit, thus establishing the dynamic nature of the unit; wa_i, wa_{ij}, and wae_{ij} are adaptable neural parameters (neural weights and internal dynamic parameters).

A customized structure of the units has been designed to benefit from oth the stability of static HONNU and the accuracy of the dynamic HONNU; internal state variables (leg position and velocity) were generated inside each neural unit itself while the state variables of the other legs entered neural units as external inputs acquired from the identified system .

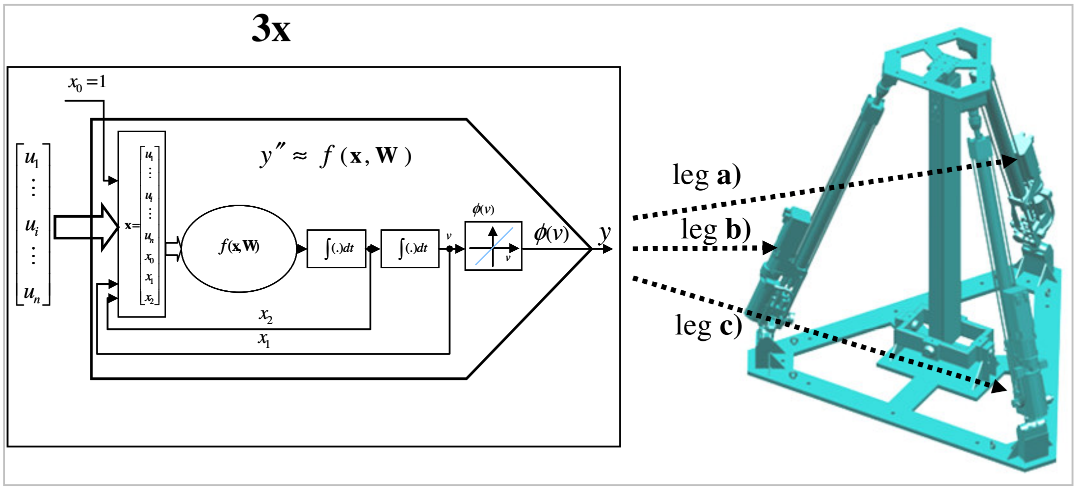


Figure 62: Three modified stand-alone HONNU were applied to the identification of the tripod leg dynamics (manipulator model courtesy of the Department of Mechanics, U12105, FS ČVUT [47]).

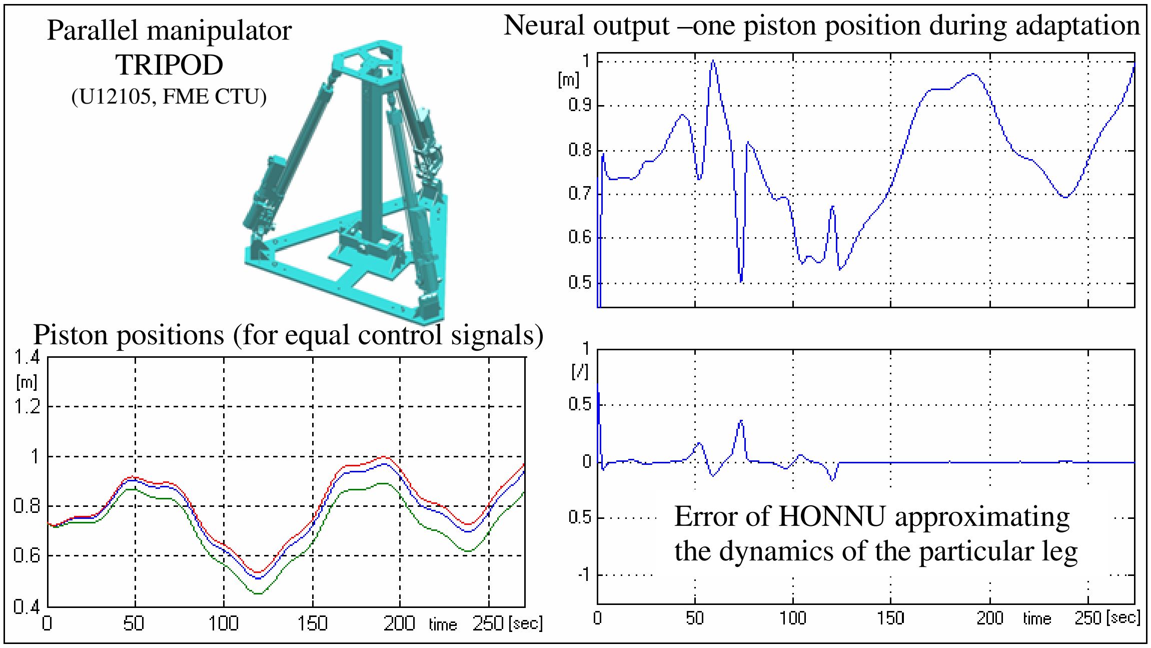


Figure 63: Simulation results of identification of dynamics of one tripod manipulator leg by hybrid dynamic HONNU (manipulator model by courtesy of the Department of Mechanics, U12105, FS ČVUT [47])

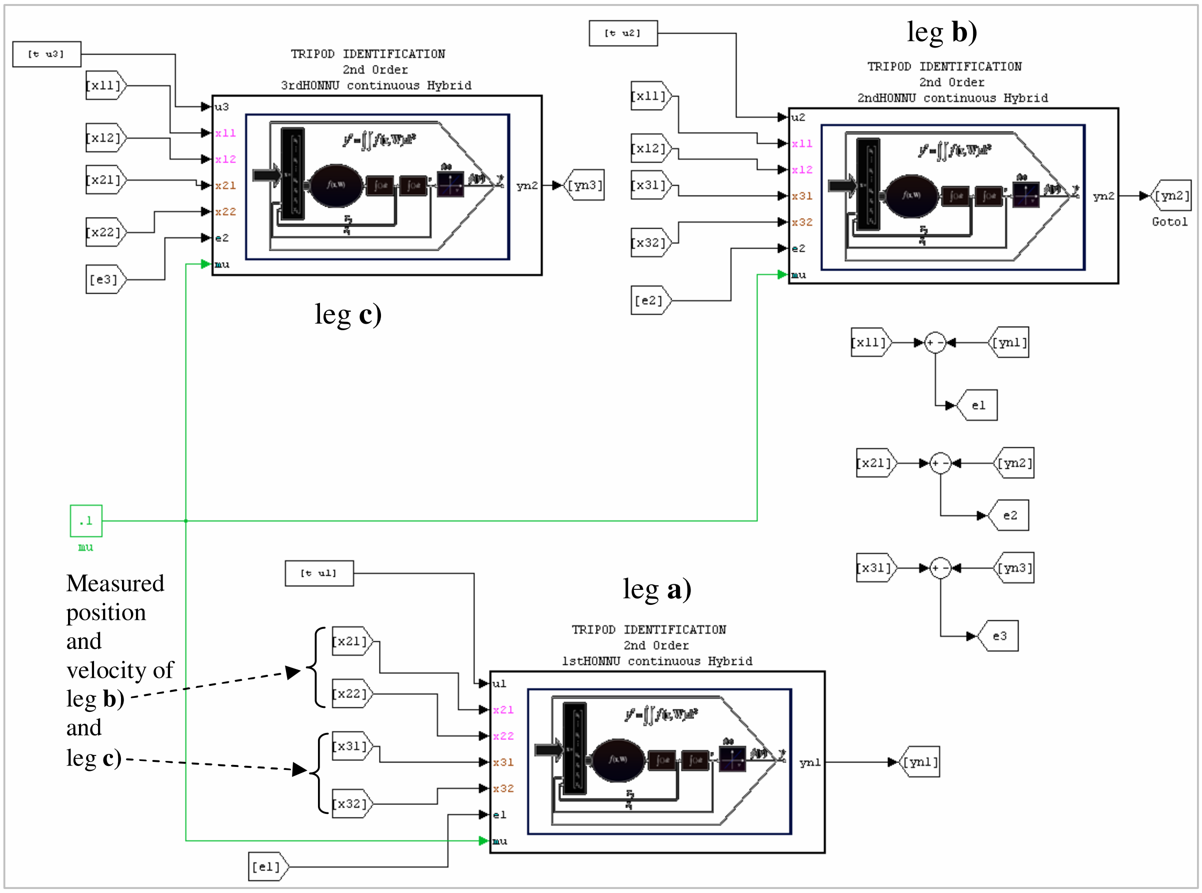


Figure 64: HONNU for identification of the dynamics of the parallel manipulator; neural inputs are the measured state variables (available position and velocity of the legs) except for the each unit's state variables, the neural outputs represents positions of the legs identified by HONNU.

7.2.2 State-Feedback Control

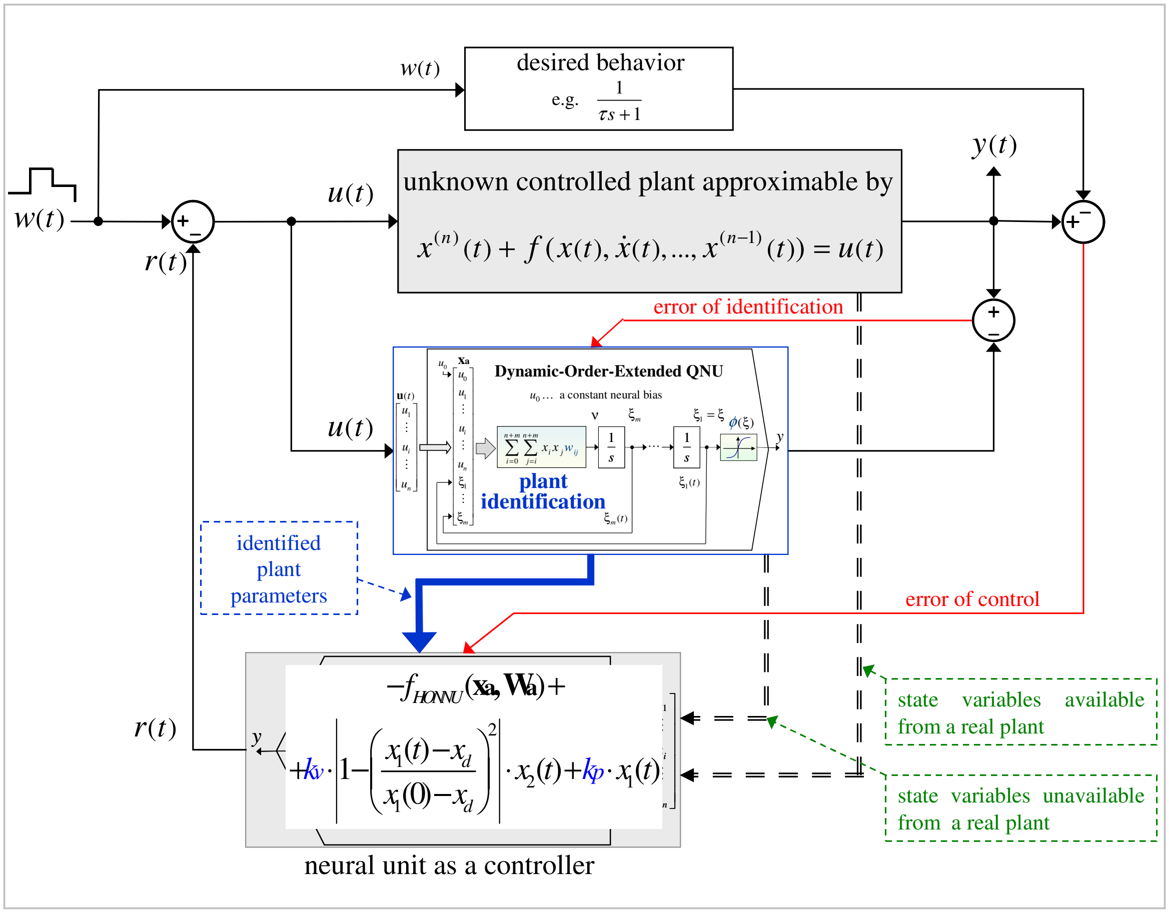
During development of HONNU, the unit's capabilities for control purposes were investigated and the structure of an adaptable neural state-feedback controller with variable damping was designed (Bukovsky, Redlapalli, Gupta, 2003 [52]). The purpose of the nonlinear state-feedback controller in Eq.(7-10) is to achieve a faster step response to square inputs by reducing the damping to a low value (initially zero) and gradually increase it to a positive optimum value such that the system will not overshoot the desired position.

$$r(t) = -f_{HONNU}(\mathbf{xa}, \mathbf{Wa}) + kv \cdot \left| 1 - \left(\frac{x_1(t) - x_d}{x_1(0) - x_d} \right)^2 \right| \cdot x_2(t) + kp \cdot x_1(t) \qquad (7\text{-}10)$$

$[x_1, x_2]^T \subset \mathbf{xa}$ and where $x_2 = \dot{x}_1$,

where $r(t)$ is output from the state-feedback controller, $f_{HONNU}(\mathbf{xa},\mathbf{Wa})$ represents the HONNU-identified dynamics of the controlled plant, x_1 is the actual position, x_2 is the velocity (second state variable), x_d is the desired value, kv and kp are adaptable neural parameters.

The structure of the fast neural nonlinear state-feedback controller, which uses a modified subset of CNU for parallel identification of an unknown, nonlinear plant with varying parameter values and structure, was simulated with promising results (Figure 67). In principle, the controller is designed to approximate (identify) any linear or nonlinear plant feauring not only the coefficients of a governing equation, but also an unknown structure that can vary during the control process. For the dynamic systems of a second order, e.g., the controller (Eq.(7-10)) performed about three times faster control than a common linear state feedback controller without overshoot and steady state error (Figure 66).



w(t)
desired behavior
e.g. $\frac{1}{\tau s+1}$
y(t)
u(t)
unknown controlled plant approximable by
$x^{(n)}(t)+f(x(t),\dot{x}(t),...,x^{(n-1)}(t))=u(t)$
w(t)
r(t)
error of identification
Dynamic-Order-Extended QNU
u_0 ... a constant neural bias
u(t)
plant identification
identified plant parameters
error of control
r(t)
$-f_{HONNU}(\mathbf{xa},\mathbf{Wa})+$
$+kv\cdot\left|1-\left(\frac{x_1(t)-x_d}{x_1(0)-x_d}\right)^2\right|\cdot x_2(t)+kp\cdot x_1(t)$
state variables available from a real plant
state variables unavailable from a real plant
neural unit as a controller

Figure 65: Simplified schema of architecture of the state-feedback neural controller using dynamic HONNU or TmD-DNU as system identifier and static HONNU as state-feedback controller according to Eq.(7-10).

The unknown controlled plant was represented by system

$$\ddot{x}(t)+0.1\cdot\dot{x}(t)+2\cdot x(t)+1.5\cdot x^3(t)=u(t). \tag{7-11}$$

Initial weights for system identification were set as [-10,-10,-10], and the learning rate for identification was chosen as μ=0.01. The learning rate for the neural control mode was set much lower than the learning rate for identification (μ=0.00033). The results shown in Figure 67 come from simulations conducted in two distinct operating modes. In the first mode, the subset of CNU identified the plant and the structure switched into the second (control) mode using the identified system parameters. It is advisable to analyze the plant to design an appropriate subset of CNU or QNU, to modify CNU or QNU by other nonlinear terms, to fit better the plant dynamics, thereby simplify the neural units and accelerate learning.

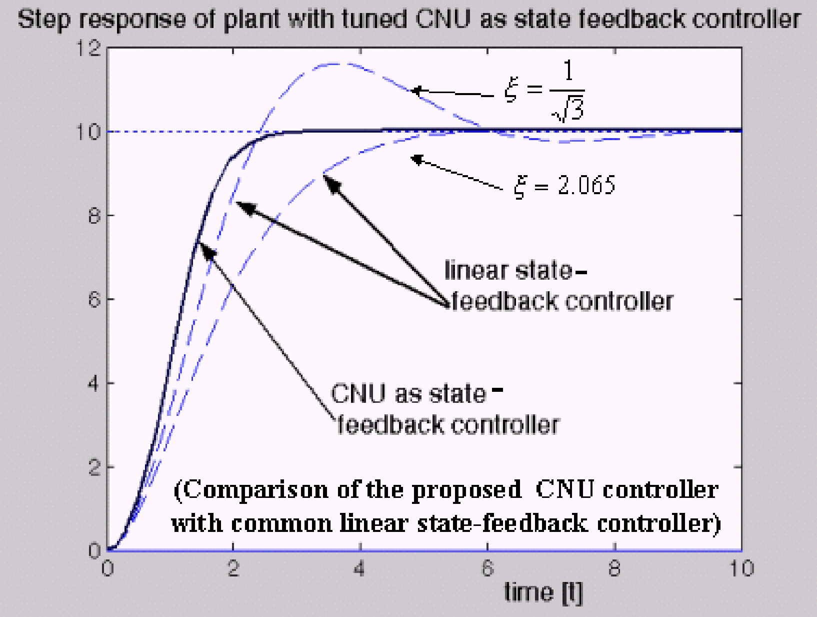


Figure 66: Step response of nonlinear neural controller with variable damping as in Eq.(7-10) compared to performance of linear state-feedback controller ($\xi=\alpha/\omega$, where $s_{1,2}= |\alpha|\pm j\omega$ is a complex conjugated root of the 2nd order linear control loop).

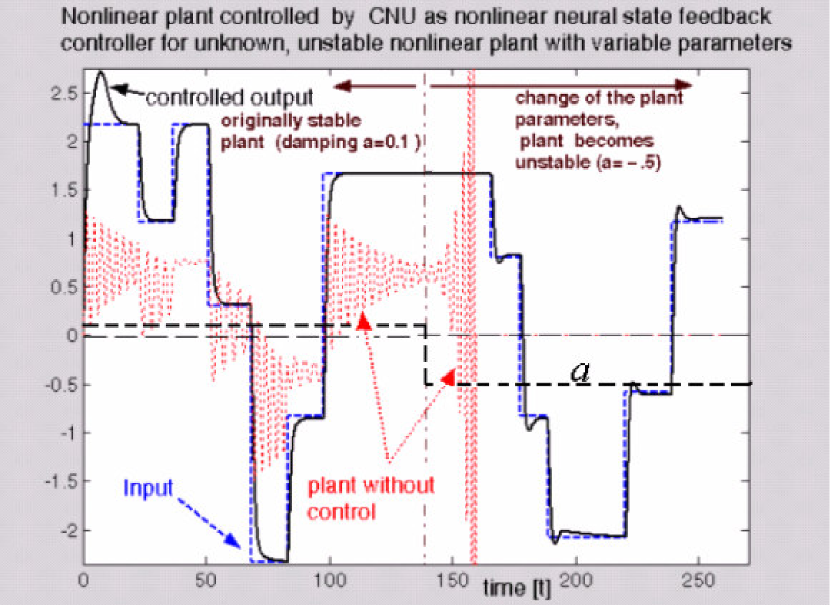


Figure 67: The simulation results for adaptive control of a nonlinear dynamic plant in Eq.(7-11) with sudden perturbation of the plant damping parameter *a*.

7.3 ADAPTIVE BEAT-BY-BEAT MONITORING OF CHANGES IN VARIABILITY USING HRV-HONNU

As with the first example, evaluation of changes in the level of chaos in logistic equations by the proposed method (Figure 39) is shown below. Following the technique in Figure 39, it would be unrealistic and merely theoretical if the same structure of a unit were used as the data generating system. To make this example more realistic and to demonstrate the universal applicability of HRV-HONNU (Chapter 6, Figure 43, Figure 44), the mathematical structure of the HRV-HONNU as the evaluating neural unit is used; it is different from the mathematical structure of the data generating system, i.e., the logistic equation

$$y(k+1) = a \cdot y(k) \cdot \left(1 - y(k)\right), \tag{7-12}$$

where a is the bifurcation parameter. The mathematical structure of the dynamic HRV-HONNU (Figure 44) that detects the changes in the level of chaos in (7-12) is

$$y_{HRV-HONNU}(k+1) = \sum_{i=0}^{n+2} \sum_{j=i}^{n+2} xa_i(k) xa_j(k) w_{ij}$$
$$\mathbf{xa}(k) = \begin{bmatrix} xa_0(k) \\ xa_1(k) \\ \dots \\ xa_{n+2}(k) \end{bmatrix} = \begin{bmatrix} 1 \\ u_1(k) = \cos(\omega_1 t + \varphi_1) \\ u_2(k) = \cos(\omega_2 t + \varphi_2) \\ xa_3(k) = y_{HRV-HONNU}(k) \\ \vdots \\ xa_{n+2}(k) = y_{HRV-HONNU}(k-n+1) \end{bmatrix}, \tag{7-13}$$

where w_{ij}, ω_1, ω_2, φ_1 and φ_2 are adaptable neural parameters, inputs u_1 and u_2 represents adaptable input signal preprocessor that increase the approximation capability of HONNU. Of course, the preprocessor does not have to be used with the logistic equation here; however, this example demonstrates the capability of HRV-HONNU to detect changes in variability even if the mathematical structure of the data generating system in Eq.(7-12) is simpler than the detecting neural unit itself (Eq.(7-13)).

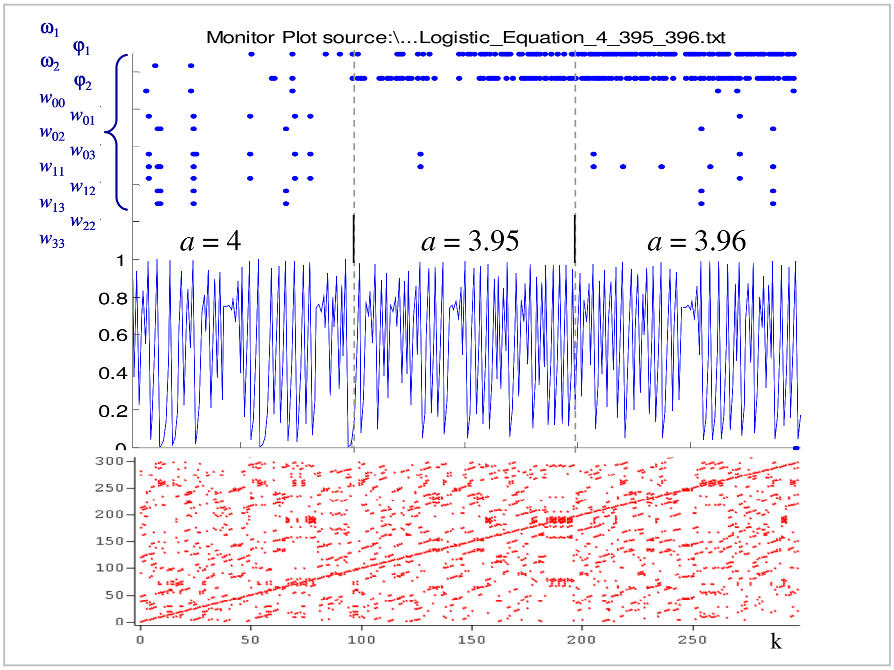


Figure 68: Detection of changes in dynamics of logistic equation in chaotic mode. The similar density of blue dots indicates intervals of similar chaos within the time series; the recurrence plot is in red.

As mentioned earlier, it might be necessary to use the adaptive signal input preprocessor (such as that in HRV-HONNU) if the mathematical structure of the data-generating system is unknown or too complicated and when error and weights of pure HONNU do not converge with an appropriate degree of accuracy.

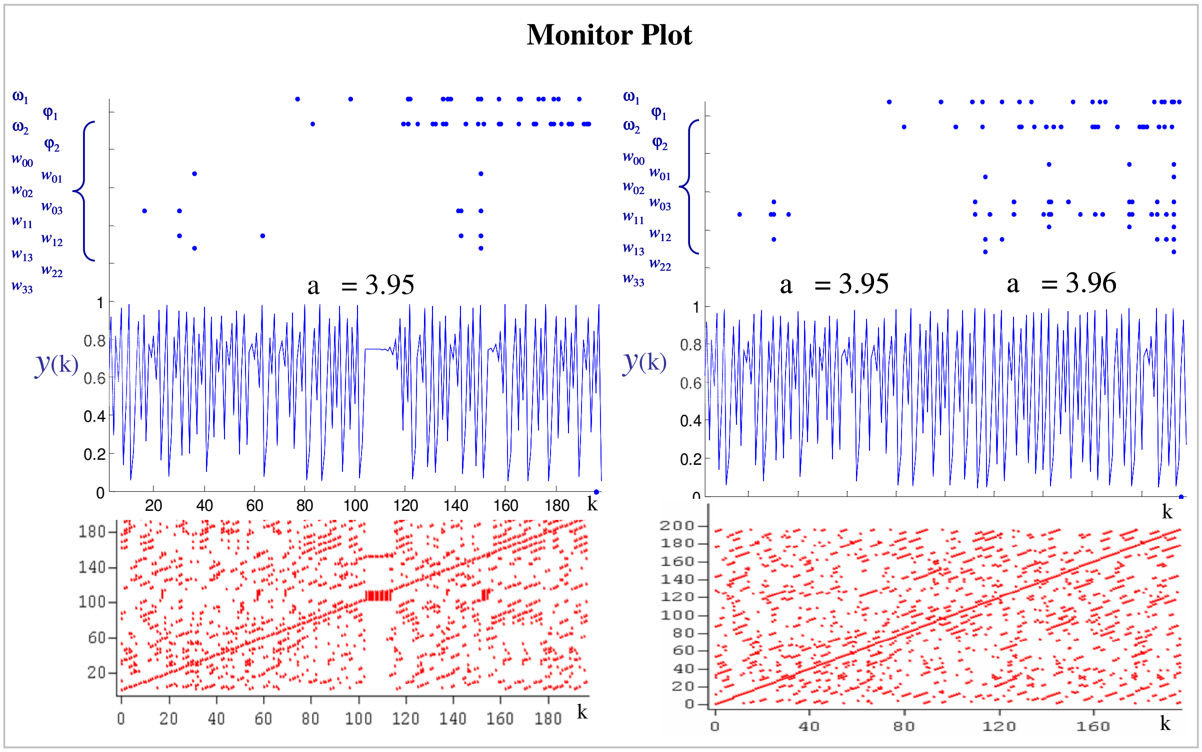
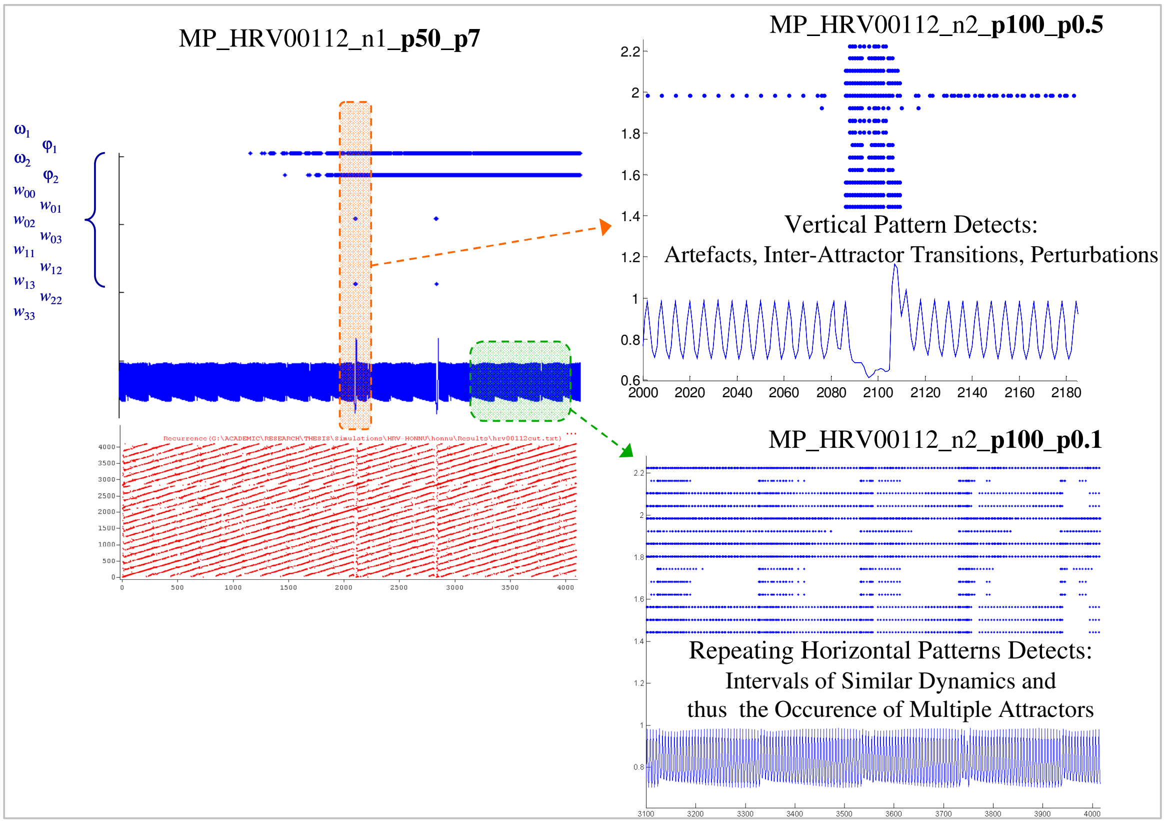


Figure 69: Detection of dynamic changes in the time-series generated by logistic equation with a small sudden increase of the bifurcation parameter *a* (on the right). The increased density of blue dots in top right corner indicates a change in the dynamics of the time series; the recurrence plot is shown in red.

Further examples show the application of the adaptive method to the evaluation of simulated heart beat tachograms (Section 3.2 in this work, [18] to [23]) with various levels of deterministic chaos from periodic to highly chaotic, possibly due to multi-attractor behavior where correlation dimension and LLE could not be reliably evaluated [23] . The data evaluated below are highly chaotic time series generated by deterministic systems with constant parameters, i.e., generated by a simplified model of the fast beat-by-beat control influences of the autonomous nervous system on cardiovascular dynamics ([18] to [23]). The generated time series feature so high a level of chaos that the correlation exponent saturated only with eleven time series of 63 generated and evaluated by the correlation dimension by the Grassberger-Proccacia algorithm [3] [23] (see section 3.2). The following results reveal intervals of similar variability (chaos), i.e., possible single-attractor intervals. The common nonlinear methods (CD, LLE) are well suited for the evaluation of signals up to a certain degree of complexity (variability), where multi-attractor behavior and excessive changes in dynamic behavior diminish the reliability of the invariant evaluation (section 2.1 in this thesis, [26]). The sensitive allocation of the intervals of similar dynamics by HRV-HONNU, e.g., supported by the Recurrence Plot, can also improve the reliability of more generally known methods.



MP_HRV00112_n1_**p50_p7**

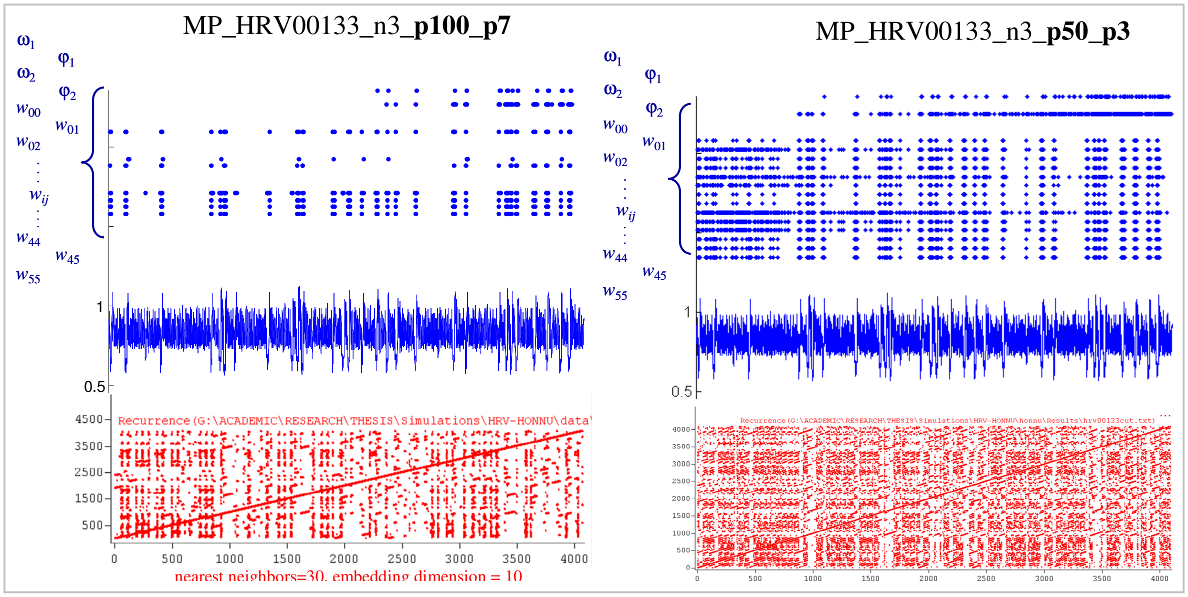
MP_HRV00112_n2_**p100_p0.5**

Vertical Pattern Detects:
Artefacts, Inter-Attractor Transitions, Perturbations

MP_HRV00112_n2_**p100_p0.1**

Repeating Horizontal Patterns Detects:
Intervals of Similar Dynamics and
thus the Occurence of Multiple Attractors

Figure 70: Monitor plot of simulated heart beat tachogram of lower variability than in Figure 71 (LLE=0.46, Figure 10, Appendix Appendix-3, [18] to [23])



MP_HRV00133_n3_**p100_p7**

MP_HRV00133_n3_**p50_p3**

Figure 71: Monitor plot of simulated heart beat tachogram of higher variability than in Figure 70 (LLE=0.91, Appendix Appendix-3, [18] to [23]); both monitor plot (blue) and recurrence plot (red) indicate similar intervals of similar variability in a signal.

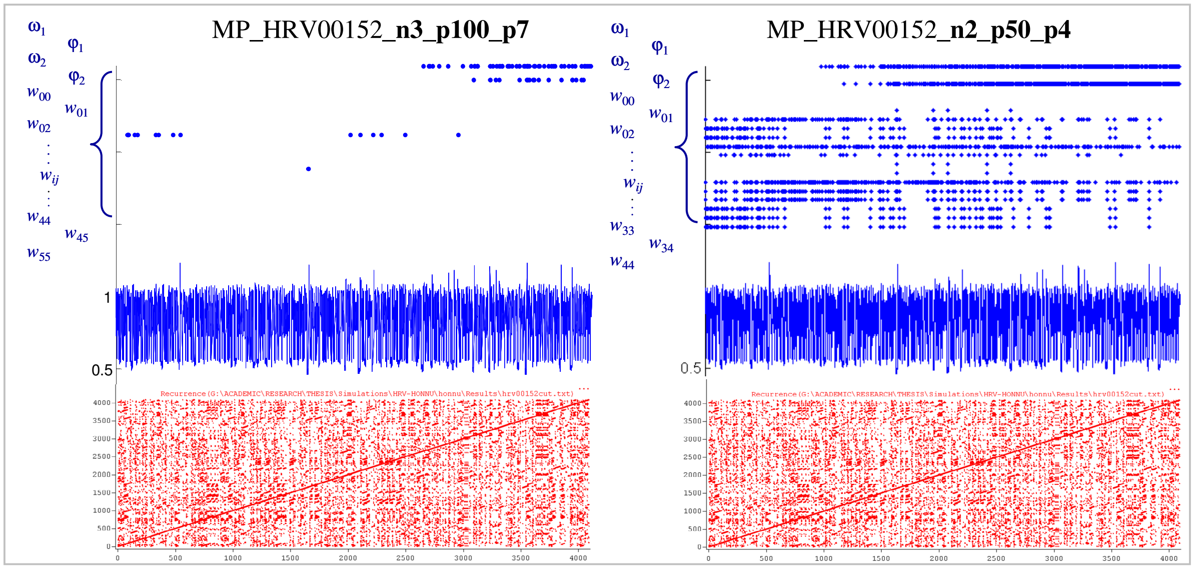


Figure 72: Monitor plot of simulated heart beat tachogram with high level of chaos than in Figure 70 and Figure 71(LLE=1.22, Figure 10, Appendix Appendix-3, [18] to [23]); the monitor plot (blue) indicates the intervals of similar variability in a signal more clearly than the recurrence plot (red).

The results on simulated chaotic heart beat tachograms shown in monitor plots in Figure 70, Figure 71, and Figure 72, show the capability of HRV-HONNU to detect singularities (such as inter-attractor transitions, perturbations, artefacts) or intervals of a similar level of chaos (single-attractor intervals) in a signal. The detection sensitivity of HRV-HONNU is scalable due to adjustable sensitivity detection parameters generally denoted as *p* (Figure 40). For highly chaotic signals, where the recurrence plot (red) may not be sensitive enough to clearly indicate the regions of similar variability and their borders (such as in Figure 69 or Figure 72), the HRV-HONNU clearly indicates the important changes in the level of chaos due to its adaptive approximation of the system dynamics hidden in the highly chaotic signal.

The simulated time series evaluated and shown from Figure 70 to Figure 72 do not include any internal or external system perturbations or any introduced artefacts; the simulated time series displayed so high level of chaos (regarding the 4096-heartbeat length) that evaluation of correlation dimension did not converge. The correlation exponent did not saturate with increasing embedding dimension. HRV-HONNU reflects the changes in signal variability even for recordings significantly shorter than the minimum signal length necessary for reliable evaluation by CD or LLE of highly chaotic time series (Figure 68, Figure 69).

Below, the capability of HRV-HONNU to detect intervals of a similar level of heart rate variability is demonstrated on real R-R diagrams measured on patients suffering from cardiac arrhythmias [17] (namely the MIT-BIH arrhythmia database).

First, the R-R diagram of a cardiac patient with significant changes in variability is shown in Figure 73 and Figure 74. Then, the results of HRV-HONNU are shown for other patients with less remarkable changes in variability (more stationary R-R diagrams) in Figure 75 and Figure 76.

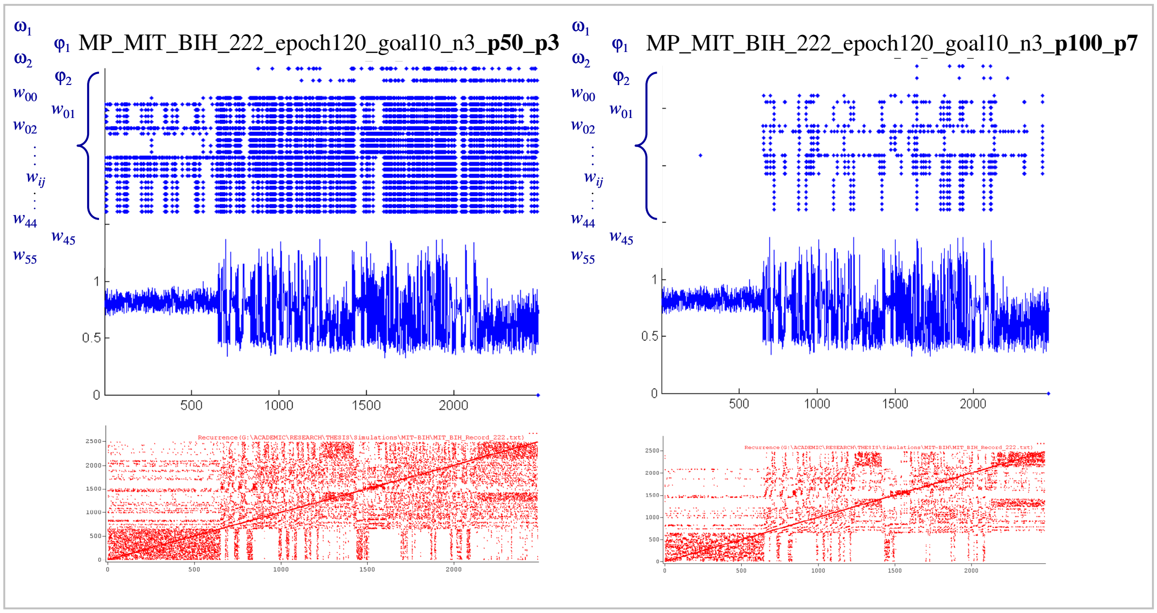


Figure 73: The monitor plot of variability changes and similar variability intervals of real R-R diagram of a cardiac patient (MIT-BIH, record # 222, female, age 84).

MIT-BIH Record 222 (MLII, V1; female, age 84)	
Sample #:	**Points of interest:**
20	6:45 Normal sinus rhythm
1052	17:32 Atrial fibrillation
1188	19:48 Atrial couplet
1207	**20:07 Atrial bigeminy**
1352	22:32 Paroxysmal atrial flutter, nodal rhythm
1483	24:43 Noise
1543	25:43 End of atrial flutter, nodal rhythm, normal
1569	26:09 Paroxysmal atrial flutter, nodal rhythm

Table 5: Table of medical annotations of ECG signal corresponding to R-R diagram in Figure 73.

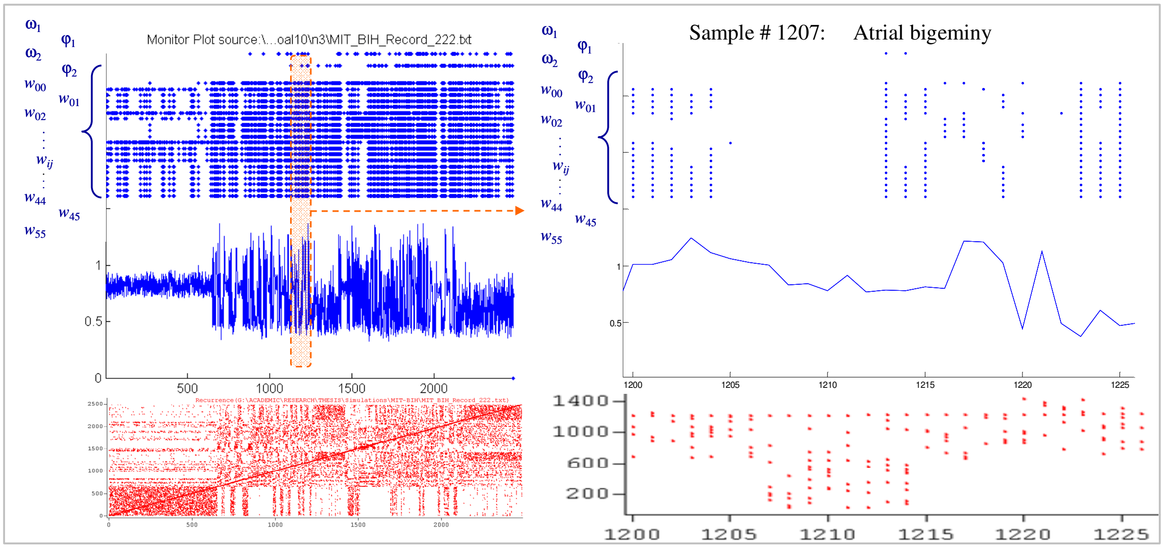


Figure 74: Detail of changes in variability during atrial bigeminy (sample # 1207) of the patient from Figure 73 and Table 5.

Record 203 (MLII, V1; male, age 43)	
Sample #:	**Points of interest:**
12.5	5:00 Ventricular tachycardia, 4 beats and 9 beats
794	**13:14 Atrial fibrillation, ventricular couplets**
902	15:02 Noise
1322	22:02 Ventricular couplet, PVCs
1405	23:25 Noise
1444	24:04 PVCs
1486	24:46 Noise
1599	**26:39 Ventricular tachycardia, 7 beats**
1611	26:51 Ventricular couplet, PVCs
1635	27:15 Ventricular tachycardia, 3 beats

Table 6: Table of medical annotations of ECG signal corresponding to another R-R diagram in Figure 75 and Figure 76.

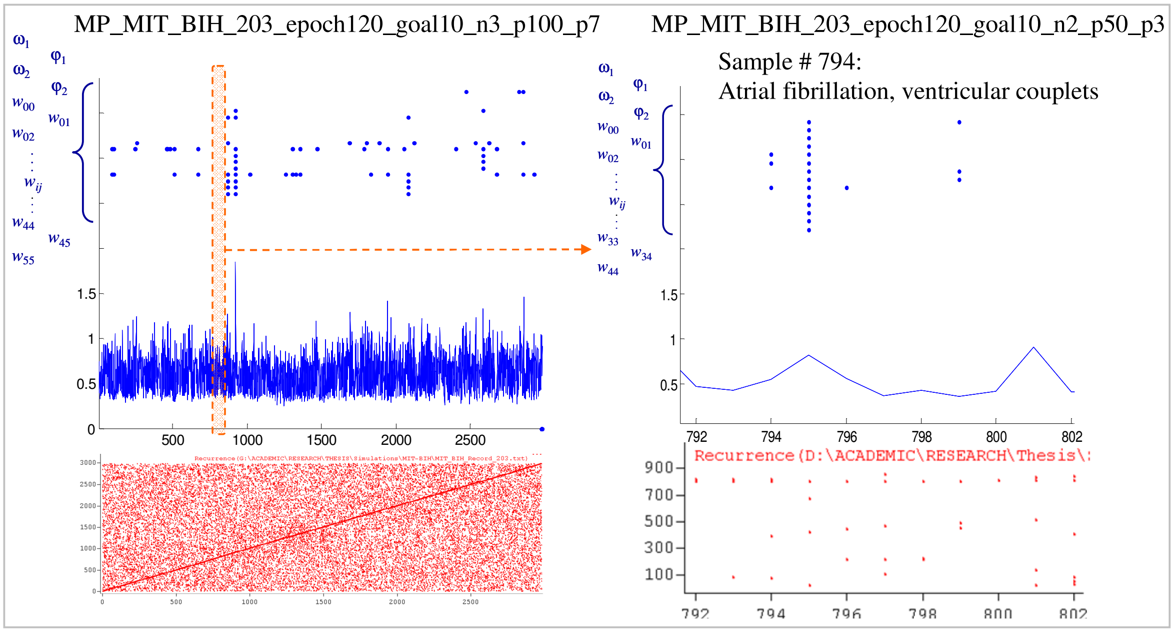


Figure 75: Detail of detection of sudden variability change at the start of atrial fibrillation and ventricular couplets (sample # 794) (MIT-BIH, record 203, male, age 43).

The results shown from Figure 73 to Figure 75 demonstrate the capability of HRV-HONNU to detect changes in variability as well as to allocate regions of similar variability within real R-R diagrams (heart beat tachograms).

The details showing beat-by-beat detection of changes in HRV during atrial bigeminy in Figure 74, atrial fibrillation in Figure 75, or ventricular tachycardia in Figure 76, do not mean that HRV-HONNU can already detect and determine these cardiac arrhytmias. These figures demonstrate that the arrhythmias were accompanied by changes in variability and these changes were clearly detected by HRV-HONNU in the monitor plot. The detection and determination of particular arrhythmias by the proposed adaptive methodology is a matter of further research and require the cooperation of specialists from relevant fields of medicine.

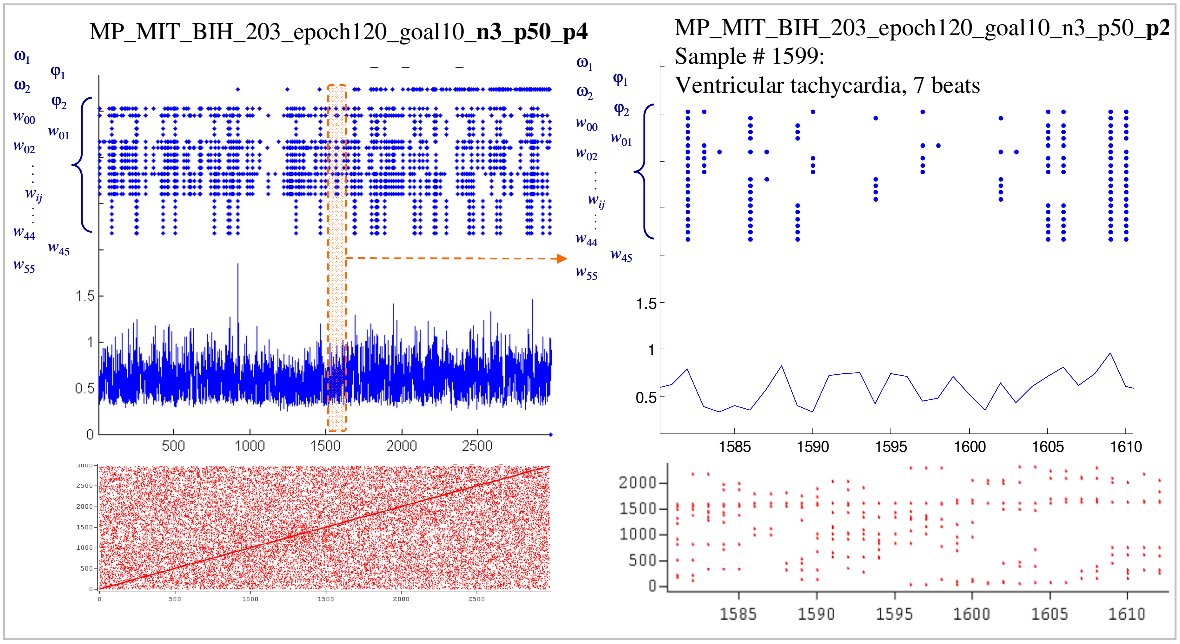


Figure 76: Detail of detection of variability changes during ventricular tachycardia (sample # 1599, 7 beats) (MIT-BIH, record 203, male, age 43).

The intrinsic strength of HRV-HONNU lies in its beat-by-beat monitoring of the variability increase or decrease (as shown in Figure 68 or Figure 69); therefore, the suitable application of HRV-HONNU to the evaluation of HRV should be based on this property. Typically, the level of heart rate variability reflects the amount of oxygen delivered to the brain of the fetus. Thus, HRV-HONNU shall perform adaptive monitoring of the decrease or increase of variability (i.e. oxygen) and thus it can help to lower the danger of a fetus choking. This possible application is a matter of future research on the adaptive evaluation of heart rate variability, which can also be generally called “beat-by-beat HRV monitoring”.

8 Conclusions and Further Research

8.1 SUMMARY OF ACHIEVEMENTS

According to the goals introduced at the beginning of this thesis (p. 5), the author considers the proposed goals of this thesis (p. 5) to be achieved.

The particular goals were achieved as follows:

1) A theory of novel neural architectures for the approximation (modeling) of complex dynamic systems is established in this thesis.

1.1) A theory of nonconventional higher-order nonlinear neural units was further developed in this thesis.

A theory of linear continuous time-delay dynamic neural units was established.

A theory of nonlinear continuous time-delay dynamic neural units was established.

A novel methodology for the approximation of complex systems using the developed neural units was founded.

First, the concept of discrete and continuous and static and dynamic higher-order nonlinear neural units (HONNU) was introduced and further developed in this thesis (Sections 4.1, 4.2, and 4.4). These units have nonlinear (polynomial or other customizable nonlinearity) aggregating functions of neural inputs that aggregate also neural state variables in case of their dynamic modifications. The parallel between the nonlinear aggregation function of HONNU and real biological neurons was drawn, especially in Sections 4.1 and 4.2, with a focus on the higher computational capability expected from single biological neurons. During the research, HONNU has proven themselves to be universal and capable system approximators. The capacity for more-accurate approximation is one of the advantages of continuous time-dynamic HONNU over discrete HONNU. On the other hand, the advantage of discrete dynamic HONNU is its natural ability to approximate complex dynamic systems with simpler neural architecture and for its natural capability to develop chaotic behavior (section 3.1) even in one-dimensional state space, due to higher-order nonlinear aggregation function.

Second, the linear continuous time-delay dynamic neural units (TmD-DNU) were developed in Subsections 4.3.1, 4.3.2 and 4.4.2 and experimentally tested with very promising results on time-delay identification and system approximation (Section 7.1). These dynamic neural units are capable of identifying time delays in system inputs as well as in the state-feedback of a system (modeled by a neural unit); moreover, TmD-DNU are capable of approximating high-order dynamic systems by low-order dynamic architecture due to time delays introduced as adaptable neural parameters. The increased approximating capability due to the increased density of poles in the left part of the complex plain belongs among the advantages of TmD-DNU.

Third, the concept of nonlinear continuous time-delay dynamic higher-order neural units (TmD-DHONNU) was introduced in Subsections 4.3.3 and 4.4.4. These neural units benefit from the increased approximating capability of TmD-DNU (due to adaptable time delays in both neural inputs and state feedback) as well as from the universal approximating capability of HONNU (due to the nonlinear aggregation function). As universal approximators, TmD-DHONNU are potentially equipped with the highest approximation ability together with simplest neural architecture and the minimum number of neural parameters out of the three proposed classes of dynamic neural units, i.e., HONNU, TmD-DNU, TmD-DHONNU. During the simulation experiments, the proposed novel neural units converged and generated good and very promising results.

1.2) The learning algorithm for the adaptation of each of the classes of units has been derived in Section 4.4. All the units can be adapted by the dynamic modification of the gradient-based backpropagation learning algorithm.

1.3) All the neural architectures (Section 4.1 to 4.3), the learning algorithm (Section 4.4), and the stability-maintaining adaptation technique of system approximation discussed in Section 5 are naturally simple, universal, highly customizable, and can be learned and used by researchers of various fields and levels. To overcome the instability issues during the learning of the dynamic HONNU, the simple approximation technique combining static and dynamic neural units was proposed in Chapter 5.

The mathematical complexity of system approximation (control, monitoring) using the proposed novel neural architectures is crucially lowered by the stand-alone implementation of the neural units.

As an aside, for the purpose of system identification in Subsection 7.2.1, a simple hybrid network of static and dynamic HONNU with extended nonlinearity and extended dynamic order were designed and applied (Figure 62 to Figure 64). Each hybrid of static and dynamic HONNU in this network uses its dynamic structure to generate its own state variables while the other state variables are measured and introduced from the real system (if available). In this way, the hybrids of static and dynamic neural units pursue both the stable nature of static HONNU and approximation accuracy of continuous-time dynamic HONNU.

The applicability of the proposed algorithms and methodology demonstrating the achievement of goal # 1 is shown in examples of system approximation and identification of time delays in Section 7.1 and control applications in Section 7.2.

Further, achievements of goal # 2 (p.5) are discussed.

2) A novel methodology for adaptive monitoring of sudden as well as smooth changes of variability (level of chaos) in signals generated by complex dynamic systems was founded in Section 5 and developed in Section 6.

In particular, the achievements regarding the novel evaluation of complex systems are as follows:

2.1) A novel class of special neural units (HRV-HONNU), with an adaptable input-signal preprocessor, was designed as an adaptive neural tool approximating the complex dynamics and responding to the sudden and continuous changes of a system in a real time (Section 6).

Even a stand-alone implementation of the HRV-HONNU unit (Figure 43, Figure 44) is capable of approximating a highly chaotic time series due to the adaptable input signal preprocessor. Results on the approximation of a complex time series (heart beat tachograms) by HRV-HONNU displayed a sufficient degree of accuracy. Therefore, the mathematical simplicity, minimum number of neural parameters, and good stability of the dynamic HRV-HONNU during its adaptation due to the technique proposed in Section 5 are all in agreement with the fulfillments of goal # 1 discussed above.

2.2) A novel and universally applicable methodology for adaptive evaluation of signal variability was established based on the monitoring of the neural parameters of the proposed novel neural units.

Due to its adaptive nature, the proposed methodology of adaptive evaluation of variability of chaotic time series is capable of:

- detecting sudden changes caused by inter-attractor transitions, artefacts, internal or external perturbation into a system, noise,
- allocating intervals of similar dynamics, e.g., single attractor behavior, in the time series,
- reflecting the level of variability (complexity) of a signal in a particular region of a single attractor,
- revealing the multiple attractor behavior of a signal by detecting the repeating patterns of changes in the system dynamics.

A simple realization of the proposed methodology was tested on deterministic yet highly chaotic signals as well as on real heart beat tachograms for which common nonlinear methods, such as correlation dimension or LLE, do not provide reliable results, either because of too complex or time-varying dynamics, multi-attractor behavior, or a lack of the required number of samples.

The variability evaluation method is scalable, i.e., the sensitivity detection parameters (Figure 40) determine the minimum changes of variability that the unit detects; thus, the neural unit sensitivity can be scaled to both long-term or detailed short-term variability monitoring. The simulation experiments have shown the promising ability of the proposed method to detect even small changes in a high level of chaos in deterministic signals, and to indicate small changes in the nonlinear dynamics even where other methods, such as the recurrence plot, do not show clear results (Figure 69, Figure 68). In such cases, the proposed method using the developed "monitor plot" provides useful visualization of the results.

The achievements regarding goal # 2, i.e., the novel evaluation and monitoring of variability of chaotic time series including real R-R diagrams (heart beat tachograms) are demonstrated in Section 7.3.

8.2 LIMITATIONS AND CHALLENGES FOR FURTHER RESEARCH

Even though the developed nonconventional neural architectures HONNU, TmD-DNU, and TmD-DHONNU are very promising universal approximators of complex systems and work very promisingly for technical systems, their use does not outperform the proper derivation of a mathematical model of a complex system if such an analysis can be done. Even though application of the proposed neural units to systems that are difficult to analyse is believed to introduce considerable improvements, the combination of customization of the internal neural architecture together with the proper mathematical analysis of a system can:

- maximize accuracy of the neural units
- minimize the time of adaptation,
- and find initial neural parameters from which the unit would converge to a more appropriate minimum of error function.

The primary challenges of further research regarding the proposed neural units in common engineering problems are as follows:

- Investigation of the applicability of stand-alone HONNU, TmD-DNU, and TmD-DHONNU to system approximation and control of nonlinear systems where piece-wise-linearization control approaches are commonly used.
- Further investigation of a neural state-feedback controller in the regime of simultaneous system approximation and state-feedback control using HONNU, TmD-DNU, and TmD-DHONNU.
- Investigation of approximating capabilities of HONNU, TmD-DNU, and TmD-DHONNU with various types of nonlinear somatic operations $\phi(\,)$ with a focus on neural networks using these neural units HONNN – Higher-Order Nonlinear Neural Networks, TmD-DNN – Time Delay Dynamic Neural Networks.
- Further research of neural units with an adaptable signal input preprocessor for identification of unknown system input signals for the purpose of advanced monitoring of internal as well as external system perturbations.

Regarding diagnostics and current state of research, the utilization of the adaptive method is limited to (sensitive beat-by-beat) detection of intervals where variability increases or decreases, to detection of repeating patterns in time series, or to detection of various kinds of singularities in complex time series (artefacts, perturbations). Currently, the method itself does not indicate the type of cardiac arrhythmias (so far), nor has it been investigated whether it could indicate their incoming occurrence. However, it has been shown that if arrhythmias occurred, they were accompanied (preceded or followed) by changes in variability that were clearly detected by the proposed method. For example, the method can also function as a simple watchdog to detect unusual patterns or sudden changes in the heart beat of a patient. Once such changes are detected, physicians can focus on the specific recordings using other methods, view and analyze particular beats in ECG. Thus the method proposed today is potentially very useful in supporting other medical diagnosis methods.

The primary challenges of further research regarding the proposed theory and methodologies in biomedical engineering problems are as follows:

- Beat-by-beat HRV fetal monitoring is the very important topic of current research. The established theory and methodology enables sensitive monitoring of the variability increase or decrease. Thus, the level of oxygen delivered to the brain of a fetus can be monitored using the adaptive method and visualized in the “monitor plot”.
- Development of Type-2 HRV-HONNU for adaptive evaluation of HRV where the frequency component of the vagal nerve tonus would be due to the limit cycle of the dynamic neural unit, rather than caused by a periodic input within the input signal preprocessor (the lower number of neural parameters, more sensitive detection of changes in variability)
- Investigation of capabilities of HONNU to detect and distinguish between particular types of cardiac arrhythmias related to the scalability of the detection sensitivity of the proposed method.
- Investigation of multi-attractor dynamics in complex systems in general. The proposed methodology can introduce new knowledge in the field, e.g., the investigation of heart beat dynamics by HRV-HONNU of patients before, during, and after a cardiac surgery.

References

[1] Alligood, K., T., Sauer, T., D., Yorke, J. , A.: *Chaos*: *An Introduction to Dynamical Systems*, New York, Springer-Verlag, 1996.

[2] Horák J., Krlín L.: *Deterministic Chaos and Mathematical Models of Turbulence* (in Czech), Academia, Prague, Czech Republic, ISBN 80-200-0416-5, 1996.

[3] Grassberger, P., Procaccia, I.: "Characterisation of Strange Attractors", *Physical Review Letters* 65, 1983, pp.346.

[4] Takens, F.: "Detecting Strange Attractors in Turbulence", *Lecture Notes in Mathematics*, Springer-Verlag, 1981, pp.898.

[5] Persson, P., B., Wagner, C., D.: "General Principles of Chaotic Dynamics", *Cardiovascular Research*, 31, Review, Elsevier, 1996.

[6] Goldberger, A., L., Rigney, D., R.: "On the Nonlinear Motions of the Heart: Fractals, Chaos and Cardiac Dynamics, in Cell to Cell Signaling: From Experiments to Theoretical Models " *(A. Goldbeter, ed.),* Academic Press, San Diego, 1989, pp.541-550.

[7] Goldberger, A., L., Rigney, D., R.: "Sudden Death is not Chaos, in the Ubiquity of Chaos" *(S. Krasner, ed.*), AAAS Press, Washington, DC, 1990, pp.23-34.

[8] Costa, M., Pimentel, I., R., Santiago, T., Sarreira, P., Melo, J., Ducla-Soares, E.: "No Evidence of Chaos in The Heart Rate Variability of Normal and Cardiac Transplant Human Subjects", *Journal of Cardiovascular Electrophysiology, 10/*(10), 1999, pp.1350-1357.

[9] Bogaert, C., Beckers, F., Ramaekers, D., Aubert, A., E.: "Analysis of Heart Rate Variability with Correlation Dimension Method in a Normal Population and in Heart Transplant Patients", *Autonomic Neuroscience: Basic&Clinical*, 90/1-2, 2001, pp.142-147.

[10] Kristal-Boneh E, Raifel M, Froom P, Ribak J.: "Heart Rate Variability in Health and Disease". *Scan J Work Environ Health,* 1995; 21: pp.85-95.

[11] Kaplan, D., T., Cohen, R., J.: "Is Fibrillation Chaos?", *Circ. Res. 67*, 1990, pp. 886-892.

[12] Storella, R., J., Wood, H., W., Mills, K., M., Kanters, J., K., Hoejgaard, M., V., Holstein-Rathlou, N-H.: "Approximate Entropy and Point Correlation Dimension of Heart Rate Variability in Healthy Subjects", *Integrative Physiological and Behavioral Science*, 33/4, 1998, pp.315-320.

[13] De Boer, R., W., Karemaker, J., M., Strackee, J.: "Hemodynamic Fluctuations and Baroreflex Sensitivity in Humans: a Beat-To-Beat Model", *American Journal on Physiology 253 (Heart Circulation Physiology 22)*, 1987, pp.680-689.

[14] Kanters, J., K., Hojgaard, M., V., Agner, E. & Holstein Rathlou, N., H.: "Short- and Long-Term Variations in Nonlinear Dynamics of Heart Rate Variability", *Cardiovascular Research*, Vol.31, 1996, pp.400-409.

[15] Eckmann, J.-P., Kamphorst, O., Ruelle, D.: "Recurrence Plots of Dynamical Systems". *Europhysics Letters*, 1987, pp.973-979.

[16] Marwan, N., Wessel, N., Meyerfeldt, U., Schirdewan, A., Kurths, J.: "Recurrence-Plot-Based Measures of Complexity and their Application to Heart-Rate-Variability Data", *Physical Review*, E 66, vol. 66 (2), 2002, pp. 026702.1-026702.8

[17] PhysioBank: *MIT-BIH Arrhythmia Database*, retrieved in April 2001 from http://www.physionet.org/physiobank/database/mitdb/

[18] Zitek, P., Bila, J., Kuchar, P.: "Blood Circularion Model Establishing Heart Rate Variability as Control Performance", *Computational Intelligence for Modelling, Control &Automation*, IOS Press, Vienna, Austria, 1999, pp.305-310

[19] Bukovsky, I.: *Analysis of Cardiovascular System Model and the Interpretation of Chaotic Phenomena in Signals ECG and HRV*. Diploma Thesis, Faculty of Mechanical Engineering, CTU in Prague, 2002.

[20] Bila J., Zitek, P., Kuchar, P. and Bukovsky, I.: "Heart Rate Variability: Modelling and Discussion", *Proceedings of International IAESTED Conference on Neural Networks*, Pittsburgh, USA, ISBN 0-88986-286-9, 2000, pp.54-59.

[21] Bila, J., Bukovsky, I.: "Modelling and Interpretation of Chaotic Phenomena in Heart Rate". In: *Proceedings of 8th International Conference on Soft Computing*, *MENDEL* 2002, Brno, Czech Republic, ISBN 80-214-2135-5, 2002, pp.292-297.

[22] Bila, J., Bukovsky, I.: "Interpretation of Chaotic Phenomena in Heart Rate". In: *Proceedings of Workshop 2002, Part B*, February 2002, Vol.6, Special Issue, Czech Technical University, Czech Republic, Prague, ISBN 80-01-02511-X, 2002, pp.908-909.

[23] Bila, J., Bukovsky, I., Oliviera, T., Martins, J.: "Modeling of Influence of Autonomic Neural System to Heart Rate Variability", *IASTED International Conference On Artificial Intelligence And Soft Computing ~Asc 2003~,* Banff, Canada, 2003, pp.345-350.

[24] Bila, J. - Ulicny, D.: "Analysis of Chaotic Signals: Non-linear Methods versus Neural Networks", *Proceedings of 3rd International Carpathian Control Conference*, Ostrava: VSB-TUO, vol. 1, 2002, pp.481-486.

[25] Bila, J., Vitkaj, J., Mánková, R. and Kváča, P.: "Prediction of Chaotic Signals by Means of Neural Networks", In: *Proc. of 3rd Int. Conf. on Soft Computing* – MENDEL 97, Brno, CR, ISBN 80-214-0884-07, 1997, pp.328-333.

[26] Vitkaj, J.: *Analysis of Chaotic signals by Means of Neural Networks*. [PhD. Thesis] (in Czech), Faculty of Mechanical Engineering, Czech Technical University in Prague, Czech Republic, 2001.

[27] Mankova, R.: *Prediction of Chaotic Signals Using Neural Networks with Focus on Analysis of Cardiosignals* [Candidate Dissertation] (in Czech), Faculty of Mechanical Engineering ,Czech Technical University in Prague, Czech Republic, 1997.

[28] Bíla, J., Vitkaj, J., Musil, M., Bukovsky, I.: "Some Limits of Neural Networks Use in Diagnostics" (in Czech), *Automatizace*, vol. 46, issue 11, 2003, Prague, ISSN 0005 -125X, pp.734-737.

[29] Howard D., Mark B.: *Neural Network Toolbox for Use with MATLAB, User's Guide*, Version 3.0, MathWorks, 1998.

[30] Hopfield, J.: "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Nat. Sci*. USA, Vol. 79, pp.2554-2558.

[31] Pineda, F. J.: "Dynamics and Architecture for Neural Computation", *Journal on Complexity*, Vol.4, pp.216-245, Sept. 1988.

[32] Ivakhnenko, A. G.: "Polynomial Theory of Complex Systems*", IEEE Transactions on Systems, Man, and Cybernetics,* 1 (4), 1971, pp. 364-378.
(available at http://www.gmdh.net/articles/history/polynomial.pdf, 11/2006).

[33] Nikolaev, N. and Iba, H.: "Learning Polynomial Feedforward Networks by Genetic Programming and Backpropagation", *IEEE Transactions on Neural Networks*, 2003, vol.14, N:2, pp.337-350.

[34] Wen Yu: "Nonlinear System Identification Using Discrete-Time Recurrent Neural Networks with Stable Learning Algorithms", *Information Sciences—Informatics and Computer Science: An International Journal*, Elsevier Science Inc., Volume 158, ISSN:0020-0255, January 2004, pp.131-147.

[35] Narendra, K., S., Parthasarathy, K.: "Identification and Control of Dynamical Systems Using Neural Networks ", *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, Mar. 1990, pp.4-27.

[36] Narendra, K., S., Parthasarathy, K.: "Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks", *IEEE Transactions on. Neural Networks*, Vol.2, 1991, pp.4-27.

[37] Polycarpou, M., M., Ionnaou, P., A.: "Learning and Convergence Analysis of Neural-Type Structured Networks," *IEEE Transaction on Neural Networks* 3 (1), 1992, pp.39-50.

[38] Xuc-Bin Liang, Jun Wang : "Absolute Exponential Stability of Neural Network with a General Class of Activation Function"*, IEEE Transaction on Circuits and Systems-I: Fundamental Theory and Applications* , IEEE Computer Society, vol. 47, No. 8, August 2000, pp.1258-1262.

[39] Xue-Bin Liang, Jennie Si: "Global Exponential Stability of Neural Networks with Global Lipschitz Continuous Activations and Its Application to Linear Variational Inequality Problem", *IEEE Transaction on Neural Networks*, IEEE Computer Society, vol. 12, No. 2, 1045-9227, March 2001, pp.349-359.

[40] Youshen Xia, Jun Wang: "Global Asymptotic and Exponential Stability of a Dynamic Neural System with Asymmetric Connection Weigths", *IEEE Transaction on Automatic Control*, IEEE Computer Society, vol. 46, No. 4, April 2001, pp.635-638.

[41] Sanging Hu, Jun Wang: “Global Asymptotic Stability and Global Exponentional Stability of Continuous-Time Recurrent Neural Networks”, *IEEE Transaction on Automatic Control*, IEEE Computer Society, vol. 47, No.5, May 2002, pp.802-807.

[42] Sabri Arik: “Global Asymptotic Stability of a Class of Dynamical Neural Networks”, *IEEE Transaction on Circuits and Systems-I: Fundamental Theory and Applications* , IEEE Computer Society, vol. 47, No. 4, April 2000, pp. 568-571.

[43] Sabri Arik: “A Note on the Global Stability of Dynamical Neural Networks”, IEEE *Transaction on Circuits and Systems-I: Fundamental Theory and Applications* , IEEE Computer Society, vol. 49, No.4, April 2002, pp.502-504.

[44] Oysal Yusuf , Sunderam Vaidy S., Albada Geert Dick van, Sloot Peter M.A., Dongarra Jack J.: “Time Delay Dynamic Fuzzy Networks for Time Series Prediction”, in *ICCS : International Conference on Computational Science,* N^{o}5, Atlanta, vol. 3516, , ISBN 3-540-26032-3, 2005, pp.775-782. Also in Lecture Notes in Computer Science ISSN 0302-9743,

[45] Zitek, P., Vyhlidal, T.: “Low Order Time Delay Approximation of Conventional Linear Model”. In: *4th MATHMOD Vienna Proceedings*, Vienna 2003, p.197-204.

[46] Vyhlidal, T. and Zitek, P.: “Control System Design Based on a Universal First-Order Model with Time Delays”, *The 2nd International Conference on Advanced Engineering Design*. Glasgow: University of Glasgow, and *Acta Polytechnica*. vol. 41, no. 4-5, 2001, pp.49-53.

[47] Andrlik, V., Jakubsky, O., Steinbauer, P., Talacko, J., Valasek, M., *et al*.: “Parallel Manipulator with Direct Hydraulic Drive” (in Czech) In: *Development of Methods and Tools of Integrated Engineering*, Prague: Czech Technical University in Prague, FME, ISBN 80-01-02652-3, 2002, pp.72-74.

[48] Gupta, M., M., Liang, Jin: “Stable Dynamic Back Propagation Learning in Recurrent Neural Networks”, *IEEE Transaction on Neural Networks*, IEEE Computer Society, vol. 10, No. 6, 1045-9227, November 1999, pp.1321-1334.

[49] Gupta, M., M., Liang J., and Homma N.: *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*, IEEE Press and Wiley-Interscience, published by John Wiley & Sons, Inc., 2003.

[50] Redlapalli, S., Song. K. –Y., and Gupta, M., M.*: “*Development of Quadratic Neural Unit With Applications to Pattern Classification”*, The Fourth International Symposium on Uncertainty Modeling and Analysis ISUMA 2003*, IEEE Computer Society, Maryland USA, ISBN 0-7695-1997-0, 2003, pp.141-146.

[51] Song, K. -Y., Redlapalli, S. and Gupta, M., M.: “Cubic Neural Units for Control Applications”, *The Fourth International Symposium on Uncertainty Modeling and Analysis ISUMA 2003*, IEEE Computer Society, Maryland USA, ISBN 0-7695-1997-0, 2003, pp.324-329.

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

[53] Bukovsky, I.: *Development of Higher-Order Nonlinear Neural Units as a Tool for Approximation, Identification and Control of Complex Nonlinear Dynamic Systems and Study of Their Application Prospects for Nonlinear Dynamics of Cardiovascular System*, Final Report from NATO Science Fellowships research,(ISRL, U. of S., Canada 2003), FME Czech Technical University, Prague (IGS #CTU0304112), 2003.

[54] Bíla, J. - Bukovský, I.: “Nonlinear Dynamic Neural Units for Parallel Manipulator TRIPOD “(in Czech) **,** In: *Seminar Proceedings VZ MSM 212200008* [CD-ROM]. Prague: CTU FME, Vol. 1, ISBN 80-01-03105-5, 2004, pp.66-68.

[55] Bukovsky I., Bila J.: “Development of Higher Order Nonlinear Neural Units for Evaluation of Complex Static and Dynamic Systems”, *Proceedings of Workshop 2004, Part A*, Vol.8, Special Issue, Czech Technical University, Czech Republic, Prague, March 2004, pp.372-373.

[56] Bukovsky, I., Bíla, J., Gupta, M., M.: “Linear Dynamic Neural Units with Time Delay for Identification and Control” (in Czech), In: *Automatizace*, Vol. 48, No. 10, Prague, Czech Republic, ISSN 0005-125X, October 2005, pp.628-635.

[57] Bukovsky, I.: “Extended Dynamic Neural Architectures HONNU with Minimum Number of Neural Parameters for Evaluation of Nonlinear Dynamic Systems “(in Czech), In: *New Methods*

and Approaches in the Fields of Control Technology, Automatic Control, and Informatics, Czech Technical University, Prague, ISBN 80-01-03240-X, 2005, pp.93-97.

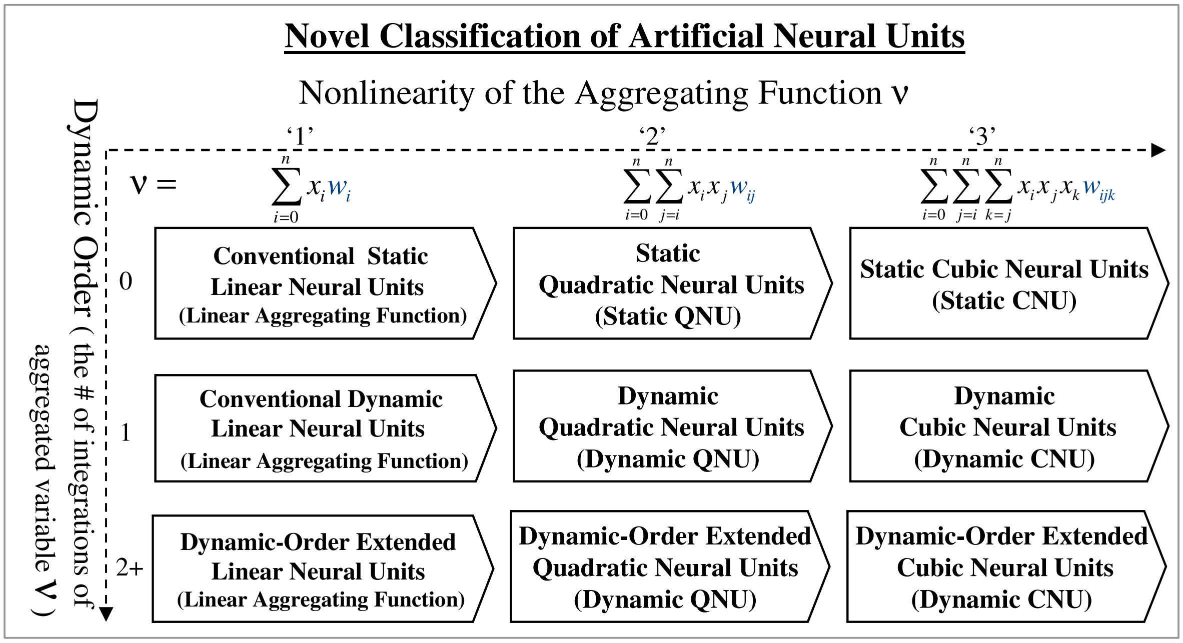
[58] Bukovsky, I., Simeunovic, G.:" Dynamic-Order-Extended Time-Delay Dynamic Neural Units", *The 8th Seminar on Neural Network Applications in Electrical Engineering, NEUREL-2006*, IEEE (SCG) CAS-SP, Belgrade, ISBN 1-4244-0432-0, 2006.

[59] Bukovsky, I., Bíla, J., Gupta, M., M.: "Stable Neural Architecture of Dynamic Neural Units with Adaptive Time Delays*", The 7th International FLINS Conference on Applied Artificial Intelligence*, ISBN 981-256-690-2, 2006, pp.215-222.

[60] Bukovsky, I., Hou, Z-G., Gupta, M., M., Bila, J.: "Foundation of Notation and Classification of Nonconventional Static and Dynamic Neural Units", acceted paper for special section on neural networks for *ICCI 2007, The 6th IEEE International Conference on COGNITIVE INFORMATICS*, California , USA, 2007

Appendix

Appendix-1 Classification of Nonconventional Neural Units



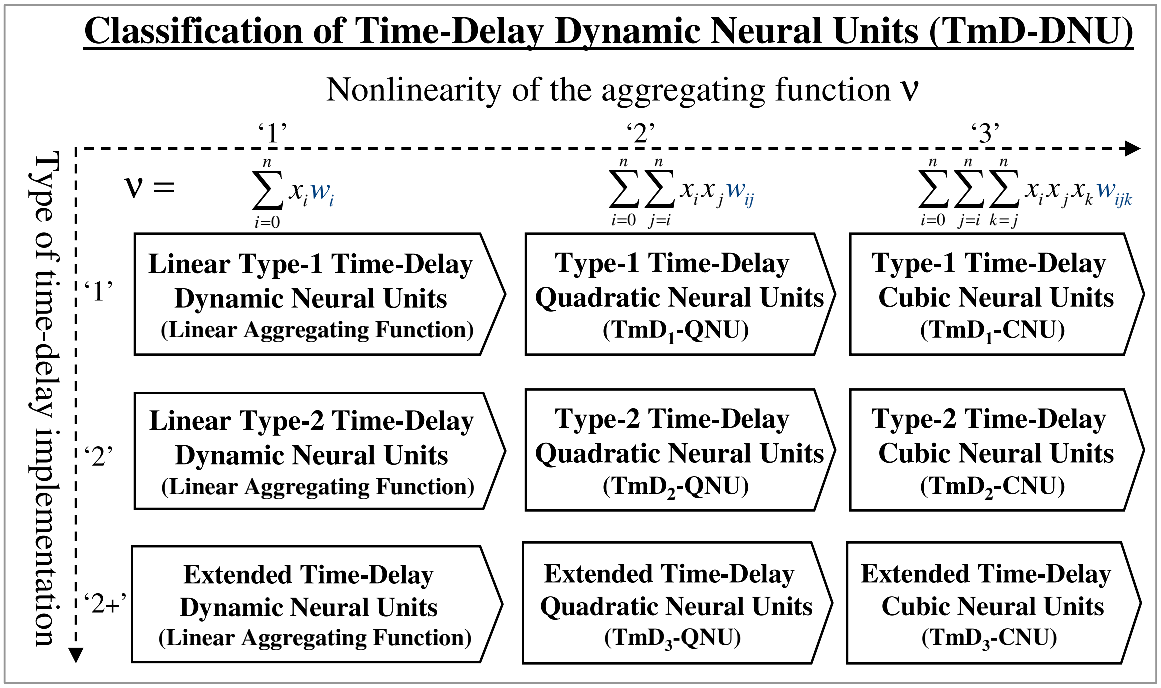
Novel Classification of Artificial Neural Units

Nonlinearity of the Aggregating Function ν

Dynamic Order (the # of integrations of aggregated variable ν)

	‘1’	‘2’	‘3’
ν =	$\sum_{i=0}^{n} x_i w_i$	$\sum_{i=0}^{n}\sum_{j=i}^{n} x_i x_j w_{ij}$	$\sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} x_i x_j x_k w_{ijk}$
0	**Conventional Static Linear Neural Units (Linear Aggregating Function)**	**Static Quadratic Neural Units (Static QNU)**	**Static Cubic Neural Units (Static CNU)**
1	**Conventional Dynamic Linear Neural Units (Linear Aggregating Function)**	**Dynamic Quadratic Neural Units (Dynamic QNU)**	**Dynamic Cubic Neural Units (Dynamic CNU)**
2+	**Dynamic-Order Extended Linear Neural Units (Linear Aggregating Function)**	**Dynamic-Order Extended Quadratic Neural Units (Dynamic QNU)**	**Dynamic-Order Extended Cubic Neural Units (Dynamic CNU)**

Figure Appendix - 1: Classification of basic nonconventional continuous artificial neural units according to aggregating nonlinearity $f_{HONNU} = \nu$ and its time integrations (dynamic order).



Classification of Time-Delay Dynamic Neural Units (TmD-DNU)

Nonlinearity of the aggregating function ν

Type of time-delay implementation

	‘1’	‘2’	‘3’
ν =	$\sum_{i=0}^{n} x_i w_i$	$\sum_{i=0}^{n}\sum_{j=i}^{n} x_i x_j w_{ij}$	$\sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} x_i x_j x_k w_{ijk}$
‘1’	**Linear Type-1 Time-Delay Dynamic Neural Units (Linear Aggregating Function)**	**Type-1 Time-Delay Quadratic Neural Units (TmD$_1$-QNU)**	**Type-1 Time-Delay Cubic Neural Units (TmD$_1$-CNU)**
‘2’	**Linear Type-2 Time-Delay Dynamic Neural Units (Linear Aggregating Function)**	**Type-2 Time-Delay Quadratic Neural Units (TmD$_2$-QNU)**	**Type-2 Time-Delay Cubic Neural Units (TmD$_2$-CNU)**
‘2+’	**Extended Time-Delay Dynamic Neural Units (Linear Aggregating Function)**	**Extended Time-Delay Quadratic Neural Units (TmD$_3$-QNU)**	**Extended Time-Delay Cubic Neural Units (TmD$_3$-CNU)**

Figure Appendix - 2: Classification of basic time-delay dynamic neural units according to aggregating nonlinearity ν and the type of delay implementation.

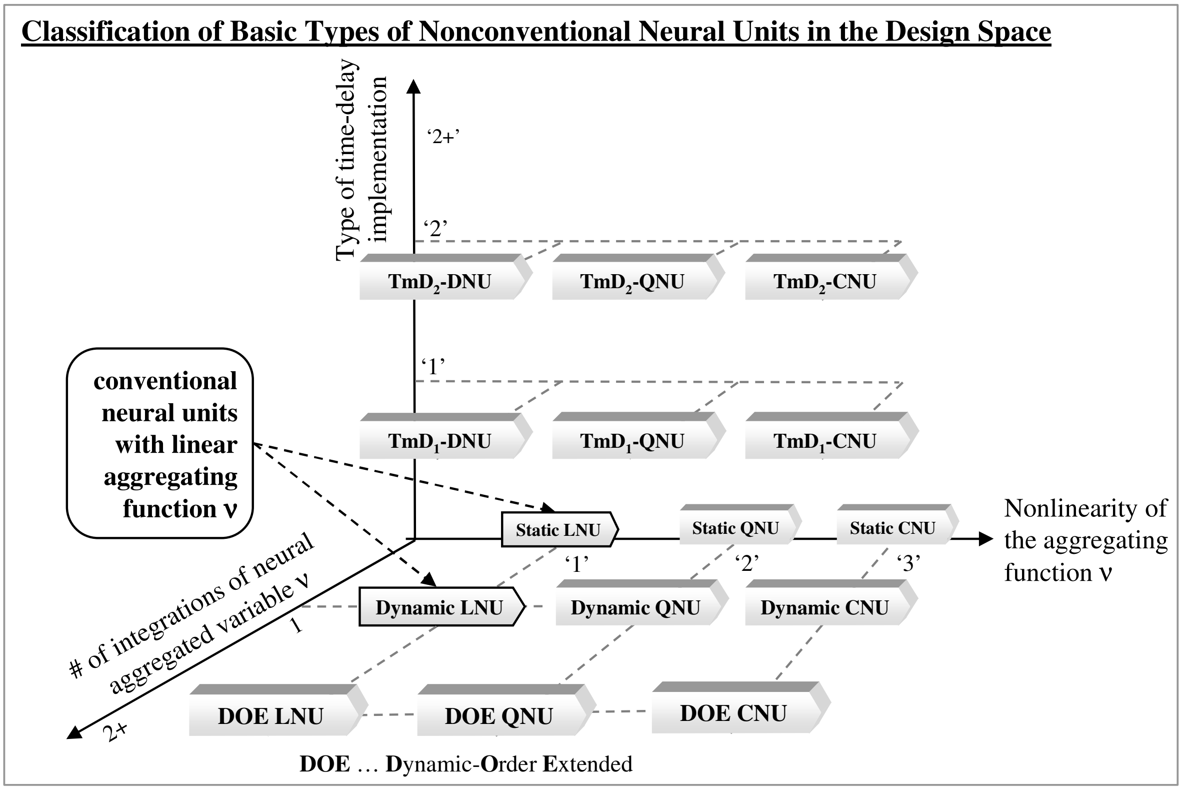

Classification of Basic Types of Nonconventional Neural Units in the Design Space
Type of time-delay implementation
'2+'
'2'
'1'
TmD$_2$-DNU
TmD$_2$-QNU
TmD$_2$-CNU
TmD$_1$-DNU
TmD$_1$-QNU
TmD$_1$-CNU
conventional neural units with linear aggregating function ν
Static LNU
Static QNU
Static CNU
Nonlinearity of the aggregating function ν
'1'
'2'
'3'
# of integrations of neural aggregated variable ν
1
2+
Dynamic LNU
Dynamic QNU
Dynamic CNU
DOE LNU
DOE QNU
DOE CNU
DOE … Dynamic-Order Extended

Figure Appendix - 3: Novel classification of basic artificial neural units according to aggregating nonlinearity *ν,* its time integrations (i.e., the dynamic order), and adaptable time-delay implementation; only some of most general types are shown for simplicity (not all types are shown for simplicity of the picture).

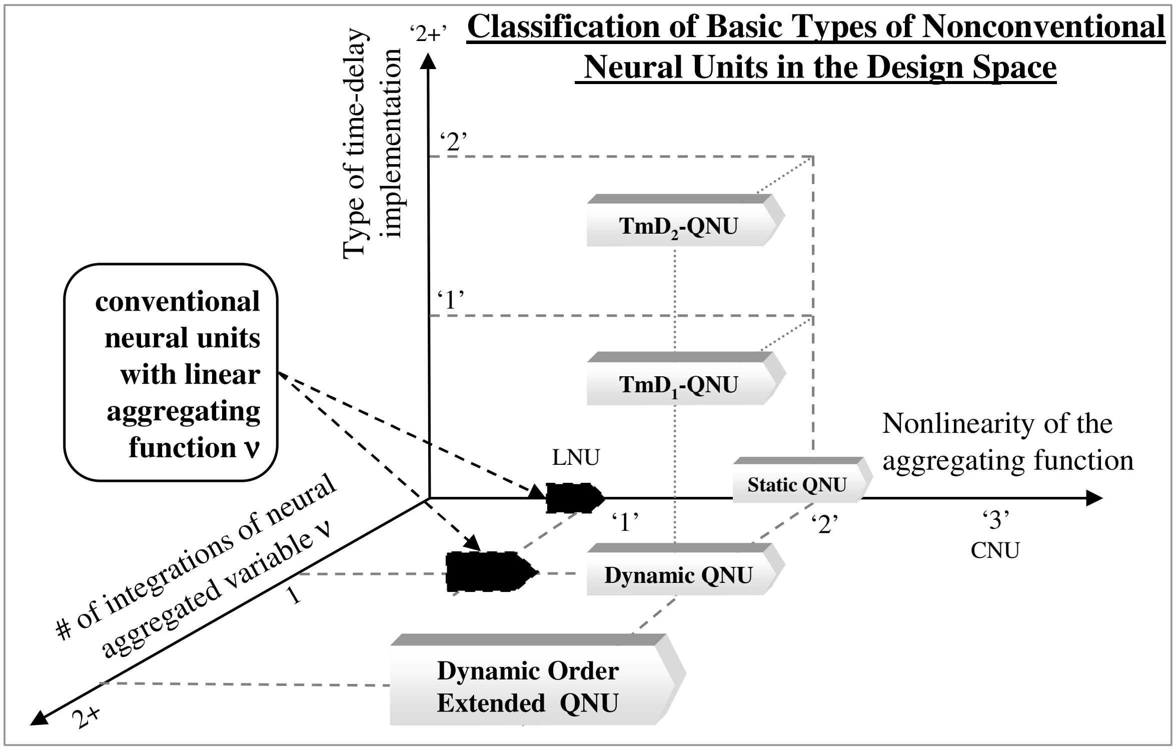

Classification of Basic Types of Nonconventional Neural Units in the Design Space
'2+'
Type of time-delay implementation
'2'
TmD$_2$-QNU
'1'
TmD$_1$-QNU
conventional neural units with linear aggregating function ν
LNU
Static QNU
Nonlinearity of the aggregating function
'1'
'2'
'3'
CNU
Dynamic QNU
# of integrations of neural aggregated variable ν
1
2+
Dynamic Order Extended QNU

Figure Appendix - 4: Novel classification of basic artificial neural units. A detail of static QNU, dynamic QNU, DOE-QNU, TmD$_1$-QNU, and TmD$_2$-QNU.

Appendix-2 Basic Architectures of New Neural Units

Discrete Dynamic HONNU

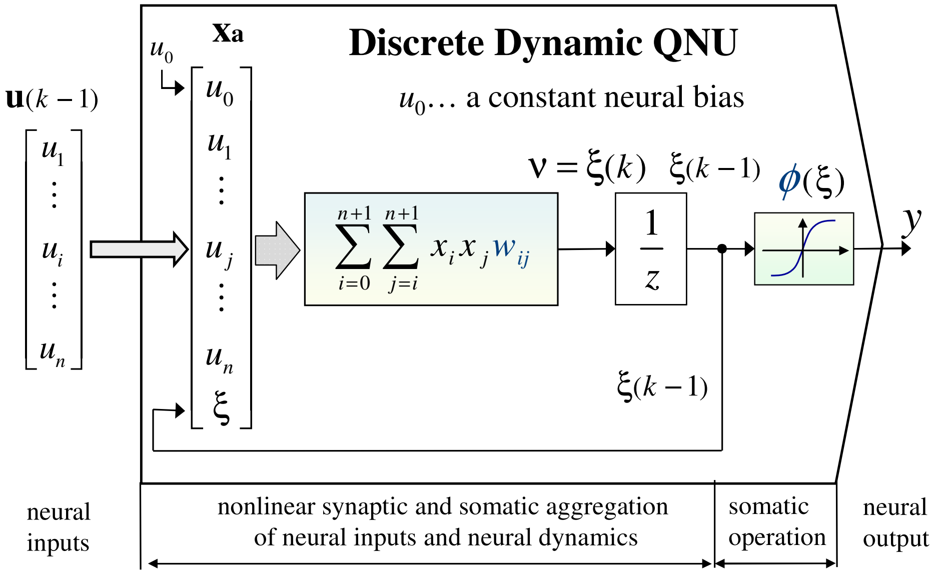


Figure Appendix - 5: Discrete dynamic quadratic neural unit.

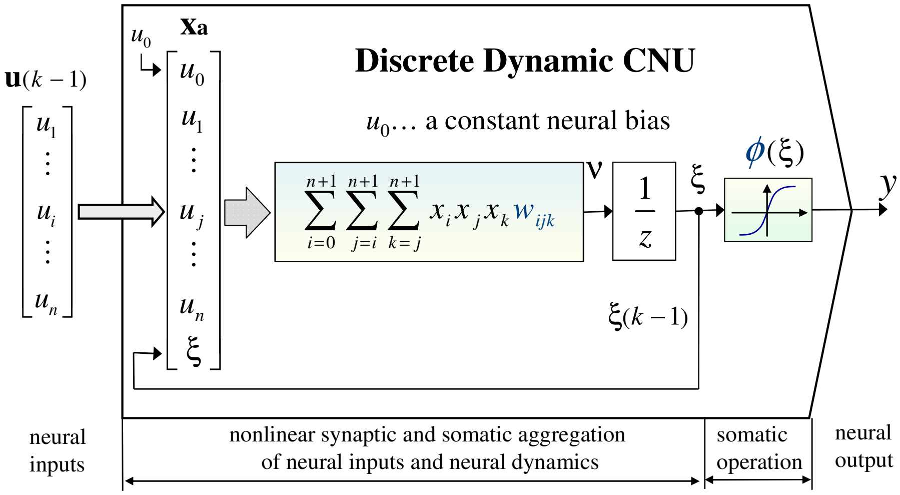

Discrete Dynamic CNU
$u_0 \ldots$ a constant neural bias
$\mathbf{u}(k-1)$
u_1
u_i
u_n
Xa
u_0
u_0
u_1
u_j
u_n
ξ
$\sum_{i=0}^{n+1}\sum_{j=i}^{n+1}\sum_{k=j}^{n+1} x_i x_j x_k w_{ijk}$
ν
$\frac{1}{z}$
ξ
$\phi(\xi)$
y
$\xi(k-1)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics
somatic operation
neural output

Figure Appendix - 6: Disrete dynamic cubic neural unit.

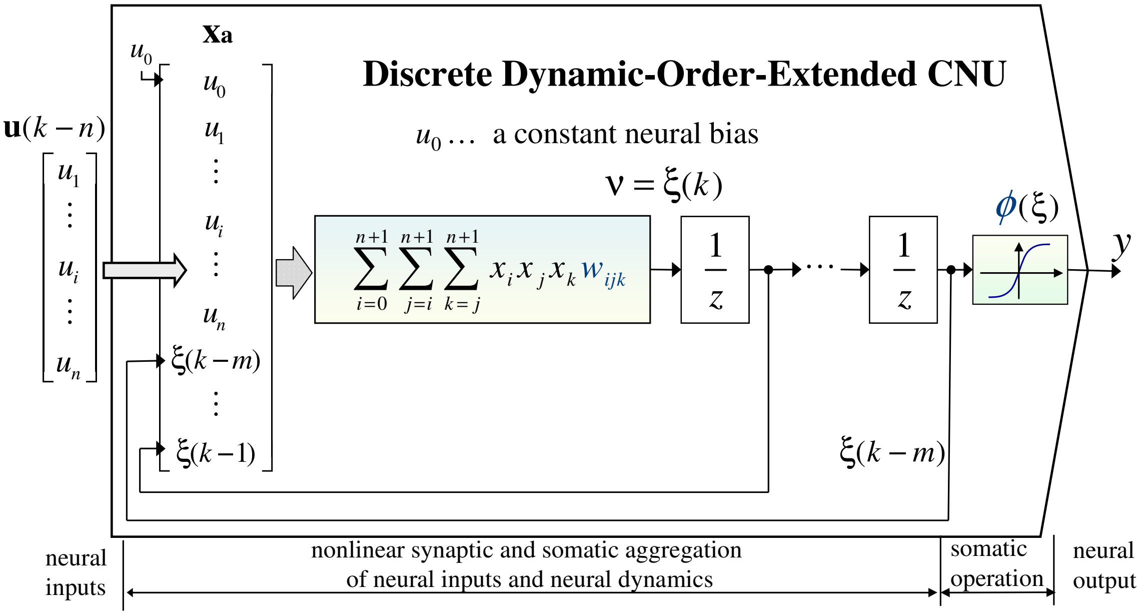
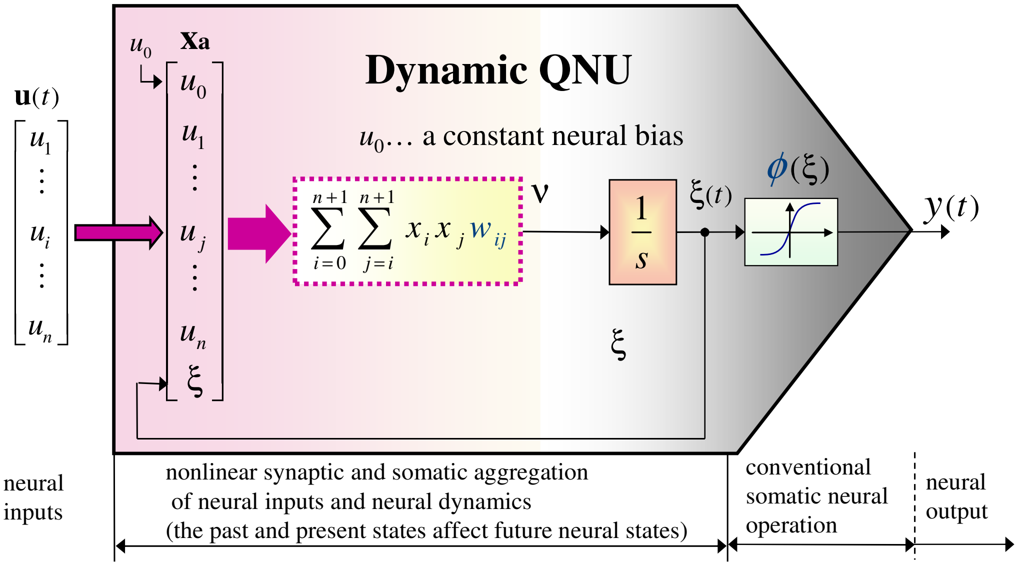

Discrete Dynamic-Order-Extended CNU
$u_0 \ldots$ a constant neural bias
$\nu = \xi(k)$
$\mathbf{u}(k-n)$
u_1
u_i
u_n
Xa
u_0
u_0
u_1
u_i
u_n
$\xi(k-m)$
$\xi(k-1)$
$\sum_{i=0}^{n+1}\sum_{j=i}^{n+1}\sum_{k=j}^{n+1} x_i x_j x_k w_{ijk}$
$\frac{1}{z}$
$\frac{1}{z}$
$\phi(\xi)$
y
$\xi(k-m)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics
somatic operation
neural output

Figure Appendix - 7: Discrete DOE-CNU, i.e. CNU with multiple step delay and multiple feedback of aggregated variable *ν*.

Continuous Dynamic HONNU


Dynamic QNU
$u_0 \ldots$ a constant neural bias
$\mathbf{u}(t)$
u_1
u_i
u_n
Xa
u_0
u_0
u_1
u_j
u_n
ξ
$\sum_{i=0}^{n+1}\sum_{j=i}^{n+1} x_i x_j w_{ij}$
ν
$\frac{1}{s}$
$\xi(t)$
ξ
$\phi(\xi)$
$y(t)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics (the past and present states affect future neural states)
conventional somatic neural operation
neural output

The neural state is represented by the variable ξ which represent the level of signal carried through axon forward to neural outputs.

Figure Appendix - 8: Continuous time dynamic quadratic neural unit.

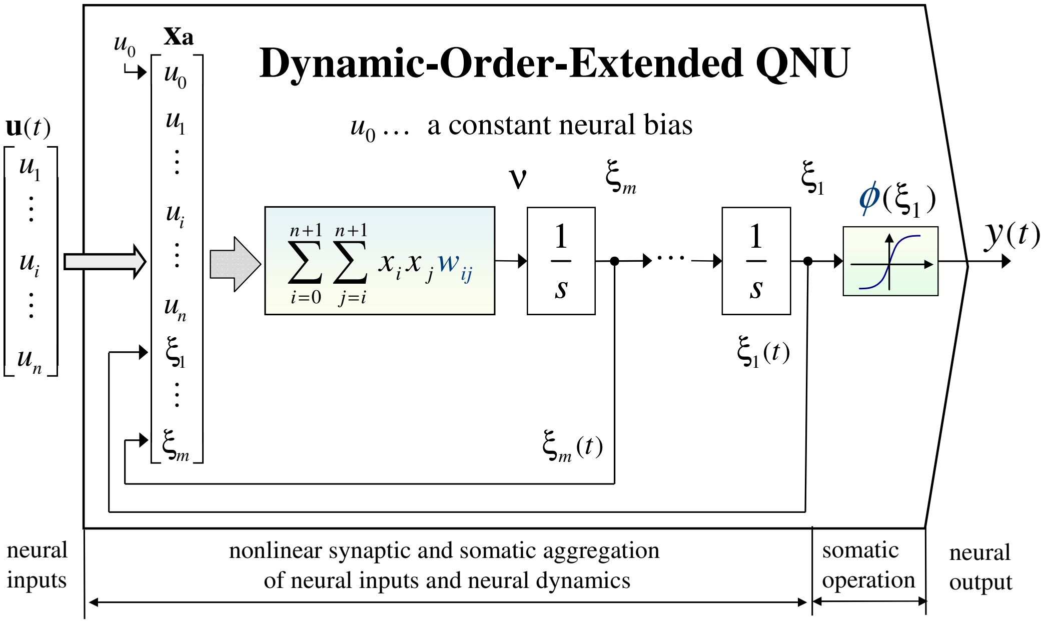


Figure Appendix - 9: DOE-QNU, i.e., QNU with multiple integration and multiple feedback of aggregated variable ν.

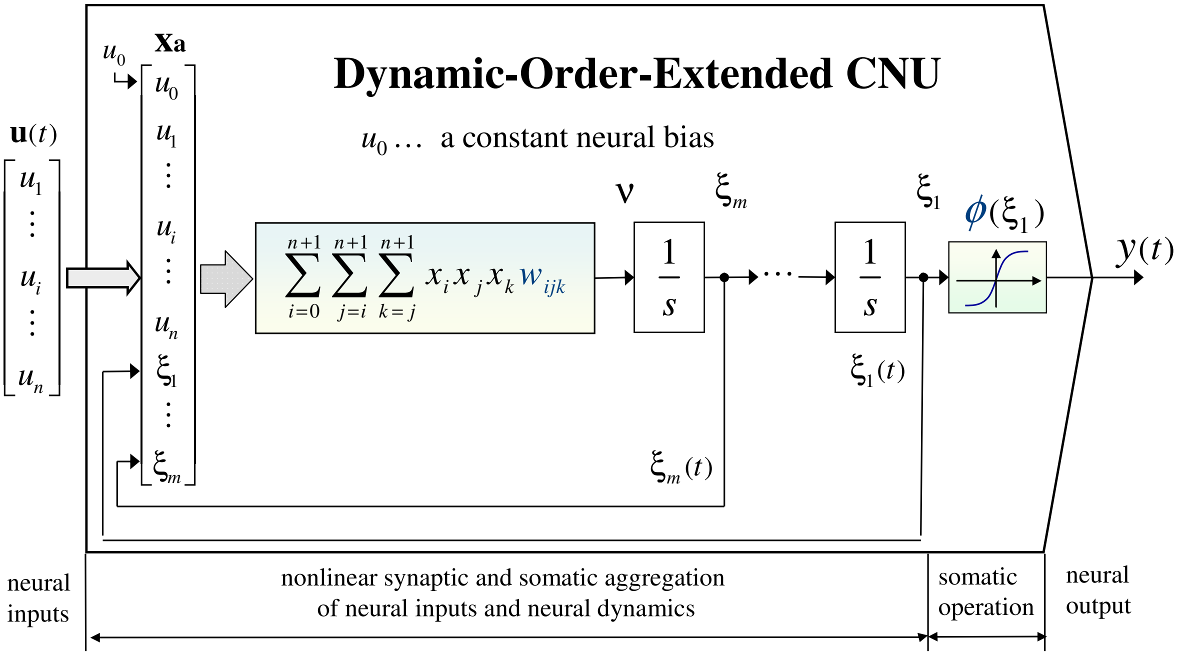
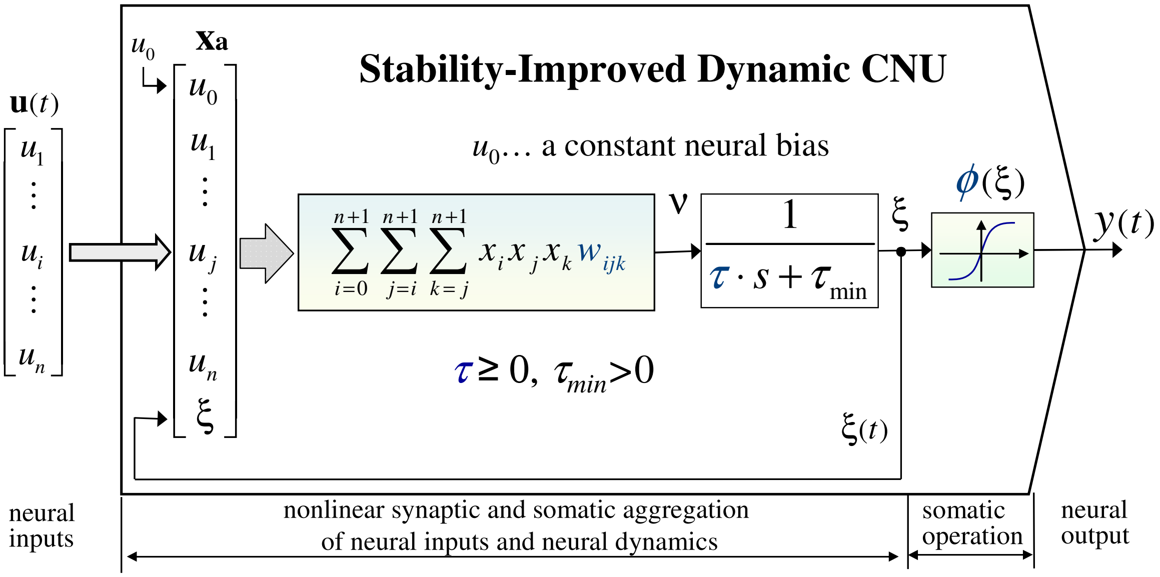
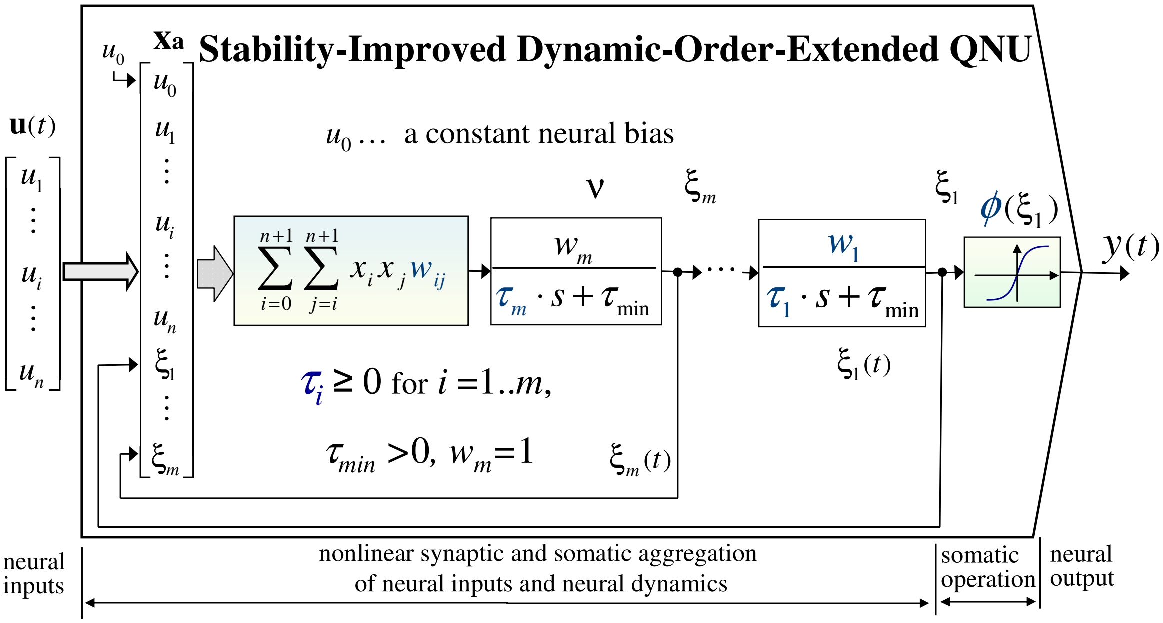

Dynamic-Order-Extended CNU
$u_0 \ldots$ a constant neural bias
$\mathbf{u}(t)$
u_1
u_i
u_n
u_0
Xa
u_0
u_1
u_i
u_n
ξ_1
ξ_m
$\sum_{i=0}^{n+1}\sum_{j=i}^{n+1}\sum_{k=j}^{n+1} x_i x_j x_k w_{ijk}$
ν
ξ_m
$\frac{1}{s}$
ξ_1
$\frac{1}{s}$
$\phi(\xi_1)$
$y(t)$
$\xi_1(t)$
$\xi_m(t)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics
somatic operation
neural output

Figure Appendix - 10: DOE-CNU, i.e., CNU with multiple integration and multiple feedback of aggregated variable ν.



Stability-Improved Dynamic CNU
u_0… a constant neural bias
$\mathbf{u}(t)$
u_1
u_i
u_n
u_0
Xa
u_0
u_1
u_j
u_n
ξ
$\sum_{i=0}^{n+1}\sum_{j=i}^{n+1}\sum_{k=j}^{n+1} x_i x_j x_k w_{ijk}$
ν
$\frac{1}{\tau \cdot s + \tau_{\min}}$
ξ
$\phi(\xi)$
$y(t)$
$\tau \geq 0,\ \tau_{min} > 0$
$\xi(t)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics
somatic operation
neural output

Figure Appendix - 11: Dynamic CNU with dynamics modified to improve unit's stability. The implementation of first-order-dynamics transfer functions instead of integrators improves the stability of the unit



Stability-Improved Dynamic-Order-Extended QNU
u_0 … a constant neural bias
$\mathbf{u}(t)$
u_1
u_i
u_n
u_0
Xa
u_0
u_1
u_i
u_n
ξ_1
ξ_m
$\sum_{i=0}^{n+1}\sum_{j=i}^{n+1} x_i x_j w_{ij}$
ν
ξ_m
$\frac{w_m}{\tau_m \cdot s + \tau_{\min}}$
$\frac{w_1}{\tau_1 \cdot s + \tau_{\min}}$
ξ_1
$\phi(\xi_1)$
$y(t)$
$\xi_1(t)$
$\tau_i \geq 0$ for $i = 1..m$,
$\tau_{min} > 0,\ w_m = 1$
$\xi_m(t)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics
somatic operation
neural output

Figure Appendix - 12: DOE-QNU with dynamics modified to improve unit's stability

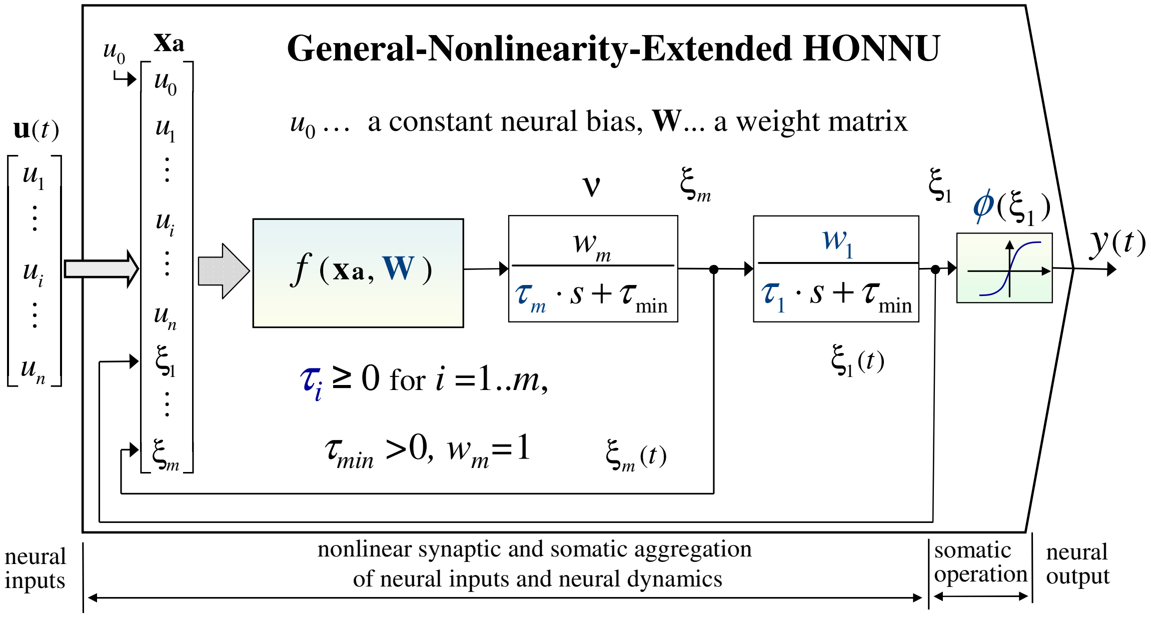

General-Nonlinearity-Extended HONNU
u_0 … a constant neural bias, $\mathbf{W}$... a weight matrix
$\mathbf{u}(t)$
u_1
u_i
u_n
u_0
$\mathbf{x_a}$
u_0
u_1
u_i
u_n
ξ_1
ξ_m
$f(\mathbf{x_a}, \mathbf{W})$
ν
ξ_m
$\frac{w_m}{\tau_m \cdot s + \tau_{\min}}$
ξ_1
$\frac{w_1}{\tau_1 \cdot s + \tau_{\min}}$
$\phi(\xi_1)$
$y(t)$
$\xi_1(t)$
$\tau_i \geq 0$ for $i = 1..m$,
$\tau_{min} > 0$, $w_m = 1$
$\xi_m(t)$
neural inputs
nonlinear synaptic and somatic aggregation of neural inputs and neural dynamics
somatic operation
neural output

Figure Appendix - 13: General sketch of dynamic HONNU with dynamics modified to improve unit's stability.

$$f(\mathbf{x_a}, \mathbf{W}) = w_0 u_1 + \sum_{i=2}^{4} w_i u_i + \sum_{i=5}^{7} w_i u_i + (u_1 - w_7) \sum_{i=2}^{4} \sum_{j=i}^{4} w_{ij} \frac{u_i' u_j'}{e^{(-w_{e_{ij}}(u_i - u_j)^2)}},$$

where u'_i and u'_j enter the unit as measurable inputs if available (see Subsection 7.2.1, p.92).

Continuous Time-Delay Dynamic Higher-Order Nonlinear Neural Units (TMD-DHONNU)

$T_f \geq 0$, $T_i \geq 0$ for $i=0\ldots n$, $\tau \geq 0$, $\tau_{min} > 0$

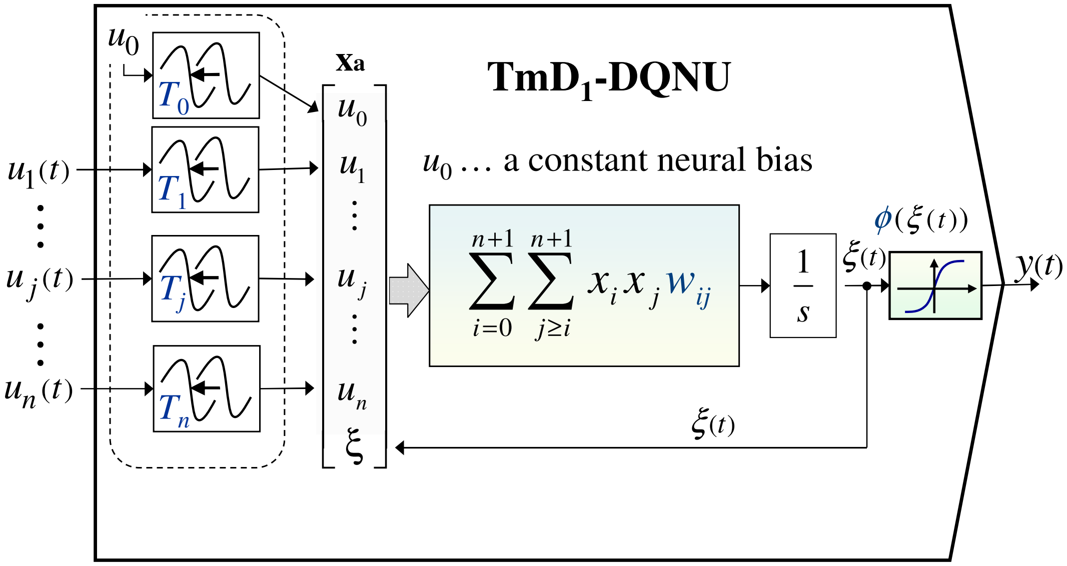


Figure Appendix - 14: Time-Delay Dynamic Quadratic Neural Unit – Type 1 (TmD$_1$-DQNU) with adaptable time delays T_i on its inputs.

$T_f \geq 0$, $T_i \geq 0$ for $i=0\ldots n$

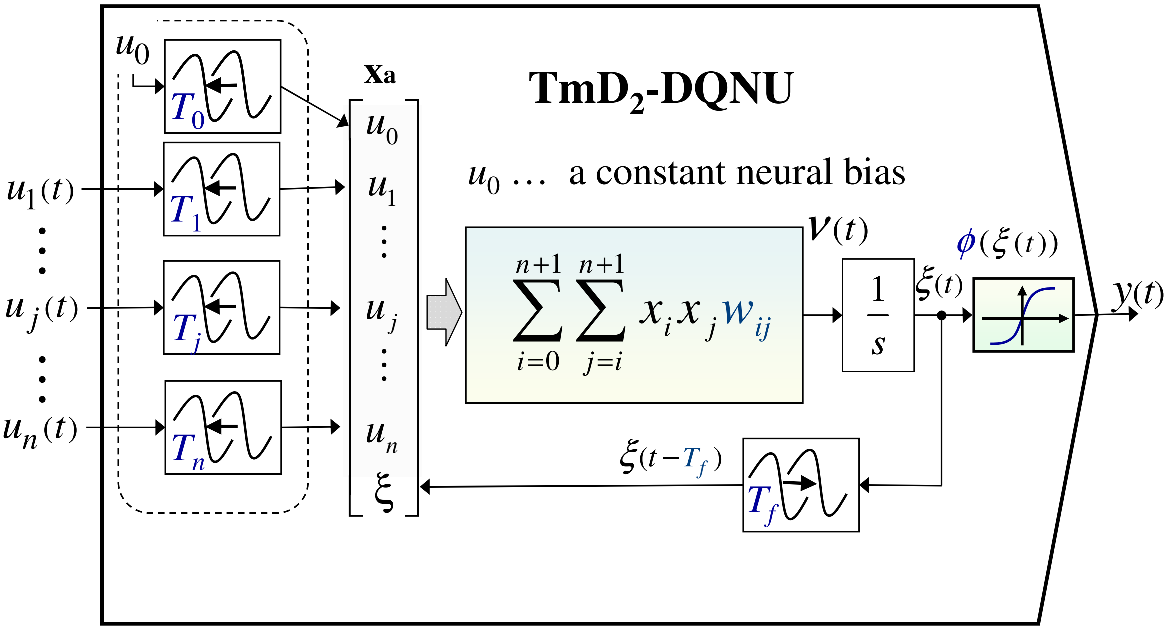


Figure Appendix - 15: Time-Delay Dynamic Quadratic Neural Unit – Type 2 (TmD$_2$-DQNU) with adaptable time delays T_i on its inputs and with adaptable delay T_f in state feedback of a unit.

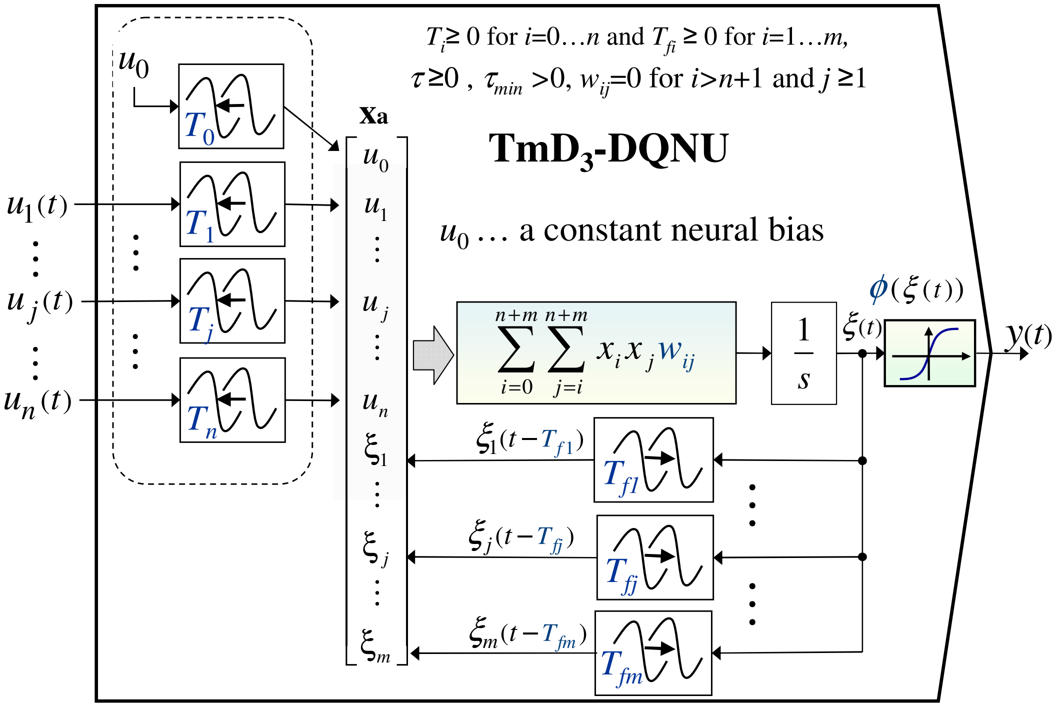


Figure Appendix - 16: Time-Delay Dynamic Quadratic Neural Unit – Type 3 (TmD$_3$-DQNU) with adaptable time delays T_i on its inputs and with adaptable delays T_{fi} in state feedback of a unit.

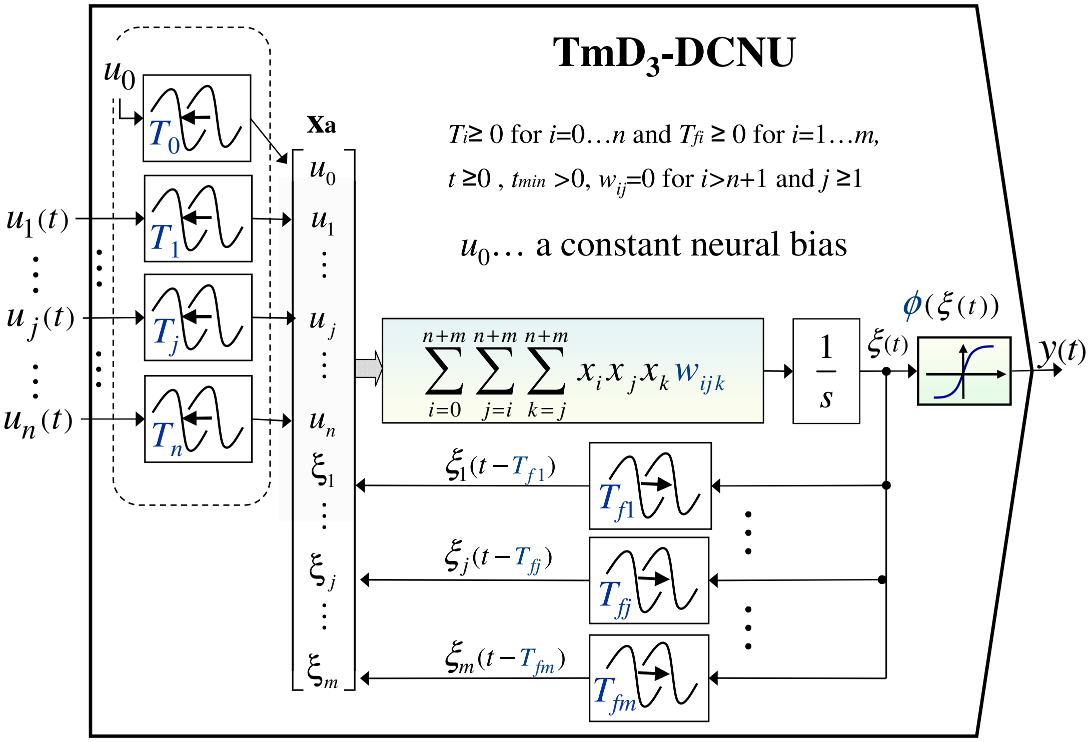

TmD$_3$-DCNU
$T_i \geq 0$ for i=0…n and $T_{fi} \geq 0$ for i=1…m,
$t \geq 0$, $t_{min} > 0$, w_{ij}=0 for i>n+1 and $j \geq 1$
u_0… a constant neural bias
u_0
T_0
T_1
T_j
T_n
$u_1(t)$
$u_j(t)$
$u_n(t)$
xa
u_0
u_1
u_j
u_n
ξ_1
ξ_j
ξ_m
$\sum_{i=0}^{n+m} \sum_{j=i}^{n+m} \sum_{k=j}^{n+m} x_i x_j x_k w_{ijk}$
$\frac{1}{s}$
$\xi(t)$
$\phi(\xi(t))$
$y(t)$
$\xi_1(t-T_{f1})$
$\xi_j(t-T_{fj})$
$\xi_m(t-T_{fm})$
T_{f1}
T_{fj}
T_{fm}

Figure Appendix - 17: Time-Delay Dynamic Cubic Neural Unit – Type 3 (TmD$_3$-DCNU) with adaptable time delays T_i on its inputs and with adaptable delays T_{fi} in state feedback of a unit.

Appendix-3 Deterministic-Chaos Component in HRV due to Autonomous Nervous System in Simulated Heart Beat Tachograms

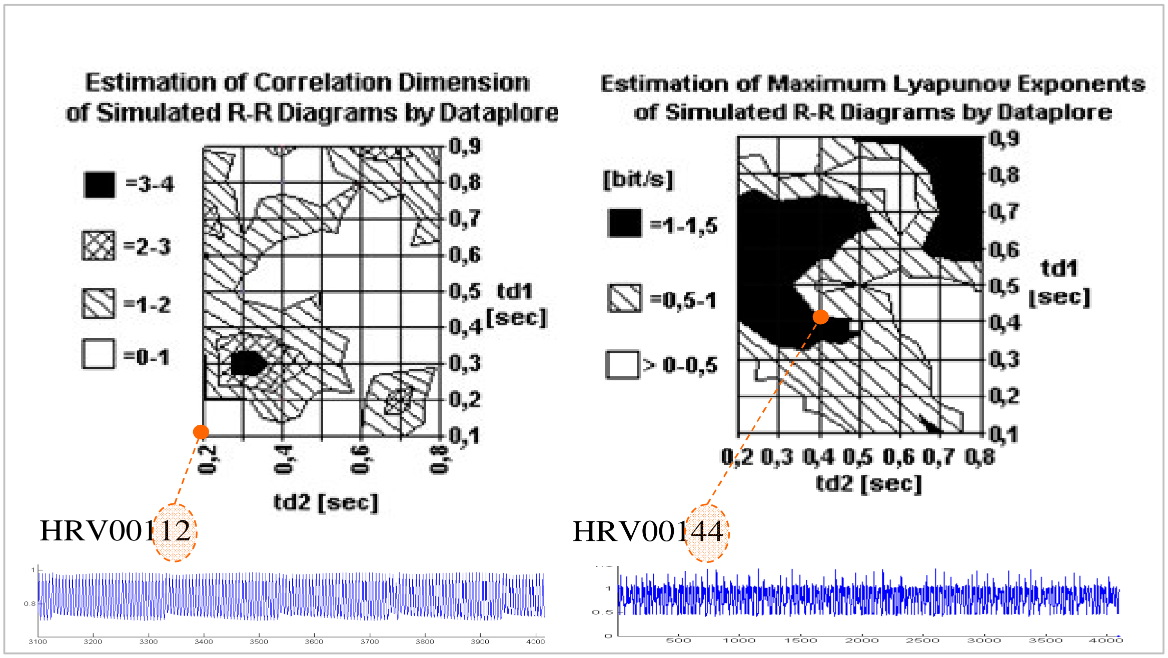


Figure Appendix - 18: Estimation of correlation dimension and largest Lyapunov exponents of the simulated (deterministic) heart beat tachograms.

		Td2						
Td1	Invariants	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.1	cd	0.9	0.597	0.33	0.9	0.24	1.31	0
	LE dataplore	0.461582	0.069697	0.1782	0.577496	0.87679	1.051364	0
0.2	cd	0.88	0.91	1.54	0.7	0.77	2.6	0
	LE dataplore	0.192896	0.112913	0.743975	0.94718	0.951438	0.349783	0
0.3	cd	0.88	3.72	2.63	1.78	0.16	0.61	0.85
	LE dataplore	0.092986	0.912021	0.931753	0.961684	0.711637	0.376044	0.345889
0.4	cd	1.12	1.61	1.4	1.01	0.64	0.79	0.64
	LE dataplore	0.98697	1.196473	1.058155	1.018447	0.408001	0.228432	0.175687
0.5	cd	1.69	0.79	0.48	0.97	0.47	0.82	0.54
	LE dataplore	1.222222	1.312413	0.365175	0.491596	0.318435	0.317947	0.550687
0.6	cd	0.93	1.28	0.53	0.14	0.39	0.62	0.44
	LE dataplore	1.152495	1.192635	1.218797	0.743845	0.723132	1.213171	1.215821
0.7	cd	2.71	0.77	1.58	1.35	0.55	0.1	1.95
	LE dataplore	1.117804	1.221	1.326558	1.247193	0	1.024518	1.4002
0.8	cd	1.58	0.57	0.69	0.36	1.02	1.45	1.27
	LE dataplore	0.829657	0.420043	0.508948	0	0	1.277436	1.406169
0.9	cd	0.91	0.82	1.13	0.8	2.34	2.26	0.59
	LE dataplore	0.160882	0.223682	0.496045	1.087937	1.222046	1.228941	1.303357

Table Appendix - 1: Estimated CD and LLE of tachograms (4096 heartbeats each, evaluated by Dataplore) simulated by model [18] (to [23]).

Method Converged		Td2						
		0.2	0.3	0.4	0.5	0.6	0.7	0.8
Td1	0.1	yes	yes					
	0.2	yes	yes				yes	
	0.3	yes						yes
	0.4							yes
	0.5							
	0.6							
	0.7	yes						
	0.8							
	0.9	yes	yes					

Table Appendix - 2: CD and LLE, Table Appendix - 1, were rarely evaluated with the saturation of the correlation exponent (4096 heartbeats, evaluated by Dataplore) due to too complex variability and possible multi-attractor nature of the heart beat tachograms simulated by model [18] (to [23]).